Using Web Structure for Classifying and Describing Web Pages

Eric J. Glover[1], Kostas Tsioutsiouliklis[1,2], Steve Lawrence[1] David M. Pennock[1], Gary W. Flake[1]

{compuman,kt,lawrence,dpennock,flake}@research.nj.nec.com[1]
{kt}@cs.princeton.edu[2]

NEC Research Institute[1]
4 Independence Way
Princeton, NJ 08540

Computer Science Department[2]
Princeton University
Princeton, NJ 08540

ABSTRACT

The structure of the web is increasingly being used to improve organization, search, and analysis of information on the web. For example, Google uses the text in citing documents (documents that link to the target document) for search. We analyze the relative utility of document text, and the text in citing documents near the citation, for classification and description. Results show that the text in citing documents, when available, often has greater discriminative and descriptive power than the text in the target document itself. The combination of evidence from a document and citing documents can improve on either information source alone. Moreover, by ranking words and phrases in the citing documents according to expected entropy loss, we are able to accurately name clusters of web pages, even with very few positive examples. Our results confirm, quantify, and extend previous research using web structure in these areas, introducing new methods for classification and description of pages.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Search and Retrieval—*Clustering,Selection process*; H.3.6 [**Information Systems**]: Library Automation

General Terms

Algorithms, Measurement, Evaluation

Keywords

web structure, classification, SVM, entropy based feature extraction, cluster naming, web directory, anchortext

1. INTRODUCTION

The Web is a large collection of heterogeneous documents. Recent estimates predict the size of the indexable web to be more than 4 billion pages. Web pages, unlike standard text collections, can contain both multimedia (images, sounds, flash, etc.) and connections to other documents (through hyperlinks). Hyperlinks are increasingly being used to improve the ability to organize, search, and analyze the web.

Copyright is held by the author/owner(s).
WWW2002, May 7–11, 2002, Honolulu, Hawaii, USA.
ACM 1-58113-449-5/02/0005.

Hyperlinks (or citations) are being actively used to improve web search engine ranking [4], improve web crawlers [6], discover web communities [8], organize search results into hubs and authorities [13], make predictions about similarity between research papers [16] and even to classify target web pages [20, 9, 2, 5, 3]. The basic assumption made by citation or link analysis is that a link is often created because of a subjective connection between the original document and the cited, or linked to document. For example, if I am making a web page about my hobbies, and I like playing scrabble, I might link to an online scrabble game, or to the home page of Hasbro. The belief is that these connections convey meaning or judgments made by the creator of the link or citation.

On the web, a hyperlink has two components: The destination page, and associated anchortext describing the link. A page creator determines the anchortext associated with each link. For example, a user could create a link pointing to Hasbro's home page, and that user could define the associated anchortext to be "My favorite board game's home page". The personal nature of the anchortext allows for connecting words to destination pages, as shown in Figure 1. Anchortext has been utilized in this way by the search engine Google to improve web search. Google allows pages to be returned based on keywords occurring in inbound anchortext, even if the words do not occur on the page itself, such as returning `http://www.yahoo.com/` for a query of "web directory."

Typical text-based classification methods utilize the words (or phrases) of a target document, considering the most significant features. The underlying assumption is that the page contents effectively describe the page to be classified. Unfortunately, very often a web page might contain no obvious clues (textually) as to its intent. For example, the home page of Microsoft Corporation (`http://www.microsoft.com/`) provides no mention of the fact that they sell operating systems. Or the home page of General Motors (`http://www.gm.com/flash_homepage/`) does not state that they are a car company (except for the word "motors" in the title or the word "automotive" inside of a form field). To make matters worse, like a majority of web pages, the General Motors home page does not have any meaningful metatags [15].

Determining if a particular page belongs to a given class, even though the page itself does not have any obvious clues, or the words do not capture the higher-level notion can be a challenge – for example determining that GM is a car manufacturer, or Microsoft designs and sells operating systems, or Yahoo! is a directory service. Anchortext, since it is chosen by people who are interested in the

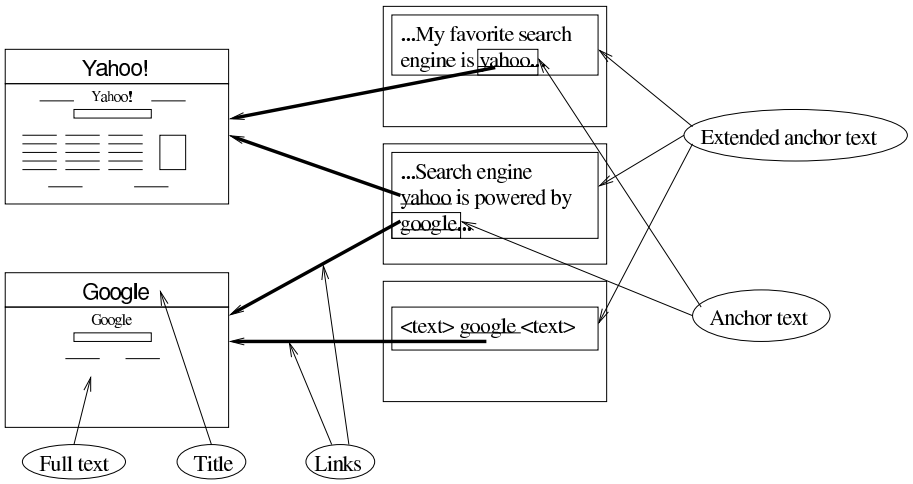

Figure 1: A diagram showing links, anchortext, and our concept of extended anchortext.

