Automatic Generation of Background Text to Aid Classification

Sarah Zelikovitz and **Robert Hafner**
The College of Staten Island of CUNY
2800 Victory Blvd
Staten Island, NY 10314
{zelikovitz,hafner}@mail.csi.cuny.edu

Abstract

We illustrate that Web searches can often be utilized to generate background text for use with text classification. This is the case because there are frequently many pages on the World Wide Web that are relevant to particular text classification tasks. We show that an automatic method of creation of a secondary corpus of unlabeled but related documents can help decrease error rates in text categorization problems. Furthermore, if the test corpus is known, this related set of information can be tailored to match the particular categorization problem in a transductive approach. Our system uses WHIRL, a tool that combines database functionalities with techniques from the information retrieval literature. When there is a limited number of training examples, or the process of obtaining training examples is expensive or difficult, this method can be especially useful.

Introduction

The machine learning approach to automatic text classification has been studied and surveyed over the past few years (Sebastiani 1999; Nigam *et al.* 2000; Joachims 2002). With the proliferation of textual data on the World Wide Web, the need to catalog and categorize textual data often becomes an expensive and time consuming task. Some examples of this include categorization of Web pages into correct directories for ease of searching, email messages flagged as urgent or spam, and news articles sent to interested users. The machine learning community approaches text-categorization problems as "supervised" learning problems. In this case the 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.

Many researchers have noted, that although labeled examples may be difficult to obtain, there are often many unlabeled examples (Nigam *et al.* 2000; Li & Liu 2003; Belkin & Niyogi) that are easily obtainable. Textual data that is related to the task, but that is not necessarily a corpus of unlabeled documents, can also be used to aid the text classification task (Zelikovitz & Hirsh 2001). Given a set of labeled data, we create queries for a search engine to obtain textual data from the World Wide Web that is related to this specific problem. This automatically created set of background text can then be used in conjunction with a text classifier to correctly classify previously unseen examples.

In order to achieve our goal, we use WHIRL (Cohen 1998) which is a conventional database system augmented with special operators for text comparison. Its use as a text classification program is a nearest neighbor approach (Cohen & Hirsh 1998), with text documents specified as TFIDF vectors, and similarity between text documents measured as cosine similarity (Salton 1989). WHIRL makes it possible to pose SQL-like queries on databases with text-valued fields. If we consider the training examples as a table, the automatically created background text as a table, and a test example as a table as well, WHIRL allows for succinct queries that specify the combination of training similarity and background similarity to a new test example.

Our paper is organized as follows. First we give a description of WHIRL for text classification, and present the queries in WHIRL that allow for the use of background text. We then describe our method of automatically generating the set of background text. We present a description of the four domains on which we tested our system and present two sets of results for each of the domains for varied data set sizes. We conclude with a discussion of the various possible dimensions that our choices along the way can take and directions for current and future research.

Our Approach

Text Classification Using WHIRL

WHIRL (Cohen 1998) is an information integration tool that is specifically designed to query and integrate varied textual sources from the Web. WHIRL's SQL-type queries can

search and retrieve textual sources based upon specified conditions. Assume that we have a corpus of training examples with labels, and a test example that must be assigned a label. The training examples can be viewed as a table with the field *instance*, to hold the textual data, and the field *label* to hold the class label. The test example is a one line table, with simply the textual field *instance*. An example of a WHIRL query (Cohen & Hirsh 1998) is:

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

Given a user-specified parameter *K*, this query will first generate an intermediate table containing the *K* tuples

$$\langle \textit{Test.instance}, \textit{Train.instance}, \textit{Train.label} \rangle$$

that maximize the similarity score between *Test.instance* and *Train.instance*. Unlike traditional SQL queries, the result of this is a set of tuples ordered by score, with the highest score representing the closest *Train.instance*, *Test.instance* pair, using WHIRL's SIM operator to compute the similarity of these textual documents.

To determine similarity WHIRL computes a model of each text document by representing each document as a vector in a vector space. This representation is computed by passing each document through a stemmer (Porter 1980) and by then computing weights for each term using the TFIDF (Salton 1989) weighting method. Distances between vectors are computed using the cosine metric, which represents the statistical similarity between documents.

