Discovering Test Set Regularities in Relational Domains

Seán Slattery SEAN.SLATTERY@CS.CMU.EDU
Tom Mitchell TOM.MITCHELL@CS.CMU.EDU
Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213, USA

Abstract

Machine learning typically involves discovering regularities in a training set, then applying these learned regularities to classify objects in a test set. In this paper we present an approach to discovering additional regularities in the *test* set, and show that in relational domains such test set regularities can be used to improve classification accuracy beyond that achieved using the training set alone. For example, we have previously shown how FOIL, a relational learner, can learn to classify Web pages by discovering training set regularities in the words occurring on target pages, and on other pages related by hyperlinks. Here we show how the classification accuracy of FOIL on this task can be improved by discovering additional regularities on the test set pages that must be classified. Our approach can be seen as an extension to Kleinberg's Hubs and Authorities algorithm that analyzes hyperlink relations among Web pages. We present evidence that this new algorithm leads to better test set precision and recall on three binary Web classification tasks where the test set Web pages are taken from different Web sites than the training set.

1. Introduction

The standard paradigm in machine learning involves learning regularities over a training data set in order to classify objects in a test set. We present here an approach to discovering relational regularities in the *test set*, to augment the regularities learned from the training set, and thereby improve classification accuracy. This work was initially motivated by our research in the Web→KB project (Craven et al., 1998a), which seeks to automatically extract useful information from the Web. One key problem there is to automatically classify Web pages (e.g., as student home pages, research project pages, etc.). In this paper we present experimental results showing that discovering test set regularities does indeed improve accuracy for this task of automatically classifying Web pages. We believe the approach presented here will also be relevant to machine learning problems in various other domains involving relational data, especially when the training and test sets include non-overlapping portions of the total relational graph.

While most approaches to classifying text documents base the classification on the content of the document (e.g., the collection of words it contains) (Yang & Pedersen, 1997; Lang, 1995; Dumais et al., 1998; Lewis et al., 1996), hypertext documents such as Web pages contain another source of information — the relationships between documents encoded as hyperlinks between pages. In previous work (Craven et al., 1998b; Slattery & Craven, 1998) we showed that a FOIL-like relational learner that takes advantage of such hyperlink relations in the training data can achieve better classification accuracy than methods that use only the words on the Web page itself.

A natural complement to this earlier work which learned relational descriptions from the training set, is to consider discovering relational regularities over the test set, then use these to improve classification accuracy over the test set. This paper suggests one particular class of test set regularities, shows how instances of this class of regularities can be discovered in the test set, and shows that these can improve classification accuracy. We then discuss how this approach is related to other recent research on using hyperlink structure and unlabeled data, other relational domains in which this approach may be useful, and our initial thoughts on other forms of test set regularities that may be useful.

2. Motivation

Exactly what kinds of useful regularities might we be able to learn over a test set of relational Web data? To illustrate, consider the task of learning to classify Web pages according to whether or not they are home pages of university students. Suppose that we have already trained a model for this classification task, using a set of labeled Web data.

Now suppose we are to apply this learned model to a new Web site (the test data), part of which is shown in Figure 1. As shown in this figure, the learned model classifies two of