page, may better summarize the contents of the page – such as indicating that Yahoo! is a web directory, or Excite@Home is an Internet Service Provider.[1] Other works have proposed and/or utilized in-bound anchortext to help classify target web pages. For example, Blum and Mitchell [3] compared two classifiers for several computer science web pages (from the WebKB dataset), one for full-text, and one for the words on the links pointing in to the target pages (inbound anchortext). From their results, anchortext words alone were slightly less powerful than the full-text alone, and the combination was better. Other work, including work by Fürnkranz [9], expanded this notion to include words beyond the anchortext that occur near (in the same paragraph) and nearby headings. Fürnkranz noted a significant improvement in classification accuracy when using the link-based method as opposed to the full-text alone, although adding the entire text of "neighbor documents" seemed to harm the ability to classify pages [5].

The web is large, and one way to help people find useful pages is a directory service, such as Yahoo! (`http://www.yahoo.com/`), or The Open Directory Project (`http://www.dmoz.org/`). Typically directories are manually created, and the judgments of where a page goes is done by a human. For example, Yahoo! puts "General Motors" into several categories: "Auto Makers", "Parts", "Automotive", "B2B – Auto Parts", and "Automotive Dealers". Yahoo! puts itself "Yahoo!" in several categories including "Web Directories." Unfortunately large web directories are difficult to manually maintain, and may be slow to include new pages. It is therefore desirable to be able to learn an automatic classifier that tests membership in a given category. Unfortunately, the makeup of a given category may be arbitrary. For example, Yahoo! decided that Anthropology and Archaeology should be grouped together under "Social Sciences", while The Open Directory Project (dmoz) separated archaeology into its own category (also under Social Sciences). A second problem is that initially a category may be defined by a small number of pages, and classification may be difficult. A third problem is naming of a category. For example, given ten random botany pages, how would you know that the category should be named botany, or that it is related to biology? Only two of six random pages selected from the Yahoo! category of Botany mentioned the word "botany" anywhere in the text (although some had it in the URL, but not the body text). For human-generated clusters it may be reasonable to assume a name can be found, however, for automatically generated clusters, naming may be more difficult.

This work attempts to utilize inbound anchortext and surrounding words to classify pages accurately, and to name (potentially very small) clusters of web pages. We make no assumptions about having a web-crawl. We also quantify the effectiveness of using just a web-page's full-text, inbound anchortext, and what we call extended anchortext (the words and phrases occurring near a link to a target page, as shown in Figure 1), and propose two methods for improving the classification accuracy: a combination method and uncertainty sampling. We also extract important features that can be used to name the clusters, and compare the ability of using only a document's full-text with using in-bound anchortexts and extended anchortexts.

Our approach to basic text-classification is based on a simple four-step procedure, described in Figure 2: First, obtain a set of positive and negative training documents. Second, extract all possible features from these documents (a feature in this case is a word or phrase). Third, perform entropy-based dimensionality reduction. Fourth, train an SVM classifier. Naming of clusters can be done by examining the top ranked features after the entropy-based dimensionality reduction. The learned classifier can then be evaluated on test data.

In comparison to other work on using link-structure to classify web pages, we demonstrate very high accuracy—more than 98% on average for negative documents, and as high as 96% for positive documents, with an average of about 90%.[2] Our experiments used about 100 web pages from each of several Yahoo! categories for positive training and test data, and random web pages as negative examples (significantly fewer than other methods). Positive pages were obtained by choosing all web documents listed in the chosen category, plus all documents from several sub-categories. The set of positive and negative documents was randomly split to create training and test sets. We also evaluated the ability to name the clusters, using small samples from several Yahoo! categories as positive examples. In every case the name of the Yahoo! category was listed as the top ranked or second ranked feature, and the name of the parent category was listed in the top 10 in every case but one. In addition, many of the top ranked features described the names of

[1] Their homepage: (`http://www.home.com/index_flash.html`) has no text, and no metatags. On a text-browser such as Lynx, the rendered page is blank.

[2] Accuracy of one class is the recall of that class.

the sub-categories (from which documents were drawn).

