Exploiting Structural Information for Text Classification on the WWW

Johannes Fürnkranz

Austrian Research Institute for Artificial Intelligence
Schottengasse 3, A-1010 Wien, Austria
juffi@ai.univie.ac.at

Abstract. In this paper, we report on a set of experiments that explore the utility of making use of the structural information of WWW documents. Our working hypothesis is that it is often easier to classify a hypertext page using information provided on pages that point to it instead of using information that is provided on the page itself. We present experimental evidence that confirms this hypothesis on a set of Web-pages that relate to Computer Science Departments.

1 Introduction

The advent of the World-Wide Web has rejuvinated the interest in text categorization problems. Vast amounts of documents are available on-line, and categorizing them into meaningful semantic categories is a rewarding and challenging research problem.

However, current approaches to text categorization on the Web mostly concentrate on simple representation schemes that are based on word occurrence and word frequency. The structural information that is inherent to documents on the Web is often neglected. There are at least two different kinds of structural information on the Web that could be used to enhance the performance of current text classification algorithms:

- the structure of an HTML representation which allows to easily identify important parts of a document, such as its headings and its title, and
- the structure of the Web itself, where pages are linked to each other in various ways.

In this paper, we report on a set of experiments that explores the utility of such structural information. Our working hypothesis is that (at least in some domains) it is easier to classify hypertext pages using information provided on pages that point to a page instead of using information that is provided on the page itself. There are several reasons for this:

Redundancy: Quite often there is more than one page pointing to a single page on the Web. The ability to combine multiple, independent sources of information can improve classification accuracy.

D.J. Hand, J.N. Kok, M.R. Berthold (Eds.): IDA'99, LNCS 1642, pp. 487–497, 1999.

Independent Labeling: Being able to rely on the information provided by multiple authors (the authors of the pages that point to the page to be classified) is less sensitive than having to rely on the vocabulary used by one particular author [11].

Page Sparseness: Web pages are often very sparse or contain mostly images. Using the links to a page increases the chances of encountering informative text about the page to classify.

To investigate our hypothesis, we represent a Web page with features derived from information of pages that point to the page. To that end, we encode each hyperlink pointing to a document with its anchor text, the headings structurally preceding it, and the text of the paragraph in which it occurs. Then we learn a set of classification rules with the inductive rule learning algorithm RIPPER [1]. The predictions of links pointing to the same page are then combined to yield a prediction for this page.

Our results show that documents can often be classified more reliably with information originating from pages that point to the document than with features that are derived from the document text itself.

2 Motivation

Our approach for the use of structural information for classifying Web-pages was motivated by the following observation that we made while working with conventional text classification techniques on the WebKB data set.[1]

Observation 1: *The text on the pages themselves is often insufficient or irrelevant for a reliable classification.*

For example, home pages of computer science departments often only consist of images with pointers to information about offered courses, student and faculty home pages, research projects, etc. Even if this information is contained on a single page, the words on the page itself do not provide many clues for the fact that we are dealing with the home page of a computer science department as opposed to any other page in a computer science department.

Observation 2: *Information on the pages that contain a pointer to a given page is much more helpful. Very often, at least one of the following three pieces of information contains an obvious clue for the intended classification of the page.*

1. *the anchor text*
2. *the context in which the anchor text appears*
3. *the headings that structurally precede the section of the document in which the link occurs*

For example, department pages typically have a large number of links pointing to them that are marked with anchor texts that include phrases like "computer science department", "CS department", "dept. of computer science", or

[1] A brief description of this domain can be found in section 5

similar. Each of them should be sufficient to identify the link as pointing to the page of a computer science department. Student home pages very often contain a pointer to their advisor's home page. Thus, faculty home pages can often be identified by the occurrence of the word "advisor" in the neighborhood of a link that points to the page. Furthermore, many computer science departments have a page that lists all students, faculty, staff, projects, courses, or other information. Typically such a page (or segment of a page) starts with a heading that identifies the type of information that is listed below it. Clearly, this information can also be very useful for classifying the pages that the list items below this heading point to.

3 Document Representation

In order to capture its structural information, we represented a document in the following way. First, the entire text of the document itself was discarded. Instead, we identified a set of pages that contain a pointer to the current page.[2] Each of these pages was turned into a separate training example using the following pieces of information:

Anchor: All words that occurred in the anchor text of the link (between the opening `<A ...>` and the closing `</A>` of the HTML link).