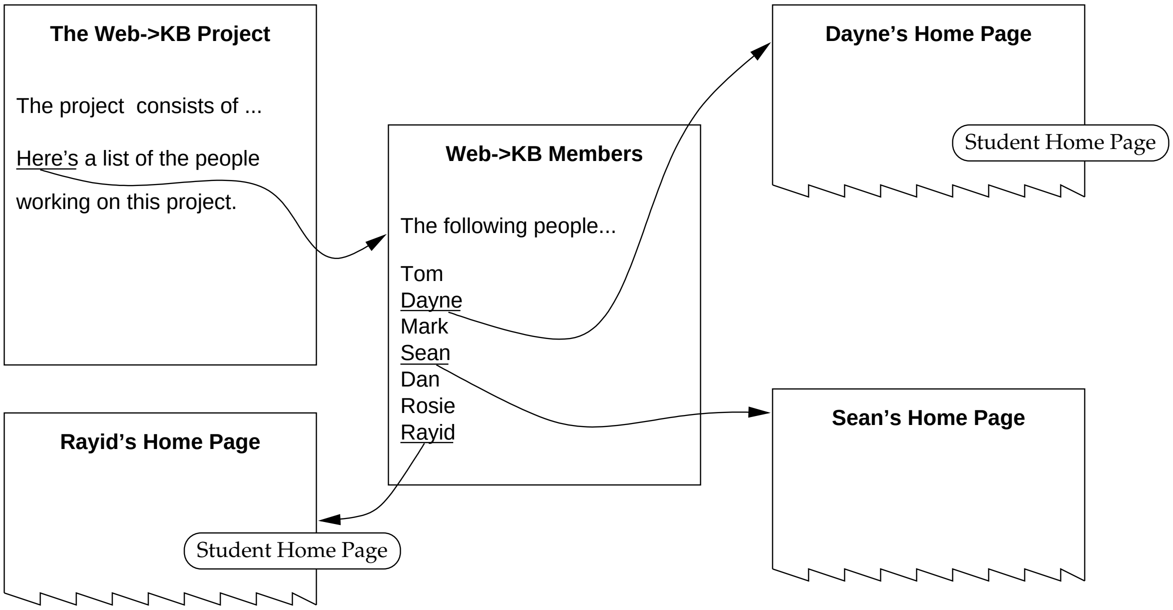

Figure 1. Web pages from a test set in which we wish to identify student home pages. The model learned from the training set has classified Dayne's Home Page and Rayid's Home Page as student home pages. This model has missed that Sean's Home Page is also a student home page.

these test set pages as student home pages ("Rayid's home page," and "Dayne's home page"). As humans examining this new Web site, we might therefore guess that "Sean's home page" is also a student home page. We humans might make this inference because Rayid's and Dayne's pages appear on a list together, both seem to be student home pages, and Sean's page appears on the same list. In fact, there is support in this test set (two out of three examples known to be correct) for the regularity:

> If page P is pointed to from the list on the "Web→KB Members" page, then P is a student home page.

In this paper, pages such as the "Web→KB Members" page will be referred to as *hubs*.

Naturally the above rule is not guaranteed to be correct. In fact Tom's page is not in reality a student home page. However, given the way information tends to be organised on the Web, this kind of observed regularity may often produce useful predictions, and it may therefore be useful to search for empirical regularities of this form within the test data. We propose below an iterative algorithm that searches for test set regularities of a similar form.

Notice one important characteristic of the above regularity: it could not possibly be learned from any amount of training data (assuming the training data is disjoint from the test set). This is because the above regularity refers directly to the constant "Web→KB Members" which occurs only in the test set. Regularities that depend on such test set constants are outside the scope of what can be learned from a disjoint training set. They can be learned only by searching for regularities in the test set.

Figure 2 shows an overall picture of the components of our approach. The relational training and test sets consist of objects and (in this case) binary hyperlink relationships between them. A model of the positive objects is learned from the training set and used to classify objects in the test set. Using the algorithm presented in Section 4.2, we then simultaneously find promising hubs (test set regularities that predict positive examples), and hub predictions (new positive examples in the test set).

3. Hubs and Authorities

Inspiration for the algorithm presented in this paper comes from an elegant information retrieval algorithm, also based on discovering relational regularities in a test set, introduced by Kleinberg (1998). Hubs and Authorities is a Web searching algorithm based on observations about how information is organised on the Web. Starting from the problem of searching for *authorities* on a particular topic, Kleinberg made the following observation:

> Hyperlinks encode a considerable amount of latent human judgement, and we claim that this

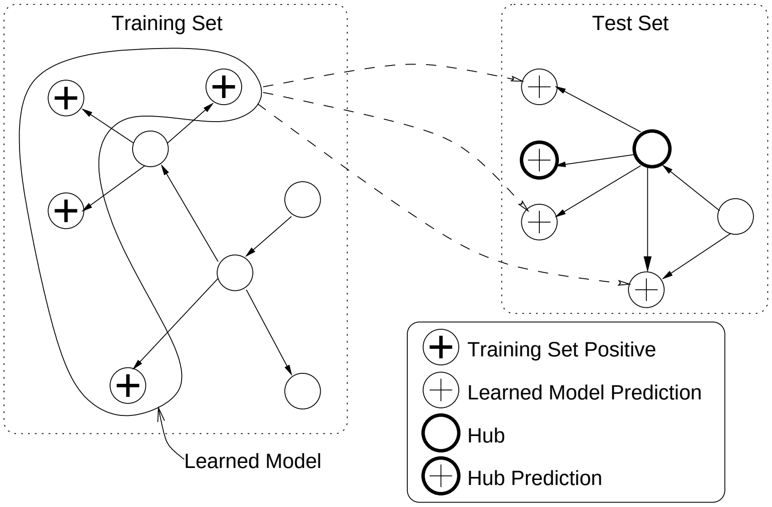

Figure 2. Illustration of how a learned model and the structure of a relational test set can be used to find more positive predictions using hubs discovered in the test set.