2. OUR METHOD

First, we describe our method for extracting important features and training a full-text classifier of web pages. Second, we describe our technique for creating "virtual documents" from the anchortext and inbound extended anchortext. We then use the virtual documents as a replacement for the full-text used by our original classifier. Third, we describe our method for combining the results to improve accuracy. Fourth, we describe how to name a cluster using the features selected from the virtual documents.

2.1 Full-Text Classifier

In our earlier works, we described our algorithm for full-text classification of web pages [10, 11]. The basic algorithm is to generate a feature histogram from training documents, select the "important features", and then to train an SVM classifier. Figure 2 summarizes the high-level procedure.

Step 1: Obtain positive and negative document sets
Step 2: Generate a positive and negative histogram of all features
Step 3: Select significant features using expected entropy loss
Step 4: Train an SVM using the selected features

Figure 2: Basic procedure for learning a text-classifier

2.1.1 Training Sets and Virtual Documents

To train a binary classifier it is essential to have sets of both positive and negative documents. In the simplest case, we have a set of positive web pages, and a set of random documents to represent negative pages. The assumption is that few of the random documents will be positive (our results suggested less than 1% of the random pages we used were positive). In our first case documents are the full-text found by downloading the pages from various Yahoo! categories.

Unfortunately, the full-text of a document is not necessarily representative of the "description" of the documents, and research has shown that anchortext can potentially be used to augment the full-text of a document [20, 9, 3]. To incorporate anchortexts and extended anchortexts, we replaced actual downloaded documents with *virtual documents*. We define a virtual document as a collection of anchortexts or extended anchortexts from links pointing to the target document. Our definition is similar to the concept of "blurbs" described by Attardi et al. [2]. This is similar to what was done by Fürnkranz [9]. Anchortext refers to the words occurring inside of a link as shown in Figure 1. We define extended anchortext as the set of rendered words occurring up to 25 words before and after an associated link (as well as the anchortext itself). Figure 1 also shows an example of extended anchortext. Fürnkranz considered the actual anchortext, plus headings occurring immediately preceding the link, and the paragraph of text containing the link. Our approach is similar, except it made no distinction between other HTML structural elements. Our goal was to compare the ability to classify web pages based on just the anchortext or extended anchortext, just the full-text, or a combination of these. Figure 3 shows a sample virtual document. For our work, we limited the virtual document to 20 inbound links, always excluding any Yahoo! pages, to prevent the Yahoo! descriptions or category words from biasing the results.

To generate each virtual document, we queried the Google search engine for backlinks pointing into the target document. Each backlink was then downloaded, the anchortext, and words before and after each anchortext were extracted. We generated two virtual documents for each URL. One consisting of only the anchortexts and the other consisting of the extended anchortexts, up to 25 words on each side of the link, (both limited to the first 20 non-Yahoo! links). Although we allowed up to 20 total inbound links, only about 25% actually had 20 (or more). About 30% of the virtual documents were formed with three or fewer inbound links. If a page had no inbound links, it was not considered for this experiment. Most URLs extracted from Yahoo! pages had at least one valid-non Yahoo! link.

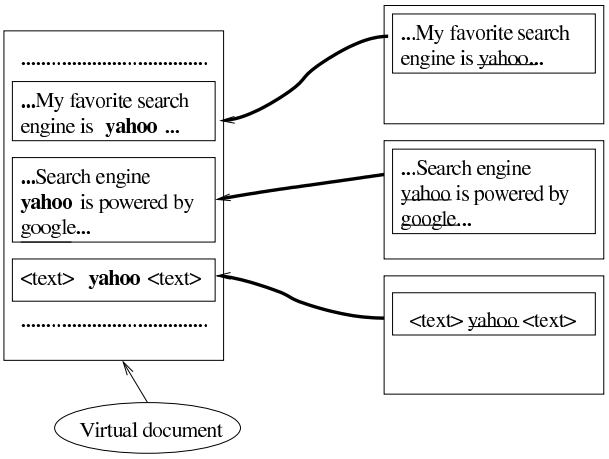

Figure 3: A virtual document is comprised of anchortexts and nearby words from pages that link to the target document.

2.1.2 Features and Histograms

For this experiment, we considered all words and two or three word phrases as possible features. We used no stopwords, and ignored all punctuation and HTML structure (except for the Title field of the full-text documents). Each document (or virtual document) was converted into a set of features that occurred and then appropriate histograms were updated.

For example: If a document had the sentence: "My favorite game is scrabble", the following features are generated: `my`, `my favorite`, `my favorite game`, `favorite`, `favorite game`, `favorite game is`, etc. From the generated features an appropriate histogram is updated. There is one histogram for the positive set and one for the negative set.

Unfortunately, there can be hundreds of thousands of unique features, most that are not useful, occurring in just hundreds of documents. To improve performance and generalizability, we perform dimensionality reduction using a two step process. This process is identical to that described in our earlier works [10, 11].

