Using Bins to Empirically Estimate Term Weights for Text Categorization

Carl Sable
450 Computer Science Building
Columbia University
New York, NY 10027
sable@cs.columbia.edu

Kenneth W. Church
AT&T Shannon Laboratory
180 Park Avenue
Florham Park, NJ 07932
kwc@research.att.com

Abstract

This paper introduces a term weighting method for text categorization based on smoothing ideas borrowed from speech recognition. Empirical estimates of weights (likelihood ratios) become unstable when counts are small. Instead of estimating weights for individual words, as Naive Bayes does, words with similar features are grouped into bins, and a single weight is estimated for each bin. This weight is then assigned to all of the words in the bin. The bin-based method is intended for tasks where there is insufficient training data to estimate a separate weight for each word. Experiments show the bin-based method is highly competitive with other current methods. In particular, this method is most similar to Naive Bayes; it generally performs at least as well as Naive Bayes, and sometimes better.

1 Introduction and Related Work

In recent years there has been considerable interest in text categorization techniques which assign documents to one or more categories. Most of these methods assume a supervised training setup, where the system is given some labeled training data: e.g., pairs of documents and category assignments. Documents, and sometimes categories, are often represented as weighted word vectors. Word weights are usually computed by combining separate features in some fashion, for example, by multiplying together the term frequency (TF) and inverse document frequency (IDF) of each word (Salton and Buckley, 1988; Salton, 1989).

The Naive Bayes method of text categorization (Lewis, 1998) empirically estimates term weights for each individual word that appears in the training set based on estimated probabilities of seeing each word in a document of each possible category. This method is prone to inaccurate term weights for words that occur infrequently in the training set. Words that have never been seen in the training set are ignored since all of the estimated probabilities are zero, and words that appear in only one category in the training set might appear to give that category infinite likelihood if they appear in a document from the test set. We will show that empirically estimating term weights for bins instead of individual words avoids these pitfalls, while at the same time providing evidence indicating which features of words are most important for indicating categories.

The Speech Recognition literature has developed a number of methods for smoothing term frequencies (e.g., Chapter 15 of (Jelinek, 1998)). These methods are important when the raw counts are small, and particularly important when the counts are zero. Both the Good-Turing method and the Deleted Interpolation method estimate $r*$, an adjusted value of r, where r is the number of times that the term t appeared in one corpus, and $r*$ is the number of times that t is expected to appear in another corpus of similar size and material.

The Deleted Interpolation method assigns each term, t, to a bin, b, usually based on the frequency r, but binning rules can also make use of other variables. Section 15.6 of (Jelinek, 1998), for example, discusses so-called "enhanced" binning rules where bigrams are assigned to bins not only based on their joint frequencies but also the frequencies of their parts. In this work, terms will be assigned to bins based on IDF and other statistics that are commonly used in the text categorization literature.

The Deleted Interpolation method splits the training collection into two pieces. The first piece is used to assign terms to bins, and to compute the number of terms that have been assigned to each bin, N_b. The second piece is used for calibrating bins. C_b is the number of times that the terms in bin b are found in the second piece. The final answer is then $r* \approx C_b/N_b$. In general, $r*$ tends to be slightly smaller than r in most cases except when $r = 0$. The adjustments are important when r is small, especially when $r = 0$. All of the terms in a bin receive the same adjusted frequency, $r*$.

(Umemura and Church, 2000) shows how the Deleted Interpolation approach can be generalized to estimate likelihood ratios instead of frequencies in an information retrieval application. In this paper, we will use a similar approach for text categorization. Text categorization is interestingly different

from information retrieval because there tends to be relatively more supervised training material.

