

# Using Links for Classifying Web-pages

**Johannes Fürnkranz**

Austrian Research Institute for Artificial Intelligence

Schottengasse 3, A-1010 Wien, Austria

E-mail: [juffi@ai.univie.ac.at](mailto:juffi@ai.univie.ac.at)

Technical Report OEFAI-TR-98-29

## Abstract

In this paper, we report on a systematic set of experiments that explore the utility of making use 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. We present a set of experiments that confirm this hypothesis on a set of Web-pages that relate to Computer Science Departments.

## 1 Introduction

The advent of the World-Wide Web has rejuvenated 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 documents, such as their heading and their 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 systematic set of experiments that explore the utility of making use 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.

**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 to 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.<sup>1</sup>

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

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 (linguistic) context in which the anchor text appears*
3. *the headings that structurally precede the section of the document in which the link occurs*

---

<sup>1</sup>A brief description of this domain can be found in section 4.1

For example, department pages typically have a large number of links pointing to them that are marked with anchor texts like “computer science department”, “CS department”, “dept. of computer science”, or 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. As student home pages are usually contain many links, it is quite likely that there are several links occurring near the word “advisor”. Thus it is reasonable to assume that the use of linguistic information could help to further focus the learner. For example, being able to recognize a sentence fragment like “my advisor is < . . . >” makes the learner’s task trivial.

However, not all pages contain grammatical text. For example, 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 that type of information that is listed below. Clearly, this information can also be very useful for classifying the pages that are pointed to in the list below this heading.

### 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.<sup>2</sup> 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).

Throwing the words of all headings together is a crude approximation because headings further away could contain misleading information (e.g., department pages that are structured into sections with informations about various aspects of departments). Our motivation for using all headings, however, was that it is not always the heading immediately preceding the link that contains the relevant information. Future work might weight the headings differently according to their structural distance to the hyperlink.

---

<sup>2</sup>We did this by scanning our collection of pages of Computer Science departments for all occurrences of an HREF that contains the address of the current page but 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.

**Paragraph:** All words of the paragraph in which the hyperlink occurs. Our method for determining the paragraph is somewhat heuristic: Pieces of text separated by <P> or an empty line are paragraphs, as are structural entities such as items in a list <LI>. Our straightforward method for grabbing the paragraphs is not perfect. E.g., it is hard to determine what to do with the break marker <BR>, which is sometimes used as a simple line break, occasionally even within a sentence, but sometimes also for generating artificial list environments. As one of our goals for the paragraph grabber was to provide a reasonable text fragment for phrase extraction via AUTOSLOG (see below), we tried to resolve such issues conservatively.

**Phrases:** We used the text fragment extracted for the paragraph information to generate phrasal features of the sort described in [6]. AUTOSLOG-TS, originally conceived for information extraction [8, 9], is able to provide a learner with features that capture some of the syntactic structure of natural language text. We used a version of AUTOSLOG-TS that uses hyperlinks as a target<sup>3</sup> and returns patterns that capture the syntactic role of the prepositional phrase on the hyperlink. Consider for example the sentence “*My advisor is* <A HREF=“http://...”>*Tom Mitchell*</A>.”. From the above input sentence, AUTOSLOG-TS would extract the feature *advisor\_is\_<\_>* that would match all pages in which a student uses this phrase to refer to his advisor’s home-page. Note that this type of feature is able to provide a better focus than using simply the word *advisor* in the paragraph feature, in particular in cases when there is more than one hyperlink in the paragraph.

**Class:** Each link was labeled with the class of the page it points to.

The four features described above were each encoded as a separate set-valued feature [2] for the separate-and-conquer rule learner [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 refer to the set-valued feature as a *feature set* and use the terms *feature* and *word* (or *phrase* for the *Phrases* feature set) interchangeably.

## 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<sup>4</sup> 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.

---

<sup>3</sup>The main difference between the version used in [6] and the version we used in these experiments is that the former extracts *all* syntactic patterns from a text, while the latter specifically targets phrases around hyperlinks.

<sup>4</sup>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.

Quite frequently, however, several links point to the same page. As our goal is to 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 devise 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:

**Majority Vote:** 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. We implemented two versions of this voting scheme: In the first version, links that were 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) were eligible to vote, while in the second version the votes of such links were ignored. If a page only received votes from the default rules, it was classified with the majority class, *Student*.

**Weighted Average:** We also associated a confidence score with each of RIPPER’s predictions, which simply consisted 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 originated from a default rule, it was assigned a score of 0. Such a score was computed for each possible class of each link. Again, we implemented two different versions. In one version, the confidence scores of each class are added up over all links that point to the page to classify. The other version does the same, but first normalizes the confidence scores in a way that distributes a total weight of 1 among the different candidate classes for each link. In both cases, we predict the class receiving the maximum score after summation over all links pointing to the same page.