First, we perform thresholding, by removing all features that do not occur in a specified percentage of documents as rare words are less likely to be useful for a classifier. A feature f is removed if it occurs in less than the required percentage (threshold) of both the positive and negative sets, i.e.,

$$\left(|\mathcal{A}_f|/|\mathcal{A}| < \mathcal{T}^+\right) \text{ and } \left(|\mathcal{B}_f|/|\mathcal{B}| < \mathcal{T}^-\right)$$

Where:

- $\mathcal{A}$: the set of positive examples.
- $\mathcal{B}$: the set of negative examples.
- $\mathcal{A}_f$: documents in $\mathcal{A}$ that contain feature f.

- $\mathcal{B}_f$: documents in $\mathcal{B}$ that contain feature f.
- $\mathcal{T}^+$: threshold for positive features.
- $\mathcal{T}^-$: threshold for negative features.

Second, we rank the remaining features based on entropy loss. No stop word lists are used.

2.1.3 Expected Entropy Loss

Entropy is computed independently for each feature. Let C be the event indicating whether the document is a member of the specified category (e.g., whether the document is about "biology"). Let f denote the event that the document contains the specified feature (e.g., contains "evolution" in the title). Let $\overline{C}$ and $\overline{f}$ denote non-membership and the absence of a specified feature respectively. The prior entropy of the class distribution is $e \equiv -\Pr(C)\lg\Pr(C) - \Pr(\overline{C})\lg\Pr(\overline{C})$. The posterior entropy of the class when the feature is present is $e_f \equiv -\Pr(C|f)\lg\Pr(C|f) - \Pr(\overline{C}|f)\lg\Pr(\overline{C}|f)$; likewise, the posterior entropy of the class when the feature is absent is $e_{\overline{f}} \equiv -\Pr(C|\overline{f})\lg\Pr(C|\overline{f}) - \Pr(\overline{C}|\overline{f})\lg\Pr(\overline{C}|\overline{f})$. Thus, the expected posterior entropy is $e_f Pr(f) + e_{\overline{f}} Pr(\overline{f})$, and the *expected entropy loss* is

$$e - \left(e_f \Pr(f) + e_{\overline{f}} \Pr(\overline{f})\right).$$

If any of the probabilities are zero, we use a fixed value. Expected entropy loss is synonymous with expected information gain, and is *always* non-negative [1].

All features meeting the threshold are sorted by expected entropy loss to provide an approximation of the usefulness of the individual feature. This approach assigns low scores to features that, although common in both sets, are unlikely to be useful for a binary classifier.

One of the limitations of using this approach is the inability to consider co-occurrence of features. Two or more features individually may not be useful, but when combined may become highly effective. Coetzee et al. [7] discuss an optimal method for feature selection in. Our method, although not optimal, can be run in constant time per feature with constant memory per feature, plus a final sort,[3] both significantly less than the optimal method described by Coetzee. We perform several things to reduce the effects of possible feature co-occurrence. First, we consider both words and phrases (up to three terms). Considering phrases reduces the chance that a pair of features will be missed. For example, the word "molecular" and the word "biology" individually may be poor at classifying a page about "molecular biology", but the phrase is obviously useful.

A second approach to reducing the problem is to consider many features, with a relatively low threshold for the first step. The SVM classifier will be able to identify features as important, even if individually they might not be. As a result, considering a larger number of features can reduce the chance that a feature is incorrectly missed due to low individual entropy. For our experiments, we typically considered up to a thousand features for each classifier, easily handled by an SVM. We set our thresholds at 7% for both the positive and negative sets.

2.1.4 Using Entropy Ranked Features to Name Clusters

Ranking features by expected entropy loss (information gain) allows us to determine which words or phrases optimally separate a given positive cluster from the rest of the world (random documents). As a result, it is likely that the top ranked features will meaningfully describe the cluster. Our earlier work on classifying web pages for Inquirus 2 [10, 11] considered document full-text (and limited structural information) and produced features consistent with the "contents" of the pages, not necessarily with the "intentions" of them. For example, for the category of "research papers" top ranked features included: "abstract", "introduction", "shown in figure". Each of these words or phrases describe "components" of a research paper, but the phrase "research paper" was not top ranked. In some cases the "category" is similar to words occurring in the pages, such as for "reviews" or "calls for papers". However, for arbitrary Yahoo! categories, it is unclear that the document text (often pages have no text) are as good an indication of the "description" of the category.

To name a cluster, we considered the features extracted from the extended anchortext virtual documents. We believe that the words near the anchortexts are descriptions of the target documents, as opposed to "components of them" (such as "abstract" or "introduction"). For example, a researcher might have a link to their publications saying "A list of my research papers can be found here". The top ranked features by expected entropy loss are those which occur in many positive examples, and few negative ones, suggesting that they are a consensus of the descriptions of the cluster, and least common in random documents.