> type of judgement is precisely what is needed to formulate a notion of authority. Specifically, the creation of a link on the WWW represents a concrete indication of the following type of judgement: the creator of page *p*, by including a link to page *q*, has in some measure *conferred authority* on *q*. Moreover, links afford us the opportunity to find potential authorities purely through the pages that point to them; this offers a way to circumvent the problem ... that many prominent pages are not sufficiently self-descriptive.

To actually find the relevant authorities for a given Web search, Kleinberg introduces the notion of *hub pages* which point to multiple relevant authorities. Noting that

> Hubs and authorities exhibit what could be called a *mutually reinforcing relationship*: a good *hub* is a page that points to many good authorities; a good *authority* is a page that is pointed to by many good hubs.

he presents a simple algorithm for finding relevant hub pages and authority pages in a given set of linked Web pages. Relating this back to our motivating example, the "Web→KB Members Page" would be a hub page and each of Sean's, Rayid's and Dayne's pages would be authority pages.

The core of the algorithm involves separate hub (h) and authority (a) weights for each page being considering and iterating the following weight update rules until convergence:

$$h(p) \quad = \sum_{q\text{: link from } p \text{ to } q} a(q)$$

$$a(p) \quad = \sum_{q\text{: link from } q \text{ to } p} h(q)$$

Section 4.2 will present an algorithm for Web page classification that is an extension of an existing relational learner to take advantage of Hubs and Authorities style inference.

4. Algorithms

The experiments presented in Section 5 compare the performance of two algorithms on a set of Web page classification tasks. The first is an existing relational learner which can create models referencing the neighbourhood of a page, but cannot discover test set regularities.

The second is an extension to the first algorithm which uses the Hubs and Authorities idea to search for test set regularities and use these to improve classification accuracy.

4.1 FOIL

Previous work (Craven et al., 1998b) described how a relational learner such as FOIL (Quinlan, 1990; Quinlan & Cameron-Jones, 1993) could be used for Web page classification. In comparison with conventional text classification approaches, relational learners can naturally use the information contained in hyperlinked pages when constructing a model of the target class.

A very simple set of background relations is used to represent the data:

- link_to(Page, Page) This relation represents Web hyperlinks. For a given hyperlink, the first argument specifies the page in which the hyperlink is located and the second argument indicates the page to which the hyperlink points.
- has_*word*(Page) This set of relations indicates the words that occur on each page. There is one predicate for each word in the vocabulary and each instance indicates the occurrence of the word on the specified page.

Note that this representation does not use theory constants to represent words because doing so would require FOIL to add two literals to a clause for each word test, instead of a single literal as in this representation.

For plotting precision-recall curves, we require that each prediction made by FOIL have an associated confidence weight. We extended the FOIL algorithm so that it assigns a probability of correct classification to each learned rule. An m-estimate (Cestnik, 1990) is used to estimate this probability with $m = 2$. For each prediction, the highest m-estimate over all matching rules was used as the confidence weight. Predictions for test examples with no matching rule were given confidence weight 0.0.

4.2 FOIL-HUBS

FOIL-HUBS is an extension to FOIL to subsume the Hubs and Authorities algorithm. The extensions are simple — two recursive rules to encode the mutual dependence between hubs and authorities, and an iterative algorithm for using the learned rule set and the recursive rules to produce predictions.

4.2.1 RECURSIVE RULES

FOIL-HUBS first applies the FOIL algorithm to the training set to learn a rule set R that describes pages of class *class*:

```
class_page(A)  :-  has_foo(A), has_bar(A).
class_page(A)  :-  link_to(A,B), ...
```

We call these the *learned classifier rules*. The following two recursive rules are then added to the rule set:

```
    class_page(A) :- link_to(B,A), class_hub_page(B)
class_hub_page(A) :- link_to(A,B), class_page(B)
```

Together with the modified evaluation rules described in the next section, these new rules mimic the weight update rules presented in Section 3.