In WHIRL's final step it takes this table of K tuples and projects it onto the fields specified in the SELECT statement. Note that this can mean that there may be many training examples among the K with the same label (i.e., multiple nearby examples in the training set), and these are combined into a single tuple in the final result table. The combination of scores is performed by treating scores as probability and the combination as a "noisy or." If the individual scores of the tuples with a given label are $\{s_1, \ldots, s_n\}$, the final score for that label is $1 - \prod_{i=1}^{n}(1 - s_i)$. Whichever label has the highest score in the resulting projected table is returned as the label for the test instance.

Incorporating Background Text into WHIRL Queries

We can assume that the World Wide Web contains much information about topics and classes that we are dealing with. We can use some of this information as background text for the classification task. Often this background text does not fit clearly into any one of our predefined categories. However, background text can give us information about the co-occurrences of words, as well as the frequency of different words in the domain. The background text can also enhance the sometimes meager vocabulary of the domain that has been created by using only the training examples. This gives us a larger context in which to test the similarity of a training example with a new test example. We can use this context in conjunction with the training examples to label a new example.

Because of WHIRL's expressive language, and the ability to create conjunctive queries simply by adding conditions to a query, WHIRL's queries for text classification can be expanded to allow for the use of "background text" on a subject. If we automatically create a set of textual data that is related to the task and place it in a relation called Background with a single field *value*, we can create the following query (Zelikovitz & Hirsh 2000; 2001):

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

Here each of the two similarity comparisons in the query computes a score, and WHIRL multiplies them together to obtain a final score for each tuple in the intermediate-results table. This table is then projected onto the *Test.instance* and *Train.label* fields as discussed before. Whichever label gives the highest score is returned as the label for the test example.

One way of thinking about this is that rather than trying to connect a test example directly with each training example, it instead tries to bridge them through the use of an element of the background table. Note that WHIRL combines the scores of tuples generated from different matches to the background table. Our use of WHIRL in this fashion thus essentially conducts a search for a set of items in the background text that are close neighbors of the test example, provided that there exists a training example that is a neighbor of the background text as well. It is important to realize that a background instance that is close to numerous training instances can be included more than once in the table returned by the WHIRL query – even if the training examples that it is close to have different classes. Similarly, a training example can also be included in the table multiple times, if it is close to numerous background instances.

Generating Background Text

In order for the background text to be useful in the WHIRL query described in the previous section, it must be related to both the training corpus and the test corpus, and so be able to form a bridge between these two corpora. Given the huge proliferation of information on the World Wide Web, we can safely make the assumption that there are many pages related to a given text classification task that are easily accessible. If we could harvest this information in useable form, and create a background set of information that can be placed in a table and used by WHIRL, our text classifier will need fewer hand-labeled training examples, and achieve higher levels of accuracy.

Suppose we wish to classify the titles of web pages for a veterinary site as belonging to specific animals (cat, dog, horse, etc) as in www.netvet.wustl.edu, to facilitate the organization of a web site that will be helpful to people interested in these individual topics. The text categorization task can be defined by looking at the titles as the individual examples, and the classes as the list of animals that could be assigned to any specific title. If many web-page titles have already been classified manually, machine learning algorithms can be used to classify new titles. If only a small number

of web-page titles are known to fit into specific categories, these algorithms may not be useful enough. However, it is clear that the WWW contains many pages that discuss each of these animals. These pages can be downloaded and organized into a corpus of background text that can be used in the text classification task. It might be the case that many of these pages do not clearly fit into any one of the specific categories. For example, pages might discuss pets, which would be relevant to both cats and dogs, but not perhaps to primates and frogs. Still, these pages are in the proper domain and have relevance to the topic and can help learners in terms of word frequencies and co-occurrences.