**Maximum Confidence:** The third 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.

## 4.1 Experimental Setup

The dataset we used consists of 1050 pages that are part of a data set that was collected for the WebKB project [4]. The pages were manually 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 was turned into a separate training example using the feature sets described in section 3. Table 1 shows the class distribution of the dataset for the 1050 pages and for the 5803 hyperlinks. The classes have very different characteristics. First of all, they differ considerably in their sizes. Second, the average number of links that point to a page varies. For example, there are only four home pages of computer science departments in the entire set, which would be too few data to apply a straight-forward text classification algorithm. However, there are more than 100 links pointing to each of these pages, which makes *Department* a class that can be learned fairly well.

Class	Links		Pages	
Student	2128	36.67%	544	51.81%
Faculty	1374	23.68%	147	14.00%
Course	915	15.77%	229	21.81%
Project	701	12.08%	83	7.90%
Department	530	9.13%	4	0.38%
Associate	121	2.09%	30	2.86%
Post Doc	34	0.59%	13	1.24%

Table 1: Class distribution of the data in links and pages.

The pages/links were collected from four universities, Cornell (243/1073), Texas (253/1264), Washington (256/1771), and Wisconsin (298/1695). We removed all HTML markers and sentence marks from the text and converted all words to lower case thereby replacing special characters within a word with ‘\_’ and all digits with the letter ‘D’.

As elaborated above, we encoded each link using 4 different set-valued features. As our main goal was to evaluate the utility of each of the four representations, we ran RIPPER on all possible combinations of the available feature sets, i.e., we first ran experiments using only single feature sets, then all pairs of feature sets, all triplets, and finally using all four feature sets.

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 all predictions from all four runs together and computed an accuracy measure on the entire set of predictions.

## 5 Results

### 5.1 Page Accuracy

Table 2 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 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

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	<b>74.76</b>
Headings	60.48	72.29	72.38	<b>72.95</b>	<b>72.95</b>
Paragraph	63.05	66.86	66.86	<b>66.95</b>	66.29
Phrases	51.90	56.29	56.29	56.38	<b>56.57</b>
Anchor+Headings	74.48	85.33	84.95	85.14	<b>86.57</b>
Anchor+Paragraph	68.00	74.29	74.00	73.90	<b>74.67</b>
Anchor+Phrases	68.86	74.95	74.76	74.48	<b>75.43</b>
Headings+Paragraph	70.48	79.90	80.19	81.14	<b>81.33</b>
Headings+Phrases	62.10	<b>77.33</b>	76.86	76.95	77.14
Paragraph+Phrases	62.57	66.86	<b>67.05</b>	66.86	66.10
w/o Anchor	68.00	76.57	76.76	76.48	<b>77.05</b>
w/o Headings	67.81	73.90	74.10	74.19	<b>74.38</b>
w/o Paragraph	74.29	85.52	84.86	85.43	<b>86.86</b>
w/o Phrases	74.19	82.29	81.71	82.67	<b>83.24</b>
All	75.33	84.54	84.57	<b>85.05</b>	<b>85.05</b>

Table 2: Accuracies for classifying the 1050 pages using various methods for combining link predictors to page predictors.

independent of each other than other pairs of feature sets.<sup>5</sup> The features in the *Phrases* feature set are used quite infrequently and do not make a big difference when added to a feature set. However, it should be noted that the highest accuracy (86.86%) was achieved when using all but the *Paragraph* feature (and combining the predictions of the links by choosing the one with maximum confidence) although the differences between this version and the version that uses only *Anchor* and *Headings* are inconclusive.

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 12 out of the 15 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 2). 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). However, the maximum technique remains mostly

---

<sup>5</sup>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.

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	<b>76.05</b>
Headings	43.34	40.01	66.62	69.89	<b>70.77</b>	64.33
Paragraph	53.40	55.09	65.91	65.81	<b>66.33</b>	58.59
Phrases	39.12	37.03	56.38	56.38	56.47	<b>56.83</b>
Anchor+Headings	62.49	61.66	86.18	85.46	<b>86.25</b>	83.22
Anchor+Paragraph	58.40	59.23	<b>73.70</b>	73.67	73.46	71.81
Anchor+Phrases	58.43	59.64	74.39	74.10	73.79	<b>75.53</b>
Headings+Paragraph	58.50	56.69	78.67	78.98	<b>80.30</b>	76.63
Headings+Phrases	45.03	40.19	76.77	75.50	76.10	<b>79.63</b>
Paragraph+Phrases	51.84	49.96	63.52	<b>63.76</b>	63.45	60.18
w/o Anchor	55.56	52.49	78.01	<b>78.25</b>	75.87	76.55
w/o Headings	55.85	58.87	71.19	<b>71.62</b>	69.95	65.95
w/o Paragraph	59.80	61.26	<b>83.68</b>	82.22	83.65	82.65
w/o Phrases	57.99	61.43	79.15	77.74	<b>79.44</b>	79.20
All	63.11	62.21	82.35	82.68	<b>83.68</b>	79.27

Table 3: Accuracies for classifying the 5803 links with various predictor combination methods. Except for the column labelled *No*, the prediction in each column were generated by assigning to each link the classification of the page it points to, which was derived as in table 2.

unaffected because it is unlikely that a default prediction receives the maximum confidence score among a number of competing link predictions (results also not shown).