4.2.2 RULE EVALUATION

Rule evaluation in FOIL-HUBS differs fundamentally from FOIL. Instead of considering a target relation being true or false for a page, we assign a weight indicating how confident we are that a page is an example of the target relation, with larger weights indicating higher confidence.

We calculate weights for each rule type on each test example as follows:

Learned classifier rules Calculate confidence weights using the scheme described in Section 4.1.

Recursive rules Sum the confidence weights from each possible instantiation of the recursive rule. The weight of a single instantiation is simply the weight of the recursively referenced literal. So if student_hub_page(p12) currently has weight 0.7, then the weight we get from this instantiation of the recursive rule:

```
student_page(p47) :-
        link_to(p12,p47), student_hub_page(p12)
```

is also 0.7.

4.2.3 ITERATIVE ALGORITHM

An iterative relaxation algorithm is used by FOIL-HUBS to apply a rule set containing recursive rules to a test set of Web pages. Since we're combining two sources of information (from the learned classifier rules and the recursive rules), we must be careful about the importance assigned to each. The policy chosen here is that the learned classifier rules are the best source of information we have about the target class and so should be weighted more heavily than the recursive rules. Our learned classifier rules generally produce confidence weights in the range 0.5-1.0. This led us to scale the recursive rule weights so that they had a maximum value of 0.1.

The iterative algorithm is as follows:

1. Calculate the learned classifier weight for each page.
2. Iterate the following until convergence
   (a) For each page, calculate the *class*_hub_page weights for that page using only the recursive rules.
   (b) For each page, calculate the *class*_page weights for that page using only the recursive rules.
   (c) Scale the *class*_page weights so that the maximum weight is 0.1.
   (d) For each page add the learned classifier weight for that page to its *class*_page weight.
3. Report the *class*_page weight for a page as the confidence that that page is a positive example of class *class*.

Table 1. The number of Web pages in the University data set. The pages were collected from the Web sites of the Computer Science departments at Cornell University, University of Texas at Austin, University of Washington and the University of Wisconsin.

CLASS	CORNELL	TEXAS	WASH.	WISC.
COURSE	44	38	76	85
DEPARTMENT	1	1	1	1
FACULTY	34	46	31	42
PROJECT	20	20	21	25
STAFF	21	3	10	12
STUDENT	128	148	126	156
OTHER	619	570	940	946

4.2.4 OBSERVATIONS

We hope this algorithm will outperform FOIL in two ways. Firstly, since FOIL's predictions are inherently discrete, its predictions occur in clumps with the same confidence. FOIL-HUBS can use regularities in the test set to find highly probable positive test pages and assign them higher confidences.

Secondly, FOIL's coverage has been found to be low on these tasks because its rules are performing keyword presence tests. FOIL-HUBS has the potential to find promising-looking positive examples among the pages FOIL had no matching rule for, such as "Sean's Home Page" in the motivating example in Section 2

One last point worth mentioning is that the original Hubs and Authorities algorithm depended on the mutual reinforcement of many hubs and authorities for its information. It is less likely that such extensive structure exists for classification tasks. However adding in the learned classifier weight to the iterative algorithm may make up for the lack of such structure allowing "singleton" hubs to make a difference.

5. Experimental Evaluation

5.1 The University Data Set

Our data set for these experiments comes from the Web→KB project mentioned in the introduction. The first version of the system looked at university Web sites and the classification tasks involved finding student home pages, course home pages etc.

To train our classification algorithms, we collected pages from the Web sites of four computer science departments. This data set includes 4,127 pages and 10,945 hyperlinks interconnecting them. Each of the pages were labelled into one of seven classes, including a catch-all other class. Details of the distribution are given in Table 1.

5.2 Data Representation

The data is represented by the relations described in section 4.1. To produce the has_*word*(Page) relations, the Web pages were stripped of HTML markup, had stopwords removed and a stemming algorithm was applied to each word. For each remaining word that occurred more than 200 times in the training set, a corresponding has_*word* relation was created.[1] Depending on the training set, this procedure produced between 341 and 540 relations. The complete set of relations used are available from `http://www.cs.cmu.edu/~WebKB/ICML2000-data.html`.

5.3 Experimental Setup

We considered three binary classification tasks in the university data set: identifying student home pages, course home pages and faculty home pages. Using *leave-two-university-out* cross validation, we created all six possible train/test splits. For each split, we ran FOIL on the training data and applied the resulting rule set to the test data. The FOIL-HUBS algorithm was also applied to the test set.