2.1.5 SVMs and Web Page Classification

Categorizing web pages is a well researched problem. We chose to use an SVM classifier [19] because it is resistant to overfitting, can handle large dimensionality, and has been shown to be highly effective when compared to other methods for text classification [12, 14]. A brief description of SVMs follows.

Consider a set of data points, $\{(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_N, y_N)\}$, such that $\boldsymbol{x}_i$ is an input and y_i is a target output. An SVM is calculated as a weighted sum of kernel function outputs. The kernel function of an SVM is written as $K(\boldsymbol{x}_a, \boldsymbol{x}_b)$ and it can be an inner product, Gaussian, polynomial, or any other function that obeys Mercer's condition.

In the case of classification, the output of an SVM is defined as:

$$f(\boldsymbol{x}, \boldsymbol{\lambda}) = \sum_{i=1}^{N} y_i \lambda_i K(\boldsymbol{x}_i, \boldsymbol{x}) + \lambda_0. \quad (1)$$

The objective function (which should be minimized) is:

$$E(\boldsymbol{\lambda}) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j \lambda_i \lambda_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) - \sum_{i=1}^{N}\lambda_i, \quad (2)$$

subject to the box constraint $0 \le \lambda_i \le C, \forall_i$ and the linear constraint $\sum_{i=1}^{N} y_i \lambda_i = 0$. C is a user-defined constant that represents a balance between the model complexity and the approximation error. Equation 2 will always have a single minimum with respect to the Lagrange multipliers, λ. The minimum to Equation 2 can be found with any of a family of algorithms, all of which are based on constrained quadratic programming. We used a variation of Platt's Sequential Minimal Optimization algorithm [17, 18] in all of our experiments.

When Equation 2 is minimal, Equation 1 will have a classification margin that is maximized for the training set. For the case of a linear kernel function ($K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i \cdot \boldsymbol{x}_j$), an SVM finds a decision boundary that is balanced between the class boundaries of the two classes. In the nonlinear case, the margin of the classifier is maximized in the kernel function space, which results in a nonlinear classification boundary.

[3] We assume that the histogram required for computation is generated separately, and we assume a constant time to look up data for each feature from the histogram.

Yahoo! Category	Parent	Training	Test
Biology	Science	100/400	113/300
Archaeology	Anthropology and Archaeology	100/400	145/300
Wildlife	Animals, Insects, and Pets	100/400	120/300
Museums, Galleries, and Centers	Arts	75/500	100/300
Management Consulting	Consulting	300/500	100/300

Table 1: Yahoo! categories used to test classification accuracy, numbers are positive / negative

Yahoo! Category	Full-Text	Anchortext	Extended-AT	Combined	Sampled	% Sampled
Biology	51.3/90	55.1/97.3	72.9/98	80.4/97.3	83.1/98	9.8
Archaeology	65.5/92.7	72.2/98.3	83.2/99.2	91.6/98.4	94.4/99.2	8.7
Wildlife	83.3/97.3	76.7/99	87.1/99	96.6/99	96.6/99	4.6
Museums	57/93.7	80/98	87/98.7	89/98.3	94/98.7	6.3
Mgmt Consulting	74/88.7	56.7/95	81.1/95	88.9/92.3	92.2/95	9.5
Average	66.2/92.5	68.3/97.5	82.2/98	89.3/97.1	92.1/98.0	7.7

Table 2: Percentage accuracy of five different methods (pos/neg), sampled refers to the uncertainty sampled case

When using a linear kernel function, the final output is a weighted feature vector with a bias term. The returned weighted vector can be used to quickly classify a test document by simply taking the dot product of the features.

2.2 Combination Method

This experiment compares three different methods for classifying a web page: full-text, anchortext only, and extended anchortext only. Section 3 describes the individual results. Although of the three, extended anchortext seems the most effective, there are specific cases for which a document's full-text may be more accurate. We wish to meaningfully combine the information to improve accuracy. The result from an SVM classifier is a real number from $-\infty$ to $+\infty$, where negative numbers correspond to a negative classification, and positive numbers correspond to a positive classification. When the output is on the interval $(-1, 1)$ it is less certain than if it is on the intervals $(-\infty, -1)$ and $(1, \infty)$. The region $(-1, 1)$ is called the "uncertain region".

We describe two ways to improve the accuracy of the extended anchortext classifier. The first is through uncertainty sampling, where a human judges the documents in the "uncertain region." The hope is that both the human judges are always correct, and that there are only a small percentage of documents in the uncertain region. Our experimental results confirm that for the classifiers based on the extended anchortext, on average about 8% of the total test documents (originally classified as negative) were considered uncertain, and separating them out demonstrated a substantial improvement in accuracy.

