Integrating Background Knowledge into Nearest-Neighbor Text Classification

Sarah Zelikovitz and Haym Hirsh

Computer Science Department
Rutgers University
110 Frelinghuysen Road
Piscataway, NJ 08855
{zelikovi,hirsh}@cs.rutgers.edu

Abstract. This paper describes two different approaches for incorporating background knowledge into nearest-neighbor text classification. Our first approach uses background text to assess the similarity between training and test documents rather than assessing their similarity directly. The second method redescribes examples using Latent Semantic Indexing on the background knowledge, assessing document similarities in this redescribed space. Our experimental results show that both approaches can improve the performance of nearest-neighbor text classification. These methods are especially useful when labeling text is a labor-intensive job and when there is a large amount of information available about a specific problem on the World Wide Web.

1 Introduction

The abundance of digital information that is available has made the organization of that information into a complex and vitally important task. Automated categorization of text documents plays a crucial role in the ability of many applications to sort, direct, classify, and provide the proper documents in a timely and correct manner. With the growing use of digital devices and the fast growth of the number of pages on the World Wide Web, text categorization is a key component in managing information.

The machine learning community approaches text-categorization problems as "supervised" learning problems. In this case a human expert simply has to label a set of examples with appropriate classes. Once a corpus of correctly labeled documents is available, there are a variety of techniques that can be used to create a set of rules or a model of the data that will allow future documents to be classified correctly. The techniques can be optimized and studied independently of the domains and specific problems that they will be used to address. The problem with the supervised learning approach to text classification is that often very many labeled examples (or "training examples") must be used in order for the system to correctly classify new documents. These training examples must be hand-labeled, which might be quite a tedious and expensive process.

The question that we address is as follows: Given a text categorization task, can we possibly find some *other* data that can be incorporated into the learning process that will improve accuracy on test examples while limiting the number of labeled training examples needed? We believe that the answer is most often "yes". For example, suppose that

we wish to classify the names of companies by the industry that it is part of. A company such as *Watson Pharmaceuticals Inc* would be classified with the label *drug*, and the company name *Walmart* would be classified as type *retail*. Although we many not have numerous training examples, and the training examples are very short, we can find other data that is related to this task. Such data could be articles from the business section of an on-line newspaper or information from company home pages. As a result of the explosion of the amount of digital data that is available, it is often the case that text, databases, or other sources of knowledge that are related to a text classification problem are easily accessible. We term this readily available information "background knowledge". Some of this background knowledge can be used in a supervised learning situation to improve accuracy rates, while keeping the hand-labeled number of training examples needed to a minimum.

One common approach to text classification is to use a k-nearest-neighbor classification method, wherein the k documents closest to a test document are found and their labels "vote" on the classification of the new example. The standard approach for representing text documents for use by such methods is to represent each document simply by the "bag" of words in the document. Each word is viewed as a dimension in a very high-dimension vector space, one dimension per word. Every document is then a vector in this space, with a zero in its vector for every word that does not appear in the document, and a non-zero value for every word that does appear in the document. The non-zero values are set by using weighting schemes whereby words that occur frequently in a document are given higher values, with values scaled down by the extent to which the word also occurs frequently throughout all documents. With each document now representable in this vector space, similarity between two documents is measured using the cosine of the (normalized) vectors representing the two documents.

This paper describes two approaches for integrating background knowledge into text classification. In the next section we describe an approach by which background knowledge is compared to both the training and test examples to determine which training examples are closest to the test example (Zelikovitz & Hirsh, 2000). Section 3 then describes our second approach, in which the background knowledge is used to reformulate both the training examples and test examples, so that document comparisons are performed in the new space (Zelikovitz & Hirsh 2001). In both cases we show that classification accuracy is generally improved by these two different approaches for incorporating background knowledge.

2 Using Background Knowledge to Assess Document Similarity

Instead of simply comparing a test example to the corpus of training examples, our first idea is to use the items of background knowledge as "bridges" to connect each new example with labeled training examples. A labeled training example is useful in classifying an unknown test instance if there exists some set of unlabeled background knowledge that is similar to both the test example and the training example. We call this a "second-order" approach to classification (Zelikovitz & Hirsh, 2000), in that data are no longer directly compared but rather, are compared one step removed, through an intermediary. To accomplish this goal we use WHIRL (Cohen, 1998) which is a conven-

tional database system augmented with special operators for text comparison. It has been shown to yield an extremely effective nearest-neighbor text classification method (Cohen & Hirsh, 1998).

WHIRL makes it possible to pose SQL-like queries on databases with text-valued fields. Using WHIRL we can view the training examples as a table with the fields *instance* and *label*, and the test example as a table with the field *instance* and the background knowledge as a table with the single field, *value*. We can then create the following query for classification:

```
SELECT Test.instance, Train.label
  FROM Train AND Test AND Background
WHERE Train.instance SIM Background.value
  AND Test.instance SIM Background.value
```

The SIM function computes distances between vectors using the cosine metric described earlier, which returns a score between 0 and 1 that represents the similarity between the documents. Here each of the two similarity comparisons in the query computes a score, one comparing the background item to the training example, and the second comparing the background item to the test example. WHIRL multiplies the two resulting scores to obtain a final score for each tuple in an intermediate-results table. The final voting is performed by projecting this intermediate table onto the *Test.instance* and *Train.label* fields (Cohen & Hirsh 1998). All examples among the k nearest neighbors having the same class label vote by combining their score using the "noisy or" operation. Whichever label has the highest score in the resulting projected table is returned as the label for the test instance.