5.4 Results

The precision and recall results for the target class on each classification task are shown in Figure 3. Precision and recall are defined as follows:

$$\begin{aligned} \text{Recall} &= \frac{\text{\# correct positive predictions}}{\text{\# of positive examples}} \\ \text{Precision} &= \frac{\text{\# correct positive predictions}}{\text{\# of positive predictions}} \end{aligned}$$

For each cross-validation run, the predictions for each algorithm were ordered according to the confidence they assigned the target class. Precision and recall were calculated at various confidence weight thresholds to get the tradeoff curve for that run. The curves were then averaged to get the final graphs presented here.

5.5 Discussion

Looking over all the precision-recall graphs we can see FOIL-HUBS performs better than FOIL for most confidence thresholds. The general form of the FOIL-HUBS curves is interesting. At low recall they show a marked improvement over the FOIL predictions. The improvement over FOIL falls away as recall increases. Then, as recall increases further, the improvement becomes more marked again.

[1] Standard information gain and related metrics are not good criteria for selecting words for a relational learner. Consider the case where the existence of the word *Guinness* on a linked page was a very useful feature for classifying the current page.

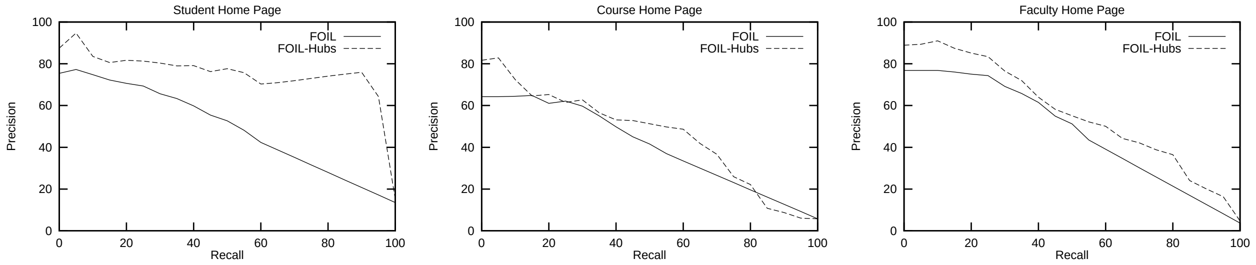

Figure 3. Recall-precision tradeoffs for each of the three binary classification problems. The graphs are the average of six cross-validation runs.

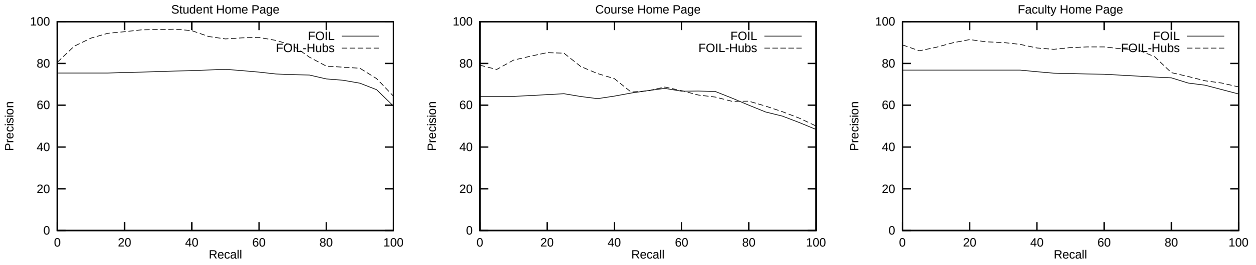

Figure 4. Recall precision tradeoffs for the test examples that matched some FOIL rule.

This behaviour stems from the way FOIL-HUBS uses the FOIL predictions. Effectively, FOIL-HUBS takes the examples FOIL made some predictions for and uses the relational regularities in the test set to produce a better ordering than the original FOIL one.

The improvement at higher recall is due to FOIL-HUBS taking the examples FOIL had no rules for, and using the test set regularities to find likely positive examples of the target concept among them. If we split the test data into those examples that matched some learned FOIL rule and those that matched none, and replot the precision-recall curves we see this effect more clearly (Figures 4 and 5).

This instantiation of the FOIL-HUBS algorithm leverages from finding useful *hub* pages in the test data. Craven et al. (1998b) noted the utility of learning a good description of the index page of graduate students (which occurs in each of our four universities) when learning to classify graduate student home pages. In the experiments presented in Craven et al. (1998b), the description of a graduate student index page generalised to cover the index page in the test set three times out of four.