Heading: All words that occurred in headings that *structurally* precede the hyperlink in the HTML document. This means a heading of type `<H`i`>` is included iff it appears before the hyperlink and no heading of type `<H`j`>` with $j \leq i$ appears in the segment between the heading and the hyperlink. Page titles and titles for definition lists (`<DT>`) were also included as headings (with $i = 0$ and $i = 7$ respectively).

Paragraph: All words of the paragraph in which the hyperlink occurs. Our method for determining the paragraph is somewhat heuristic and certainly not perfect. Pieces of text separated by `<P>` or an empty line are paragraphs, as are structural entities such as items in a list `<LI>`.

The three features described above were each encoded as a separate set-valued feature [2] for the separate-and-conquer rule learner RIPPER [1], which achieves noise-tolerance through an extension of incremental reduced error pruning [7]. A set-valued feature may be viewed as an efficient encoding of a group of binary features that correspond to the occurrences of words in the document. We will also refer to set-valued features as *feature sets* and use the terms *feature* and *word* interchangeably. Each training example was labelled with the appropriate class information, which is the class of the page the link points to.

[2] We did this by scanning a collection of pages of Computer Science departments for all occurrences of an HREF that contains the address of the current page. In principle, this could also be performed on-line using a search engine like ALTAVISTA that allows to query for pages that point to a given address.

4 Voting Schemes

As discussed above, the training sets for RIPPER consist of one example for each hyperlink. From such a set, RIPPER induces a set of unordered rules[3] that discriminate the examples of each class from the examples of all other classes. At prediction time, RIPPER selects among all rules that fire for a given example the one that has the highest confidence associated with it and uses it to classify the example.

Quite frequently, however, several links point to the same page. As our goal is to the predict the class of a page (and not of each individual link) we can try to exploit the redundancy that is provided by such multiple links. In order to do so, we have to device strategies for combining the predictions of all hyperlinks pointing to a page into a single prediction for the class of the page. We implemented the following five straight-forward techniques:

Voting *(Vote (all))*: The simplest technique is to give each link that points to a page one vote, and predict the class that receives the most votes. Ties are broken in favor of larger classes. Links that are classified using the default rule learned by RIPPER (i.e., the rule specifying that if no other rule applies, predict the majority class among all unexplained examples in the training set) are eligible to vote.

Restricted Voting *(Vote)*: It is reasonable to assume that there will always be a few links that are classified by the default rules. Thus we implemented another version of the voting scheme, where votes of such links are ignored and only links that were classified by non-default rules are eligible to vote. If a page only receives votes from default rules, it is classified with the majority class.

Weighted Sum *(Weight)*: We also associate a confidence score with each of RIPPER's predictions, which simply consists of the Laplace-estimate $\frac{p+1}{p+n+2}$ of the probability that an example covered by the rule is positive (estimated on the training set). If the prediction originates from a default rule, it is assigned a score of 0. Such a score is computed for each possible class of each link. The *Weight* voting scheme simply returns the sum of all weights as the confidence score of the prediction.

Weighted Normalized Sum *(Norm)*: This voting scheme is identical to the previous one, except that the confidence scores are first normalized in a way that distributes a total weight of 1 among the different candidate classes for each link. This is necessary because the confidence score that is associated with each class only depends on the number of positive and negative examples covered by the best rule that predicts this class and covers the example. Therefore the confidence scores associated with each class cannot be interpreted as class probability estimates unless they are normalized.

[3] We have also experimented with ordered rule sets, but the results were usually a little worse. Besides, in "ordered" mode, RIPPER treats one class as the default class and does not learn rules for that class.

Maximum Confidence *(Max)*: The last combination method simply chooses the class prediction that receives the highest score over all links that point to the page to classify and predicts that class. This is an attempt to use only the most accurate of all applicable rules to classify a page.

From a Machine Learning perspective, the problem can be viewed as combining the predictions for different training examples, for which it is known that they have the same class label. To the extent to which the predictions of the classifier for different training examples are independent of each other (which roughly corresponds to the extent to which the feature vector representation of the examples differ), it can be expected that combining the predictions may yield a performance gain [5].

5 Experimental Setup

We performed a series of experiments on 1050 pages of the WebKB domain. These pages are classified into one of the categories *Student*, *Department*, *Faculty*, *Research Project*, *Research Associate*, *Post Doc*, and *Course*. Within these pages, 5803 hyperlinks point to another page within this set. Each of these is turned into a separate training example using the set-valued features described above.