The second method is to combine results from the extended anchortext based classifier with the less accurate full-text classifier. Our observations indicated that the negative class accuracy was approaching 100% for the extended anchortext classifier, and that many false negatives were classified as positive by the full-text classifier. As a result, our combination function only considered the full-text classifier when a document was classified as negative, but uncertain, by the extended anchortext classifier. For those documents, a positive classification would result if the full-text classifier resulted in a higher magnitude (but positive) classification. Our automatic method resulted in a significant improvement in positive class accuracy (average increase from about 83% to nearly 90%), but had more false positives, lowering negative class accuracy by about a percentage point from 98% to about 97%.

3. RESULTS

Our goal was to compare three different sources of features for training a classifier for web documents: full-text, anchortext and extended anchortext. We also wished to compare the relative ability to name clusters of web documents using each source of features.

To compare these methods, we chose several Yahoo! categories (and sub-categories) and randomly chose documents from each. The Yahoo! classified documents formed the respective positive classes, and random documents (found from outside Yahoo!) comprised the negative class. In addition, the Yahoo! assigned category names were used as a benchmark for evaluating our ability to name the clusters. In all cases virtual documents excluded links from Yahoo! to prevent using their original descriptions to help name the clusters.

We also tried classifying the categories of courses and faculty from the WebKB dataset used by Blum and Mitchell [3] and Fürnkranz [9]. The WebKB dataset provided a set of data called "neighborhood words" which was the text occurring in the same "paragraph" as the inlink to a given document. Unfortunately most of the inlinks were in list items, causing neighborhood words to be only slightly more than the anchortext itself. The dataset also only considered pages from within four Universities, so the number of inlinks was very limited—most pages had only one inlink.

3.1 Text Classification

The categories we chose for classification, and the training and test sizes are listed in Table 1. For each case we chose the documents listed in the category itself (we did not follow Yahoo! links to other Yahoo! categories) and if there were insufficient documents, we chose several sub-categories to add documents. Table 2 lists the results for each of the classifiers from Table 1.

In addition to the Yahoo! categories, we tried applying SVM classification to the WebKB categories of courses and faculty. For training of courses, we used 144 positive and 1000 negative (from the "other" category), and for training of the faculty category we used 84 positive and the same 1000 negative. For the category courses there were 1000 negative test documents, and 70 positive test examples, for an accuracy of 96.8% negative and 67% for the positive. For the category of faculty, there were 70 positive and 1000 negative test, with an accuracy of 99% negative, and 64.3% positive. Both of these are similar to the accuracy reported for full-text classification of the WebKB data by Fürnkranz [9].[4] The use

[4]It is difficult to make a comparison between a binary classifier and

biology (full-text)	**biology** (anchortext)	**biology** (extended)	**archaeology** (full-text)	**archaeology** (anchortext)	**archaeology** (extended)
biology	http	**biology**	**archaeology**	**archaeology**	**archaeology**
dna	http www	science	archaeological	archaeological	archaeological
biological	edu	molecular	ancient	museum	ancient
cell	html	biological	archaeologists	the	museum
university	**biology**	university	stone	museum of	anthropology
molecular	the	university of	Title:archaeology	of	history
research	human	human	excavation	archeology	of archaeology
protein	cell	research	of archaeology	http	research
human	of	molecular biology	museum	university	prehistoric

Table 3: Top 10 ranked features by expected entropy loss. Bold indicates a category word, underline indicates a parent category word.

wildlife (full-text)	**wildlife** (anchortext)	**wildlife** (extended)	**museums** (full-text)	**museums** (anchortext)	**museums** (extended)
wildlife	**wildlife**	**wildlife**	**museum**	art	**museum**
Title:wildlife	species	conservation	museum of	**museum**	museum of
species	org	species	art	contemporary	art
endangered	endangered	animals	of art	museum of	of art
wild	conservation	wild	**gallery**	contemporary art	**gallery**
conservation	endangered species	endangered	contemporary art	**gallery**	contemporary art
habitat	sanctuary	animal	contemporary	org	contemporary
animals	http	nature	art museum	museums	art museum
endangered species	refuge	and wildlife	arts	of	arts

Table 4: Top 10 ranked features by expected entropy loss. Bold indicates a category word, underline indicates a parent category word.

of the words occurring in the same paragraph of the inbound links produced slightly worse accuracy than the full-text, likely due to the very small number of inlinks, and the small number of words occurring in the same paragraph.

When evaluating the accuracy, it is important to note several things. First, the negative accuracy is a lower-bound since negative pages were random, and thus some could actually be positive. We did not have time to manually examine all random pages. However, a cursory examination of the pages classified as positive, but from the random set, showed about 1 in 3 were actually positive – suggesting negative class accuracy was more than 99% in many cases. It is also important to note the relatively small set sizes used for training. Our positive sets typically had 100 examples, relatively small considering there were as many as 1000 features used for training. Positive accuracy is also a lower bound since sometimes pages may be misclassified by Yahoo!. It is also important to note that we are performing binary classification. We believe that pages may belong to multiple (or zero) categories, so it is reasonable to create a separate classifier for each one.