An example of the top page that is returned by Google for the training example:

> health management for your horse, purdue university

is:

> http://www.amazon.co.uk/exec/obidos/external-search/202-8569362-7607027?tag=featureditems21&keyword=health-+management+for+your+horse+purdue+univers&index-=blended

which is a page from *amazon.co.uk*. If we look at the text words that are on the page, we can see that many of the words are related to the animal/pet domain that our categorization task is from. A portion of this page can be seen below. As can be seen, the words *stable, equipment, grooming,* and *pony* all occur in this text, and they can be helpful in future categorization of documents.

```
The Horse in Winter: How to Care for Your Horse
 During the Most Challenging Season of the Year
-- Susan McBane (Hardcover - Lyons Press - 1 August, 2003)
 Our Price: #16.82 -- Used from: #14.96
Natural Healing for Horses: The Complete Guide to
Complementary Health Care for Your Horse -- Jenny Morgan
(Hardcover - Gaia Books - 27 February, 2002)
 Our Price: #19.99 -- Used from: #99.95
Caring for Your Horse: The Comprehensive Guide to
 Successful Horse and Pony Care : Buying a Horse, Stable
 Managements, Equipment, Grooming and First Aid -- Judith Draper,
 Kit Houghton (Hardcover - Smithmark Pub - 1 March, 1997)
```

Given a set of training examples, and a set of test examples, we automatically created a corpus consisting of related documents from the World Wide Web. Our application to do this, written in Java, chose each training example to be used as an individual query to be input to the Google search engine. Google provides an API, which can be used to query its database from Java, thus eliminating the need to parse sophisticated set of parameters to be passed to it through the API. We restricted our queries to only retrieve documents written in the English language and we restricted the document type to be of *html* or *htm*. This avoided the return of .pdf files, .jpeg files, as well as other non-text files that are on the WWW. Once the Google API returned the results of the search, our application then started a thread to download the each individual page from the URLs. The thread was given a maximum of 5 seconds to download and retrieve the document before timing out. We saved the textual sections of the documents that were downloaded, and each one became an entry into our background text relation.

Our program allows for the specification of the number of web pages that should be returned for each query. In the following results sections, we present results using only the top one web page returned by Google. This allowed for the creation of background sets that had the same number of entries as the training examples, but that contained examples that were much longer. In the NetVet example introduced above, the training set examples had an average length of 4.9 words, whereas the background text entries had an average length of 1135.8 words.

Data Sets and Experimental Methodology

We have tested the systems that we present with numerous different text-categorization problems. (These data sets can be downloaded from www.cs.csi.cuny.edu/~zelikovi/data.htm.)

Technical papers: One common text categorization task is assigning discipline or sub-discipline names to technical papers. We used a data set from the physics papers archive (http://xxx.lanl.gov), where the training set consists of *titles* for all technical papers in two areas in physics (astrophysics, condensed matter) for the month of March 1999 (Zelikovitz & Hirsh 2000). In total there were 953 in the training-test set combined. Since we performed five-fold cross validation, for each run 80% of these 953 examples were used for training and 20% were held for testing.

Web page titles: As discussed above, the NetVet site (http://netvet-.wustl.edu) includes the Web page headings for pages concerning cows, horses, cats, dogs, rodents, birds and primates (Cohen & Hirsh 1998). Each of these titles had a URL that linked the title to its associated Web page. For the labeled corpus, we chose half of these titles with their labels, in total 1789 examples. Once again, five-fold cross-validation was used on this half of the titles to divide it into training and test sets.

Business Names: Another data set consisted of a training set of company names, 2472 in all, taken from the Hoover Web site (http://www.hoovers.com) labeled with one of 124 industry names. The class *retail* had the most business names associated with it; at 303 examples, and there were a few classes with only one example each. The average number of examples per class was 20.

Thesaurus: Roget's thesaurus places words in the English language into one of six major categories: space, matter, abstract relations, intellect, volition, and affection. For example, a word such as "superiority" falls into the category *abstract relations*, while the word "love" is placed into the category *affection*. Following the six links, one for each of these categories, from http://www.thesaurus.com, we created a labeled set of 1000 words, some including explanations for disambiguation from the web site, with each word associated with one category.

For all the data sets we used five fold cross validation to obtain accuracy results. We divided each dataset into five disjoint partitions, using four partitions each time for training and the fifth partition for testing. The results that we present are averages across all five runs. To see how are system performed when limited data is available, we kept the test set for one cross validation trial steady and we varied the number of training examples used for classification in the following manner. We used 100% of the training examples for each of the five runs, then only 80% of the training examples for each of the cross validation trials, then 60%,

40% and 20%. In this way we were able to see how accuracy changed when fewer training examples were used. The results that we present for each percentage of data are averages across five runs.