The pages/links were collected from four universities. All reported results are from a 4-fold leave-one-university-out cross-validation, i.e., for each experiment we combined the examples of three universities to learn a classifier which was then tested on the data of the fourth university. Because of the different test set sizes for each of the four results, we used micro-averaging for evaluating the accuracy of the predictors, i.e., we lumped the predictions from all four runs together and computed an accuracy measure on the entire set of predictions.

More details on the experimental setup can be found in [8], while the dataset is described in [4].

6 Results

6.1 Page Accuracy

Table 1 shows the accuracies measured for predicting the page labels. The rows list the different representation schemes, starting from the default prediction accuracy (using no features), to the classifier that uses all features. The columns of the table give the accuracy for each of the 5 implemented prediction combination techniques, starting with the voting scheme including default prediction, voting without default prediction, normalized weighted average, weighted average, and finally the maximum method (see section 4).

In terms of representation, it becomes apparent that using additional feature sets will generally result in higher accuracies. The exception to the rule is the *Paragraph* feature set. Whenever its features are added to a representation that

Table 1. Accuracies for classifying the 1050 pages using various methods for combining link predictors to page predictors.

Classifier	Combination Method				
	Vote (all)	Vote	Normal	Weight	Max
Default	51.81	51.81	51.81	51.81	51.81
Anchor	67.52	74.67	74.38	74.19	**74.76**
Headings	60.48	72.29	72.38	**72.95**	**72.95**
Paragraph	63.05	66.86	66.86	**66.95**	66.29
Anchor+Headings	74.48	85.33	84.95	85.14	**86.57**
Anchor+Paragraph	68.00	74.29	74.00	73.90	**74.67**
Headings+Paragraph	70.48	79.90	80.19	81.14	**81.33**
All	74.19	82.29	81.71	82.67	**83.24**

already includes the *Anchor* features, the result is a loss of predictive accuracy. A reason for this might be that these two feature sets are much less independent of each other than other pairs of feature sets.[4] The best results were achieved when relying only on the anchor text and the information from the headings.

Among the five different techniques for combining the link predictions to a page prediction, taking the prediction with the maximum confidence is a clear winner. In 7 out of 8 runs, using this method gave the best results (shown in **bold** face). However, in general, the differences among the combination methods are not nearly as large as the differences among the different document representations. The only exception is the voting scheme that also allowed the default rule to vote (first column of table 1). Apparently, the learned rules have a fairly low coverage, so that many of the links have to be classified using the default rule. It happens quite frequently that a few good rules are outnumbered by a number of default predictions. We have also found that the performance deteriorates similarly when default predictions are included into the weighted prediction combiner (results not shown). The maximum technique remains mostly unaffected by this because it is unlikely that a default prediction receives the maximum confidence score among a number of competing link predictions (results also not shown).

6.2 Link Accuracy

One question that remains unanswered by table 1 is how much has actually been gained by combining the prediction of different links pointing to a single page. To investigate this question, we computed a weighted accuracy estimate by weighting each page with the number of links that point to that page. In

[4] Note, however, that even though the set of words occurring on the anchor text is a subset of the set of words occurring in its surrounding paragraph, the resulting *Anchor* feature set is *not* a subset of the resulting *Paragraph* feature set because the feature x`_occurs_in_anchor_text` is semantically different from the feature x`_occurs_in_paragraph`, the former being more specific than the latter.

Table 2. Accuracies for classifying the 5803 links with various predictor combination methods.

Encoding	Combination Method					
	No	Vote(all)	Vote	Normal	Weight	Max
Default	36.67	36.67	36.67	36.67	36.67	36.67
Anchor	57.92	58.80	75.93	75.56	75.37	**76.05**
Headings	43.34	40.01	66.62	69.89	**70.77**	64.33
Paragraph	53.40	55.09	65.91	65.81	**66.33**	58.59
Anchor+Headings	62.49	61.66	86.18	85.46	**86.25**	83.22
Anchor+Paragraph	58.40	59.23	**73.70**	73.67	73.46	71.81
Headings+Paragraph	58.50	56.69	78.67	78.98	**80.30**	76.63
All	57.99	61.43	79.15	77.74	**79.44**	79.20

other words, all links that point to the same page perform an internal vote to decide upon a common classification for the page they point to. Each link of such a group is then classified with this common label. The resulting accuracy estimate counts the number of correctly predicted links over all links, and can thus be directly compared to the accuracies of the base classifiers that predict the class labels of each link independently.