FOIL-HUBS provides an independent route for leveraging off pages like the index page of graduate students. Table 2 shows the highest weighted hub pages for a particular run of the faculty home page classification task. It makes intuitive sense even from the titles of these pages, that they would contain hyperlinks to faculty home pages.

Table 2. Top five weighted hubs for the faculty home page class using pages from Washington and Wisconsin as training data and testing on pages from Cornell and Texas.

UTCS Faculty
Research Interests of the Faculty and Senior Researchers
UT Artificial Intelligence Laboratory
CS195T: Introduction to Graduate Computer Science
Faculty Research Interests

6. Related Work

Several other researchers have examined extensions to the original Hubs and Authorities algorithm. Bharat and Henzinger (1998) investigated extending it to take into account information about the content of Web pages, whereas the original algorithm treated pages as atomic features. Their algorithm scaled both the hub and authority weights by an Information Retrieval motivated similarity score computed between each page and the search query. They found that this approach improved precision by about 10% for a variety of search queries. Chakrabarti et al. (1998) define a *bridge* page which corresponds to the notion of a hub page discussed in this paper. They give evidence for bridge pages containing links to pages of the same topic, although they also discovered that links occurring closer together on a bridge page are even more likely to point to pages of similar topic than links occurring further apart. They also use

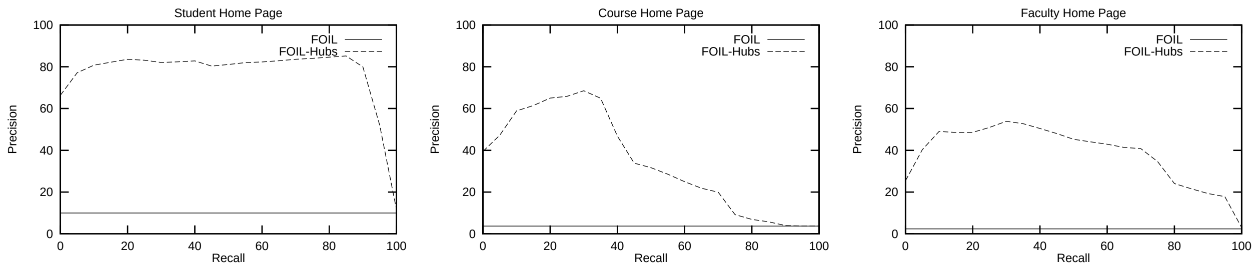

Figure 5. Recall precision tradeoffs for the test examples that matched no FOIL rule.

an iterative algorithm to label pages, although the approach collapses all the features into a single "engineered document" for learning and classification, essentially creating a propositional representation. Our work differs in that we focus both on document classification and on learning relational descriptions.

One way to view the approach we propose in here is as an attempt to use information in the unlabeled test set to improve classification accuracy. There has recently been significant interest in this topic within the machine learning research community. For example, Bennett and Demiriz (1998) discuss a transduction method that uses unlabelled test data together with labeled training data to improve accuracy in support vector machines. Nigam et al. (2000) discuss the application of EM to combining labeled and unlabeled data for naive Bayes text classifiers. Blum and Mitchell (1998) propose a cotraining method that uses unlabeled data to improve classification accuracy when the examples are described by redundantly sufficient features. All of these approaches to using unlabeled data make the assumption that the training and test data are drawn from the same underlying distribution, exhibit the same regularities, and that if we are given sufficient training data we could learn these regularities from the training data alone. In contrast, our approach was motivated by the fact that our training and test data tend come from different Web sites, that the detailed regularities inherent in each Web site are different, and that the statement of some of these regularities requires referring to specific constants in the first order descriptions of the data. In essence, this requires that we search for new regularities in the test data that cannot in principle be learned from the training data.

7. Future Directions

7.1 Other Relational Domains

While we have focused thus far on the problem domain of Web page classification, our approach appears to be relevant to a variety of relational domains. To see how this approach can generalize, notice in our Web domain we have instructed the FOIL-HUBS algorithm to search for test set regularities of the form

class_page(X) :- link_to(c,X)

where X is a variable, and c is a constant Web page (discovering hubs corresponds to choosing the constant c). In general, we believe user knowledge of the likely forms of test set regularities may be used in a similar way in other domains.