Results

In Figures 1–4 we present three sets of accuracy rates for each of the data sets that we have used. The first, termed WHIRL-nn, presents classification using WHIRL without the use of background text. This is a reasonable baseline, since it has been shown that this nearest neighbor method is competitive with many other state-of-the-art approaches (Cohen & Hirsh 1998). The second, termed WHIRL-data, uses the automatically created set of background text from the set of data. For each one of the cross validated runs, we created a new set of background data, based upon those training examples that were in the training set for that particular run. The average accuracy across five runs is reported for the entire data set, and 80%, 60%, 40%, and 20% of the data. For the smaller sets of training examples, only those web pages that were returned by Google for the examples in that particular small training set were used to create the background corpus.

For the third reported set of accuracy results, we took what we can term a "transductive" approach to the creation of the background text. A transductive learner, as defined by Vapnik (Vapnik 1998), as opposed to an inductive one, makes use of the test examples in the learning process. In this paradigm, we assume that the test examples are available during the learning process. Hence, the learner can mold itself to the actual set of test examples that it will need to classify. Taking advantage of the fact that the test examples are accessible, we used the test examples as queries to the search engine to obtain pages for the background corpus. These test examples are known, although the classes of the test examples are unknown. In this scenario, we are tailoring the learner to the particular test set that is being used, by using background text that is related to the test corpus. Of course, for each cross validated run we created a new set of background text, based upon that particular test set. We lablel this method WHIRL-test.

For our particular nearest neighbor approach, it is intuitive that the approach of searching on test examples should be useful. Since we connect training examples and test examples by using a piece of bacgkround text, background text can only be useful in classfying a test example if it is close to that specific test example. Searching on test examples to obtain a background text corpus is a way of assuring that the background text can be useful.

For Figures 1, 2, and 4 WHIRL with automatically created background text outperformed WHIRL with no background text. For the NetVet data set and the thesaurus data set, using a paired t-test on the average cross validated results for the different size data sets, the improvement in accuracy between WHIRL-nn and WHIRL-data is determined to be significant at the 99% level. For the physics paper title, NetVet and thesaurs data sets, a paired t-test comparing WHIRL-nn and WHIRL-test showed improvement to be statistically significant at the 95% level. In these three cases, for smaller data set sizes, our transductive WHIRL approach outperformed both other methods. For the Business data set our background text did not seem to be helpful, although accuracy did not degrade when it was added to the learner. This can perhaps be due to the fact that this set is a list of company names that was created by other researchers in 1998 (Cohen & Hirsh 1998), and current data on the web may be unrelated to many of these companies.

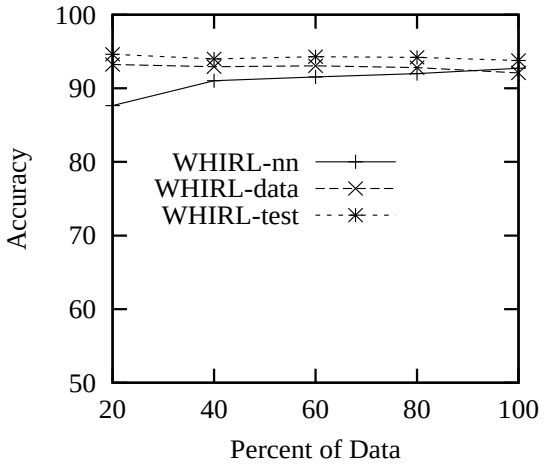

Figure 1: Technical Papers dataset

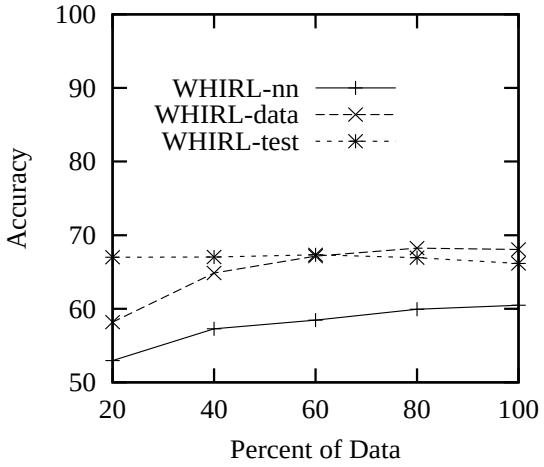

Figure 2: NetVet dataset

Related Work

As we noted above, unlabeled examples can be looked upon as a limited type of background text in the context of our experiments. Most of the work on unlabeled examples has been empirical in nature, and it can be shown that unlabeled examples can improve classification accuracy when added into many different classifiers. (Nigam *et al.* 2000; Joachims 1999; Zelikovitz & Hirsh 2000; 2001; Blum &