These results are shown in table 2. The first thing to note is a substantial difference between the independent classifier (first column) and the classifiers that rely on combining the predictions for different links for all methods except voting with inclusion of default predictions. Obviously, many mistakes could be corrected by combining the predictions of different links and thus being able to rely on good features that appeared in a different link pointing to the same page.

Secondly, the differences between the voting scheme that includes default predictions (second column) and the voting schemes that ignore them is more remarkable than in table 1. We explain this with the fact that for pages with many incoming links, there are good chances that many of the links are classified by default rules, and that the combination of these predictions overrides the few "educated" guesses. With the voting schemes that ignore default predictions, the situation is the opposite: A few correct rule-base classifications can override many wrong default classifications and thus gain substantially in accuracy.

It is also interesting to observe that in table 2, the *Max* prediction method (last column) is not as dominant as in table 1 and, in some cases, it performs substantially worse than its competitors. The reason for this is that the maximum prediction is much less susceptible to variations in the number of link predictions that are combined to a single page prediction. If an erroneous link prediction has the maximum confidence score, it is used for predicting the class of the page. The voting and weighting methods, on the other hand, can make use of a number of unanimous predictions with lower confidence scores to override a prediction with a higher confidence score. Thus, it can be expected that pages with a higher number of incoming links are classified more reliably by voting or weighting,

Table 3. Recall and Precision for the page predictors. *Recall* is the percentage of pages that are not classified with the default rule and *Precision* is how many of these classifications were correct.

Classifier	Recall	Precision			
		Vote	Normal	Weight	Max
Anchor	40.76	83.64	82.30	82.48	83.88
Headings	74.10	88.05	88.19	88.95	88.95
Paragraph	46.67	75.71	75.91	75.92	74.49
Anchor+Headings	85.90	92.35	91.91	92.13	93.79
Anchor+Paragraph	60.19	83.23	82.81	82.59	83.86
Headings+Paragraph	82.38	88.44	88.79	89.94	90.17
All	78.00	86.94	86.20	87.42	88.16

while pages with a lower number incoming links are better classified by taking the prediction with the maximum score. As the latter category is more frequent, the page accuracies tend to be higher for the maximum prediction method, while the link accuracies tend to be higher for the voting and weighting schemes.

6.3 Recall and Precision

We have discussed above that many of the test examples are classified using the default rule and that it seems to be advisable to ignore these default link predictions for computing the page predictions. But what happens in cases where *all* links that point to a page are classified by default rules, i.e., no link contains any information that could be used for a justified prediction? In the experiments reported in the previous section, we have simply predicted the majority class *Student* for each of these pages. What if we ignore these predictions? It can be expected that the classification accuracy goes up at the expense of classifying fewer pages. This trade-off is commonly measured in terms of precision and recall.

Table 3 lists recall and precision estimates that shed some light upon this question. *Recall* is the percentage of pages which were classified using at least one rule different from the default rule. Note that this estimate is the same for all combination methods because the underlying link classifiers are the same and hence the links that are classified with default rules are the same. *Precision* is the percentage of classified pages that were correctly classified. In general, the precision scores are much higher than the accuracy results of table 1. This is not surprising because accuracy can be viewed as a weighted sum between the precision on the recalled examples and the precision on the examples classified by default rules (which should be about the default accuracy, although the variance can be very high). The recall scores are more differentiated. *Headings* features not only have the highest, but also achieve this precision at significantly higher recall scores.

Table 4. Accuracy and number of features for using feature subset selection on the full-text classifier.

Classifier	# Features	Accuracy
Link-Based	8,075	85.05%
Full-Text	20,322	70.67%

Table 5. Accuracy results for feature subset selection on the full-text classifier.

# Features	Accuracy
100%	70.67
50%	73.90
10%	74.19
5%	74.76
1%	71.33
0.1%	54.67

6.4 Comparison to Full-Text Classifier

We also compared the predictive accuracies of the link-based page classifiers to those of a classifier that uses the words occurring on the page as a feature. Table 4 shows a comparison between the link-based classifier using all four feature sets and a full-text classifier in terms of predictive accuracy and the number of features used by both representations.[5] The link-based classifiers discussed in this paper are considerably more accurate while using less than half of the number of features.