Other works comparing accuracy of full-text to anchortext have not shown a clear difference in classification ability, or a slight loss due to use of anchortext alone [9]. Our results suggest that anchortext alone is comparable for classification purposes with the full-text. Several papers agree that features on linking documents, in addition to the anchortext (but less than the whole page) can provide significant improvements. Our work is consistent with these results, showing significant improvement in classification accuracy when using the extended anchortext instead of the document full-text. For comparison, we applied our method (for both classification and naming) to full-texts for the categories of courses and faculty from the WebKB dataset.

Our combination method is also highly effective for improving positive-class accuracy, but reduces negative class accuracy. Our method for uncertainty sampling required examining only 8% of the documents on average, while providing an average positive class accuracy improvement of almost 10 percentage points. The automatic combination also provided substantial improvement over the extended anchortext or the full-text alone for positive accuracy, but caused a slight reduction in negative class accuracy as compared to the extended anchortext case.

3.2 Features and Category Naming

The second goal of this research is to automatically name various clusters. To test our ability to name clusters we compared the top ranked features (by expected entropy loss) with the Yahoo! assigned names. We performed several tests, with as few as 4 positive examples. Tables 3, 4 and 5 show the top 10 ranked features for each of the five categories above for the full-text, the anchortext only, and extended anchortext.

The full-text appears comparable to the extended anchortext, within all five cases, the current category name appearing as the top or second ranked feature, and the parent category name appearing in the top 10 (or at least one word from the category name). The extended anchortext appears to perform similarly, with an arguable advantage, with the parent name appearing more highly ranked. The anchortext alone appears to do a poor job of describing the category, with features like "and" or "http" ranking highly. This is likely due to the fact people often put the URL or the name of the target page as the anchortext. The relatively high thresholds (7%) removed most features from the anchortext-only case. From the five cases there was an average of about 46 features surviving the threshold cut-offs for the anchortext only case. For the full-text and extended anchortext, usually there were more than 800 features surviving the thresholds. Table 6 shows the results for small clusters for the same categories and several sub-categories. In every case the category name was ranked first or second, with the parent name

an n-way classifier.

management consulting (full-text)	**management consulting** (anchortext)	**management consulting** (extended anchortext)
management	**consulting**	**management**
consulting	inc	**consulting**
clients	**management**	associates
Title:management	group	consultants
strategic	associates	business
business	com	group
Title:consulting	consulting group	firm
consultants	group inc	consulting firm
services	com www	**management consulting**

Table 5: Top 10 ranked features by expected entropy loss. Bold indicates a category word, underline indicates a parent category word.

biology (20)	**botany (8)**	**wildlife (4)**	**conservation and research (5)**	**isps (6)**
biology	plant	**wildlife**	wildlife	internet service
science	**botany**	animals	**conservation**	isps
biological	of plant	conservation	endangered	modem
molecular	the plant	insects	natural	earthlink
genetics	botanical	endangered	species	broadband
human	plants	the conservation	research center	providers
evolution and	biology	facts	society http www	service provider
genomics	internet directory	wild	wildlife trust	prodigy
anatomy	botanic	bat	society http	internet service provider
paleontology	botanical garden	totally	wildlife society	atm

Table 6: Ranked list of features from extended anchortext by expected entropy loss. Number in parentheses is the number of positive examples.

ranked highly.[5] In addition, most of the other top ranked features described names of sub-categories. The ISP example was one not found in Yahoo!. For this experiment, we collected the home pages of six ISPs, and attempted to discover the commonality between them. The full-text based method reported features common to the portal home pages, "current news", "sign in", "channels" "horoscopes", etc. However, the extended anchortext method correctly named the group "isps" or "internet service provider", despite the fact that none of the pages mentioned either term anywhere on their homepage, with only Earthlink and AT&T Worldnet mentioning the phrase "internet service provider" in a metatag. A search on Google for "isp" returned none of the ISPs used for this experiment in the top 10. A search for "internet service provider" returned only Earthlink in the top 10.

We also examined the top ranked features (by expected entropy loss) from the full-text of the WebKB dataset categories of courses and faculty. From our training data described in Section 3.1, the top two ranked features from courses were: "courses" and "office hours". The top two ranked features for the faculty category were: "professor" and "ph d".

4. SUMMARY AND FUTURE WORK

This paper describes a method for learning a highly accurate web page classifier, and using the intermediate feature-set to help name clusters of web pages. We evaluated our approach on several Yahoo! categories, with very high accuracy for both classification and for naming. Our work supports and extends other work on using web structure to classify documents, and demonstrates the usefulness of considering inbound links, and words surrounding them. We also show that anchortext alone is not significantly better (arguably worse) than using the full-text alone. We also present two simple methods for improving the accuracy of our extended anchortext classifier. Combining the results from the extended anchortext classifier with the results from the full-text classifier produces nearly a 7 percentage point improvement in positive class accuracy. We also presented a simple method for uncertainty sampling, where documents that are uncertain are manually evaluated, improving the accuracy nearly 10 percentage points, while requiring on-average less than 8% of the documents to be examined.