We ran both the base approach for nearest neighbor classification and our method incorporating background knowledge on a range of problems from nine different text classification tasks. Details on the data sets can be found elsewhere (Zelikovitz 2002); each varied on the size of each example, the size of each piece of background knowledge, the number of examples and number of items of background knowledge, and the relationship of the background knowledge to the classification task. Results are graphed in Figure 1. The x axis corresponds to percent accuracy on the test set when using the base method and the y axis corresponds to percent accuracy on the test set when background knowledge is incorporated. Each plotted point represents a different set of data, and those above the line $y = x$ have higher accuracy with the inclusion of background knowledge. As can be seen, on some data sets performance was hurt slightly by using background knowledge, but in most cases performance was improved, in some cases by more than 50%.

3 Using Background Knowledge to Reformulate Examples

In our second approach the background knowledge is used to redescribe both the training and the test examples. These newly expressed documents therefore contain information based upon the set of background knowledge. The newly expressed training examples are then compared to the redescribed test example so that the nearest neighbors can be found.

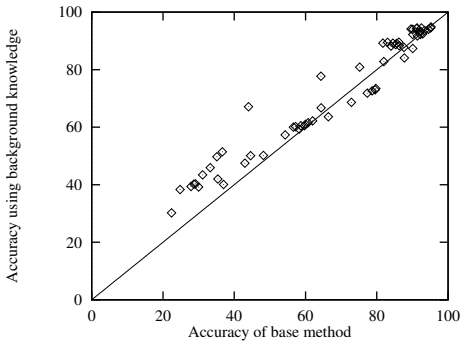

Fig. 1. Comparison of accuracy rates with and without background knowledge

A corpus of text documents represented as vectors can be looked at as a large, sparse term-by-document ($t \times d$) matrix. Latent Semantic Indexing (LSI) (Deerwester *et al*, 1990) is an automatic method that uses the $t \times d$ matrix to redescribe textual data in a new smaller semantic space using singular value decomposition. The original space is decomposed into linearly independent dimensions or "factors", and the terms and documents of the training and test examples are then represented in this new vector space. Documents can then be compared as described in the last section, only now comparisons are performed in this new space. Documents with high similarity no longer simply share words with each other, but instead are located near each other in the new semantic space.

LSI is traditionally used for text classification by performing the singular value decomposition using the training data. Our key idea is to use the background text in the creation of this new redescription of the data, rather than relying solely on the training data to do so. The background knowledge is added to the training examples to create a much larger $t \times d$ matrix (Zelikovitz & Hirsh 2001), where the terms now include all terms from the background knowledge, and the documents in the background knowledge are added as columns to the original matrix. LSI is then used to reduce this matrix so that the training examples are redescribed in a smaller semantic space that was created based upon the background knowledge. New test examples can be redescribed in this space as well, and a test example can then be directly compared to the training examples, with a cosine similarity score determining the distance between the test example and each training example. A test document is then classified with the label associated with the highest score.

Results for using LSI for classification without background knowledge versus with background knowledge is presented in Figure 2. As before, each point represents a data set, with the y-axis presenting the accuracy of the LSI method using background knowledge, and the x-axis presenting the accuracy of the base method. Here, too, the use of background knowledge most often improves the learner.

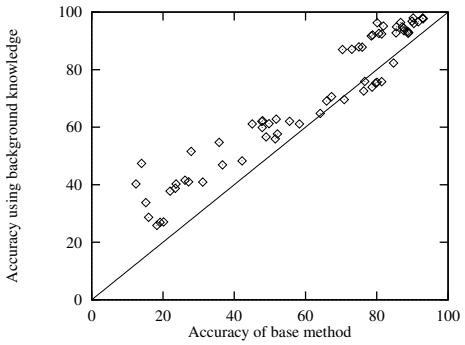

Fig. 2. Comparison of accuracy rates with and without background knowledge

4 Summary

Text classification is a process of extrapolating from the labels of given training data to assign labels to test data. Nearest neighbor methods perform this process by finding the training examples near each test example and having them vote for the label of the example. This paper has described two ways to modify nearest-neighbor text classification methods to incorporate background knowledge. Our first approach redefines the similarity metric by bridging each training and test example by one or more pieces of background knowledge. Our second approach redefines the space in which similarity is assessed, using Latent Semantic Indexing on the training data and background knowledge to create a new vector space in which the documents are placed. In both cases we were able to show consistent improvements in classification accuracy on a range of benchmark problems.

References

1. William Cohen. A web-based information system that reasons with structured collections of text. *Proceedings of Autonomous Agents*, 1998.
2. W. Cohen and H. Hirsh. Joins that generalize: Text categorization using WHIRL. *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, pages 169–173, 1998.
3. S. Deerwester, S. Dumais, G. Furnas, and T. Landauer. Indexing by latent semantic analysis. *Journal for the American Society for Information Science*, 41(6):391–407, 1990.
4. S. Zelikovitz. *Using Background Knowledge to Improve Text Classification*. PhD thesis, Rutgers University, 2002.
5. S. Zelikovitz and H. Hirsh. Improving short text classification using unlabeled background knowledge to assess document similarity. *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 1183–1190, 2000.
6. S. Zelikovitz and H. Hirsh. Using LSI for text classification in the presence of background text. *Proceedings of the Tenth Conference for Information and Knowledge Management*, 2001.