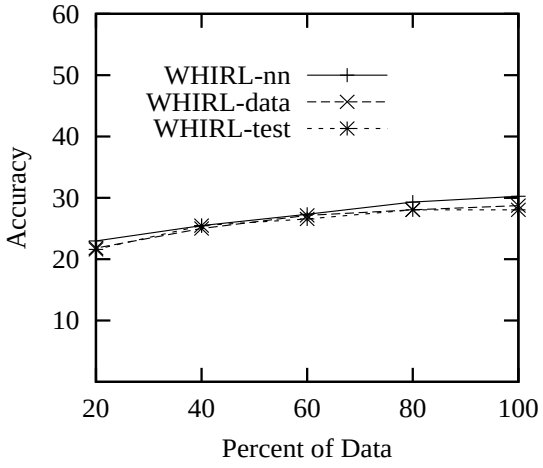

Figure 3: Business dataset

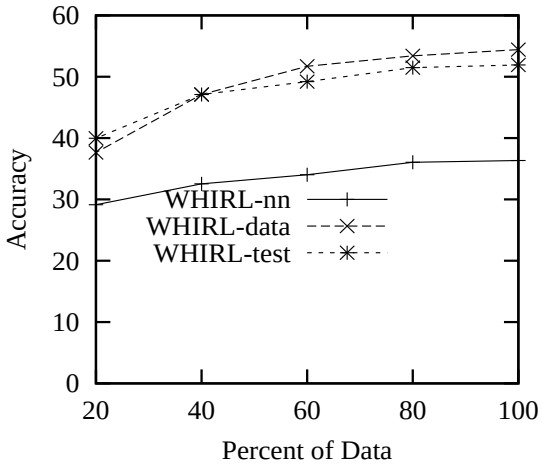

Figure 4: Thesaurus dataset

Mitchell 1998; Mitchell 1999). Nigam et al (Nigam *et al.* 2000) have done work on this using Expectation Maximization (EM) and a naive Bayes classifier. The parameters of the naive Bayes classifier are set using labeled examples. The learned model is then used by EM to probabilistically classify unlabeled documents, with the resulting collection of classified documents used to estimate a new set of parameters for naive Bayes. Li and Liu (Li & Liu 2003) use a corpus of positive examples and a large set of unlabeled examples. They extract probable negative examples from the unlabeled corpus, and use support vector machine iteratively to create an accurate learner.

Joachims (Joachims 1999) presents an algorithm for transductive SVM that chooses, from among all the possible hyper-surfaces, the one that correctly classifies the data while maximizing the margin[1] of both the training and test data. In a sense the use of the test data during the text classification task injects some prior knowledge of words that can be used in the decision of the choice of the hyperplane that would be used for classification. Joachims presents results for text classification tasks that show the validity of this approach. Results by others (Bennet & Demiriz 1998) make use of what they term a "working set", which is essentially a corpus of unlabeled data, in SVMs as well. Joachims (Joachims 2003) more recently introduced the concept of transductive k nearest-neighbor, which makes use of the test examples as well.

A different approach to combining labeled and unlabeled information is taken by Blum and Chawla (Blum & Chawla 2001). They create a graph from the examples by connecting all labeled examples to the node corresponding to the class assignment and to each other based upon their similarities. Unlabeled examples are connected to their nearest neighbors. The min-cut algorithm is then used to find the best division of the data into positive and negative classes. Issues of how to measure similarities and how many edges to connect to each unlabeled example are some of the directions of current research.