Utilizing only extended anchortext from documents that link to the target document, average accuracy of more than 82% for positive documents, and more than 98% for negative documents was achieved, while just considering the words and phrases on the target pages (full-text) average accuracy was only 66.2% for positive documents, and 92.5% for negative documents. Combing the two resulted in an average positive accuracy of almost 90%, with a slight reduction in average negative accuracy. The uncertainty sampled case had an average positive accuracy of more than 92%, with the negative accuracy averaging 98%.

Using samples of as few as four positive documents, we were able to correctly name the chosen Yahoo! category (without using knowledge of the Yahoo! hierarchy) and in most cases rank words that occurred in the Yahoo!-assigned parent category in the top 10 features. The ability to name clusters comes for free from our entropy-based feature ranking method, and could be useful in creating automatic directory services.

Our simplistic approach considered only up to 25 words before and after (and the included words) an inbound link. We wish to expand this to include other features on the inbound web pages, such as structural information (e.g., is a word in a link or heading), as well as experiment with including headings of the inbound pages near the anchortext, similar to work done by Fürnkranz [9]. We also

[5]In the case of "conservation and research", the Yahoo! listed parent category was "organizations", which did not appear as a top ranked feature, there were only three top level sub-categories under wildlife, suggesting that conservation and research could be promoted.

wish to examine the effects of the number of inbound links, and the nature of the category by expanding this to thousands of categories instead of only five. The effects of the positive set size also need to be studied.

5. REFERENCES

[1] N. Abramson. *Information Theory and Coding*. McGraw-Hill, New York, 1963.

[2] G. Attardi, A. Gullí, and F. Sebastiani. Automatic Web page categorization by link and context analysis. In C. Hutchison and G. Lanzarone, editors, *Proceedings of THAI-99, 1st European Symposium on Telematics, Hypermedia and Artificial Intelligence*, pages 105–119, Varese, IT, 1999.

[3] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT: Proceedings of the Workshop on Computational Learning Theory, Morgan Kaufmann Publishers*, 1998.

[4] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *WWW7*, Brisbane, Australia, 1998.

[5] S. Chakrabarti, B. Dom, and P. Indyk. Enhanced hypertext categorization using hyperlinks. *SIGMOD Record (ACM Special Interest Group on Management of Data)*, 27(2):307–318, June 1998.

[6] J. Cho, H. García-Molina, and L. Page. Efficient crawling through URL ordering. *Computer Networks and ISDN Systems*, 30(1–7):161–172, 1998.

[7] F. Coetzee, E. Glover, S. Lawrence, and C. L. Giles. Feature selection in web applications using ROC inflections. In *Symposium on Applications and the Internet, SAINT*, pages 5–14, San Diego, CA, January 8–12 2001. IEEE Computer Society, Los Alamitos, CA.

[8] G. W. Flake, S. Lawrence, and C. L. Giles. Efficient identification of web communities. In *Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD-2000)*, Boston, MA, 2000. ACM Press.

[9] J. Fürnkranz. Exploiting structural information for text classification on the WWW. In *Intelligent Data Analysis*, pages 487–498, 1999.

[10] E. Glover, G. Flake, S. Lawrence, W. P. Birmingham, A. Kruger, C. L. Giles, and D. Pennock. Improving category specific web search by learning query modifications. In *Symposium on Applications and the Internet, SAINT*, San Diego, CA, January 8–12, 2001.

[11] E. J. Glover. *Using Extra-Topical User Preferences To Improve Web-Based Metasearch*. PhD thesis, University of Michigan, 2001.

[12] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Tenth European Conference on Machine Learning ECML-98*, pages 137–142, 1999.

[13] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[14] J. T.-Y. Kwok. Automated text categorization using support vector machine. In *Proceedings of the International Conference on Neural Information Processing (ICONIP)*, pages 347–351, Kitakyushu, Japan, 1999.

[15] S. Lawrence and C. L. Giles. Accessibility of information on the web. *Nature*, 400(July 8):107–109, 1999.

[16] S. Lawrence, C. L. Giles, and K. Bollacker. Digital libraries and Autonomous Citation Indexing. *IEEE Computer*, 32(6):67–71, 1999.

[17] J. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Scholkopf, C. Burges, and A. Smola, editors, *Advances in kernel methods - support vector learning*. MIT Press, 1998.

[18] J. Platt. Using sparseness and analytic QP to speed training of support vector machines. In *Advances in Neural Information Processing Systems*, 1999.

[19] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 1995.

[20] Y. Yang, S. Slattery, and R. Ghani. A study of approaches to hypertext categorization. *Journal of Intelligent Information Systems*. Kluwer Academic Press, (accepted), 2001.