## 5.2 Link Accuracy

One question that remains unanswered by table 2 is how much has actually been gained by combining the prediction of different links pointing to a single page. To answer this question, we computed a weighted accuracy estimate by weighting each page with the number of links that point to that page. This can be thought of as classifying each link with the class that was eventually assigned to the page it points to, no matter whether the classifier predicted this class for this particular link. In other words, all links that point to the same page perform an internal vote to decide what their vote should be. Then, they all vote in the same way. 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 label of each link independently.

These results are shown in table 3. 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 2. 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-based classifications can override many wrong default classifications and thus gain substantially in accuracy.

It is also interesting to observe that in table 3, the *Max* prediction method (last column) is not as dominant as in table 2 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, 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.

### 5.3 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 4 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 2. 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).

More informative are the recall results. For example, phrasal features only have a recall of slightly more 10%. This means that only very few links are classified with a reasonable rule. Adding these features to other feature sets usually does not increase recall but small increases

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
Phrases	10.57	70.27	66.94	71.17	72.97
Anchor+Headings	85.90	92.35	91.91	92.13	93.79
Anchor+Paragraph	60.19	83.23	82.81	82.59	83.86
Anchor+Phrases	42.38	82.92	81.68	81.80	84.04
Headings+Paragraph	82.38	88.44	88.79	89.94	90.17
Headings+Phrases	74.48	92.20	91.58	91.69	91.94
Paragraph+Phrases	46.95	78.09	78.43	78.09	76.47
w/o Anchor	65.71	86.67	86.98	86.52	87.39
w/o Headings	57.71	83.17	83.39	83.66	83.99
w/o Paragraph	86.00	92.14	91.36	92.03	93.69
w/o Phrases	78.00	86.94	86.20	87.42	88.16
All	87.05	91.03	91.14	91.68	91.68

Table 4: 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	# Features	Accuracy
Link-Based Classifier	8,075	85.05%
Full-Text Classifier	20,322	70.67%

Table 5: Accuracy and number of features for using feature subset selection on the full-text classifier.

in accuracy are noticeable. This would be consistent with the results of [6] where it was shown that such features can increase precision. The low recall of *Paragraph* is somewhat surprising, as one could expect that the high number of features should be able to achieve a higher coverage. However, a higher number of features also increases the danger of overfitting, which, in our opinion, is what happens with the *Paragraph* feature sets. *Headings* features have by far the highest recall. We explain this with the fact that the words occurring in headings tend to be more repetitious.

## 5.4 Comparison to full-text classifier

Another obvious question that should be answered is a comparison between 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 5 shows a comparison between the link-based classifier using all four

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

Table 6: Accuracy results for using feature subset selection on the full-text classifier.

feature sets and a full-text classifier in terms of predictive accuracy and the number of features used by both representations.<sup>6</sup> 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 6 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.

## 6 Conclusion

In this paper, we have investigated the use of information on pages with hyperlinks pointing to a certain page to classify that page. This approach allows to exploit different information sources, typically provided by different authors, and thus effectively reduces the dependence on the vocabulary choice of the authors.

We represented each hyperlink with its anchor text, the headings structurally preceding it, the text of the paragraph in which it occurs, and a set of linguistic phrases the capture the syntactic role of the anchor text in this paragraph. A rule-based classifier was trained to predict the class labels of individual links. The predictions of links pointing to the same page were then combined to yield a prediction for this page. The best results were achieved when relying on the anchor text, the headings, and the phrase information, when combining the link predictions by selecting the prediction with the maximum confidence. Phrases seem to provide a better focus than using the paragraphs only, but they do not seem to be as useful as in previous work [6], the main problem being their low coverage.

Although the encoding scheme we used is rather simple, it illustrates that the use of information about the HTML structure of pages and about the structure of the Web itself can be useful

---

<sup>6</sup>The reported number of features is the total number of features in the entire dataset. Each of the four training sets contained on average a little more than 80% of the features for both types of classifiers.

for improving text categorization on the Web. The use of more elaborate representation schemes (e.g., including information about the type of headings or even using an entire HTML-tree as in [5]) suggests itself as a rewarding topic for further research as does the use of relational learning techniques, such as those introduced in [3, 10], which do not exploit the redundancy provided by multiple incoming links but, instead, are able to use hyperlink paths of arbitrary length.

## 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. Mark Schmelzenbach has implemented the described modifications of AUTOSLOG-TS.

## 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 Intelligence (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] 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.
- [6] 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.
- [7] Johannes Fürnkranz and Gerhard Widmer. Incremental Reduced Error Pruning. In W. Cohen and H. Hirsh, editors, *Proceedings of the 11th International Conference on Machine Learning (ML-94)*, pages 70–77, New Brunswick, NJ, 1994. Morgan Kaufmann.

- [8] Ellen Riloff. Automatically generating extraction patterns from untagged text. In *Proceedings of the 13th National Conference on Artificial Intelligence (AAAI-96)*, pages 1044–1049. AAAI Press, 1996.
- [9] Ellen Riloff. An empirical study of automated dictionary construction for information extraction in three domains. *Artificial Intelligence*, 85:101–134, 1996.
- [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.