The co-training paradigm (Blum & Mitchell 1998) takes advantage of the fact that a data set can often be expressed in more than one way. For example, a web page can be described by the words on the page, or by words in hyperlinks that point to that page. This structural knowledge about the data can be exploited to use unlabeled data. Initially, two hypothesis, H_1 and H_2 are formed on each of the views using the labeled data. A subset of the unlabeled data is then classified alternatively by h_1 and h_2 and added to the other view to be part of the labeled set.

Unlabeled data has been used in text classification tasks where no labeled data is available, but a class hierarchy or set of keywords per class is known (McCallum & Nigam 1999; Riloff & Jones 1999). In these cases, the keywords that are given for each class are used to assign preliminary labels to the documents. These labels are then used as a starting point for expectation maximization and naive Bayes in conjunction with unlabeled examples to arrive at a set of

[1]The margin is defined to be the distance between the hyperplane and the nearest examples of the given classes.

parameters for a generative classifier.

Research Questions

We have a presented a method for automatically producing background text for a categorization problem, and have shown its usefulness in conjunction with a nearest neighbor learner using WHIRL.

However, there are a number of issues that we wish to explore further in the creation of background text. Our approach has been to search on individual training and test examples. We wish to determine whether it would be beneficial to search on the important words in the corpus. Preliminary results have shown use that simply searching on class names did not return useful corpora, but if we search on words with the highest information gain in the training set perhaps it would be more useful. It is also possible that we can combine training and test examples to create new queries, which can possible retrieve web pages that can be most useful with WHIRL in terms of forming bridges between the test examples and the training examples. We are currently also exploring the size of the background corpus, and how many pages should be returned for each query.

References

Belkin, M., and Niyogi, P. Semi-supervised learning on manifolds. *Machine Learning Journal: Special Issue on Clustering, Forthcoming*.

Bennet, K., and Demiriz, A. 1998. Semi-supervised support vector machines. *Advances in Neural Information Processing Systems* 12:368–374.

Blum, A., and Chawla, S. 2001. Learning from labeled and unlabeled data using graph mincut. In *Proceedings of the Eighteenth International Conference on Machine Learning*.

Blum, A., and Mitchell, T. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*, 92–100.

Cohen, W., and Hirsh, H. 1998. Joins that generalize: Text categorization using WHIRL. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, 169–173.

Cohen, W. 1998. Integration on heterogeneous databases without common domains using queries based on textual similarity. In *Proceedings of ACM-SIGMOD 98*.

Joachims, T. 1999. Transductive inference for text classification using support vector machines. In *Proceedings of the Sixteenth International Conference on Machine Learning*, 200–209.

Joachims, T. 2002. *Learning to Classify Text using Support Vector machines*. Ph.D. Dissertation, University of Dortmund.

Joachims, T. 2003. Transductive learning via spectral graph partitioning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 290–297.

Li, X., and Liu, B. 2003. Learning to classify text using positive and unlabeled data. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, 587–594.

McCallum, A., and Nigam, K. 1999. Text classification by bootstrapping with keywords, EM and shrinkage. In *ACL '99 Workshop for Unsupervised Learning in Natural Language Processing*.

Mitchell, T. 1999. The role of unlableled data in supervised learning. In *Proceedings of the Sixth International Colloquium on Cognitive Science*.

Nigam, K.; Mccallum, A. K.; Thrun, S.; and Mitchell, T. 2000. Text classification from labeled and unlabeled documents using EM. *Machine Learning* 39(2/3):103–134.

Porter, M. F. 1980. An algorithm for suffix stripping. *Program* 14(3):130–137.

Riloff, E., and Jones, R. 1999. Learning dictionaries for information extraction using multi-level boot-strapping. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, 474–479.

Salton, G., ed. 1989. *Automatic Text Processing*. Reading, Massachusetts: Addison Welsley.

Sebastiani, F. 1999. Machine learning in automated text categorization. *Technical Report IEI-B4-31-1999*.

Vapnik, V. N. 1998. *Statistical Learning Theory*. Wiley.

Zelikovitz, S., and Hirsh, H. 2000. Improving short text classification using unlabeled background knowledge to assess document similarity. In *Proceedings of the Seventeenth International Conference on Machine Learning*, 1183–1190.

Zelikovitz, S., and Hirsh, H. 2001. Using LSI for text classification in the presence of background text. In *Proceedings of the Tenth Conference for Information and Knowledge Management*, 113–118.