An obvious question at this point is, of course, whether the differences in accuracy are at least partly due to the different number of features. Could we produce a similar effect by employing feature subset selection on the full-text classifier? We have already seen, that in general, adding additional feature sets improves accuracy. On the other hand, it is also known that too many features can lead to overfitting. Table 5 shows the results for using only the top n% of the features of the full-text classifier (selected by entropy). Feature subset selection results in some improvement, but the best result is still more than 10% behind the link-based classifiers. We have not checked whether feature subset selection would improve the link-classifiers as well.

7 Conclusion

Our results show that it is possible to classify documents more reliably with information originating from pages that point to the document than with features that are derived from the document text itself. Furthermore, it proved to be beneficial to be able to exploit redundant information on the WWW by combining multiple predictions (one for each hyperlink pointing to a page). However, we have shown this for one domain only, so our results can only be considered as preliminary. More experimental work in other domains must be conducted in order to establish a conclusive result.

[5] The reported number of features is the total number of features in the entire dataset. Each of the four training sets of the cross-validation contained on average a little more than 80% of the features for both types of classifiers.

Although the encoding scheme we used is quite straight-forward, it illustrates that the use of information about the HTML structure of pages and about the structure of the WWW itself can be useful for improving text categorization on the WWW. The use of more elaborate representation schemes (e.g., distinguishing different types of headings or even using an entire HTML-tree [6] as background knowledge) suggests itself as a rewarding topic for further research as does the use of relational learning techniques (see, e.g., [3,10]). We have already performed preliminary experiments using linguistic phrases of the kind used in [9] as an additional feature set, but found that they did not make much difference [8]. Such approaches also need to be investigated in more detail.

Acknowledgements

This work was performed during the author's stay at Carnegie Mellon University, which was enabled by a *Schrödinger-Stipendium* (J1443-INF) of the Austrian *Fonds zur Förderung der wissenschaftlichen Forschung (FWF)*. The author has benefited greatly from discussions with Mark Craven, Tom Mitchell, and Ellen Riloff. Thanks to the CMU text learning group for making the WebKB dataset available, in particular to Dayne Freitag and Tom Mitchell for providing a preprocessed version.

References

1. William W. Cohen. Fast effective rule induction. In A. Prieditis and S. Russell, editors, *Proceedings of the 12th International Conference on Machine Learning (ML-95)*, pages 115–123, Lake Tahoe, CA, 1995. Morgan Kaufmann.
2. William W. Cohen. Learning trees and rules with set-valued features. In *Proceedings of the 13th National Conference on Artificial Intelligene (AAAI-96)*, pages 709–716. AAAI Press, 1996.
3. Mark Craven. Using statistical and relational methods to characterize hyperlink paths. In D. Jensen and H. Goldberg, editors, *Artificial Intelligence and Link Analysis: Papers from the 1998 AAAI Fall Symposium*, pages 14–20, Orlando, Florida, 1998. AAAI Press. Technical Report FS-98-01.
4. Mark Craven, Dan DiPasquo, Dayne Freitag, Andrew McCallum, Tom Mitchell, Kamal Nigam, and Seán Slattery. Learning to extract symbolic knowledge from the World Wide Web. In *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98)*. AAAI Press, 1998.
5. Thomas G. Dietterich. Machine learning research: Four current directions. *AI Magazine*, 18(4):97–136, 1997.
6. Dan DiPasquo. Using HTML structure to aid in automatic information retrieval from the world wide web. Senior Honors Thesis, School of Computer Science, Carnegie Mellon University, May 1998.
7. Johannes Fürnkranz. Pruning algorithms for rule learning. *Machine Learning*, 27(2):139–171, 1997.
8. Johannes Fürnkranz. Using links for classifying web-pages. Technical Report OEFAI-TR-98-29, Austrian Research Institute for Artificial Intelligence, 1998.

9. Johannes Fürnkranz, Tom Mitchell, and Ellen Riloff. A case study in using linguistic phrases for text categorization on the WWW. In M. Sahami, editor, *Learning for Text Categorization: Proceedings of the 1998 AAAI/ICML Workshop*, pages 5–12, Madison, WI, 1998. AAAI Press. Technical Report WS-98-05.
10. Seán Slattery and Mark Craven. Combining statistical and relational methods for learning in hypertext domains. In D. Page, editor, *Proceedings of the 8th International Conference on Inductive Logic Programming (ILP-98)*, pages 38–52, Madison, WI, 1998. Springer-Verlag.
11. Ellen Spertus. ParaSite: Mining structural information on the Web. *Computer Networks and ISDN Systems*, 29(8-13):1205–1215, September 1997.