Consider, for example, a relational data set that describes people, their income, relatives, employers, etc., and assume the task is to learn to predict where each person lives. As in the Web domain, we may use our prior knowledge to instruct the system to search for test set regularities of the form

live_in(P, c1) :- employer_of(P, c2)

where c1 represents some specific constant city, and c2 a second constant employer. Given that some companies do employ only people in a single city, this is a reasonable form of regularity to search for. Furthermore, if the training and test sets are personnel databases from disjoint sets of employers, then we are again in a setting where the general form of regularities is known in advance, but the actual regularities can only be learned from the test set, just as in our Web setting.

7.2 Learning Regularity Forms

The recursive rules given in Section 4.2.1 can be intepreted as background knowledge given to FOIL-HUBS to enable it to search the test set for regularities. We would certainly expect the *form* of such a regularity to exist in the training set and could conceivably build an algorithm for finding the its form from the training set.

8. Conclusion

This paper has presented an approach to discovering regularities in the test set, and used these to improve classification accuracy. More specifically, we presented an extension to the FOIL algorithm, called FOIL-HUBS, and showed that it improves classification performance on three binary Web page classification tasks. Although we considered only Web classification tasks here, we believe similar uses of the test set may be helpful in other relational domains.

Acknowledgements

This research was supported in part by the DARPA HPKB program under contract F30602-97-1-0215.

References

Bennett, K., & Demiriz, A. (1998). Semi-supervised support vector machines. *Advances in Neural Information Processing Systems 11*. Denver: MIT Press.

Bharat, K., & Henzinger, M. R. (1998). Improved algorithms for topic distillation in a hyperlinked environment. *Proceedings of the Twenty-First Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 104–111).

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. *Proceedings of the Eleventh Annual Conference on Computational Learning Theory* (pp. 92–100). New York: ACM Press.

Cestnik, B. (1990). Estimating probabilities: A crucial task in machine learning. *Proceedings of the Ninth European Conference on Artificial Intelligence* (pp. 147–149). London: Pitman.

Chakrabarti, S., Dom, B., & Indyk, P. (1998). Enhanced hypertext categorization using hyperlinks. *Proceedings of ACM SIGMOD International Conference on Management of Data* (pp. 307–318). Seattle: ACM Press.

Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., & Slattery, S. (1998a). Learning to extract symbolic knowledge from the World Wide Web. *Proceedings of the Fifteenth National Conference on Artificial Intelligence*. Madison, WI: AAAI Press.

Craven, M., Slattery, S., & Nigam, K. (1998b). First-order learning for Web mining. *Proceedings of the Tenth European Conference on Machine Learning* (pp. 250–255). Chemnitz, Germany: Springer-Verlag.

Dumais, S. T., Platt, J., Heckerman, D., & Sahami, M. (1998). Inductive learning algorithms and representations for text categorization. *Proceedings of the Seventh International Conference on Information and Knowledge Management* (pp. 148–155). New York: ACM Press.

Kleinberg, J. (1998). Authoritative sources in a hyperlinked environment. *Proceedings of the Ninth ACM-SIAM Symposium on Discrete Algorithms* (pp. 25–27). San Francisco.

Lang, K. (1995). Newsweeder: Learning to filter netnews. *Proceedings of the Twelfth International Conference on Machine Learning* (pp. 331–339). Lake Tahoe: Morgan Kaufmann.

Lewis, D., Schapire, R. E., Callan, J. P., & Papka, R. (1996). Training algorithms for linear text classifiers. *Proceedings of the Ninteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 298–306).

Nigam, K., McCallum, A., Thrun, S., & Mitchell, T. (2000). Text classification from labeled and unlabeled documents using em. *Machine Learning*, *39*, 103–134.

Quinlan, J. R. (1990). Learning logical definitions from relations. *Machine Learning*, *5*, 239–266.

Quinlan, J. R., & Cameron-Jones, R. M. (1993). FOIL: A midterm report. *Proceedings of the Fifth European Conference on Machine Learning* (pp. 3–20). Springer-Verlag.

Slattery, S., & Craven, M. (1998). Combining statistical and relational methods for learning in hypertext domains. *Proceedings of the Eighth International Conference on Inductive Logic Programming*. Madison, WI.

Yang, Y., & Pedersen, J. (1997). A comparative study on feature selection in text categorization. *Proceedings of the Fourteenth International Conference on Machine Learning*. Nashville: Morgan Kaufmann.