2 Data Sets

Our research has focused on the categorization of news documents and their embedded images, and we have experimented with categories representing various levels of abstraction. The first experiment discussed in this paper involves the categorization of images based on associated captions as either Indoor or Outdoor; the second considers the categorization of entire news documents into the categories Struggle, Politics, Disaster, Crime, or Other. For these experiments, we compare our system to two competing systems previously implemented at Columbia University (Sable and Hatzivassiloglou, 2000; Sable, 2000) as well as several systems which comprise the publicly available Rainbow package (McCallum, 1996). In Section 6, we will discuss a third experiment involving Reuters documents and topic categories, for which we compare our system to all systems tested by Yang and Liu in (Yang and Liu, 1999).

The first two experiments use data sets from a corpus previously collected by researchers at Columbia University (Sable and Hatzivassiloglou, 2000; Sable, 2000). The raw data consists of news postings from November 1995 through January 1999 from a variety of Usenet newsgroups. Using one web interface, 1,675 captioned images were each labeled by two humans as Indoor, Likely Indoor, Ambiguous, Likely Outdoor, and Outdoor. 1,339 images were assigned definite labels in the same direction by both humans, and these images comprise the data set used in our first experiment. 894 images are used for training and 445 are used for testing. We use the same breakdown as (Sable, 2000) to allow for direct comparison. Using a second web interface, 1,750 news documents, each consisting of an article, image, and caption, were each labeled by two humans into one of the mutually exclusive categories Struggle, Politics, Disaster, Crime, or Other. We will refer to these as our Events categories in the remainder of this paper. 1,328 documents were assigned identical labels by both humans, and these documents comprise the data set used in our second experiment. 885 documents are used for training and 443 are used for testing, again using the same breakdown as (Sable, 2000).

Instructions provided to volunteers who labeled images as Indoor or Outdoor, including category definitions and guidelines, can be viewed at http://www.cs.columbia.edu/~sable/research/readme.html. The corresponding information provided to volunteers who labeled documents according to our Events categories can be seen at http://www.cs.columbia.edu/~sable/research/instr.html. Figure 1 and Figure 2 show two sample images with captions. Most captions include a first sentence which describes the associated image and one or two additional sentences which provide background information about the related story. All header information, including locations and dates, is automatically stripped before our experiments. Articles generally consist of many paragraphs and are typical in length to what you would expect in a standard newspaper.

3 Bins and Methodology

The general idea behind bins is to group words with similar features into a common bin. For example, you will see that in our first two experiments, we determine category counts, burstiness values, and quantized IDFs for all words, and then assign each word to a bin consisting of all words that share the same values. Once words have been placed into bins, we can estimate the likelihood of a word from a specific bin appearing in a document of a specific category with a specific occurrence count.

For each of our first two experiments, we create a separate set of bins for each category. For each category, every word is placed into a single bin based on the three features of the word. The first binning feature is the word's quantized IDF, or inverse document frequency. The IDF of a word represents a measure of the rarity of a word, and is calculated according to the formula:

$$IDF(word) = \log \frac{\text{Total number of documents}}{DF(word)}$$

where a word's DF, or document frequency, is the number of documents that contain the word. It is expected that words with higher IDFs will be more indicative of categories. Based on some initial experiments within the training set which were performed to determine how to quantize IDFs (i.e. to determine IDF boundaries which would be used to separate bins), we decided to simply truncate these values.

It is customary to compute the IDFs of words based on the training set, but another possibility is to compute these values based on some other, larger set of documents. On the one hand, using the training set ensures that the documents will have a similar style and format to documents that appear in the test set, but on the other hand, sometimes more data is better data. We ran some initial experiments comparing the use of our training set against the use of a year's worth of AP news documents consisting of approximately one million articles. We found that using the AP news documents leads to a significant improvement (i.e. more data is better), so we decided to use this larger corpus, instead of our train-

Intuition	Word	Indoor category count	Outdoor category count	IDF	burstiness
Clearly Indoor	conference	15	0	2.5	0
	bed	1	0	4.5	0
Clearly Outdoor	airplane	0	2	5.4	1
	earthquake	0	4	4.6	1
Unclear	Gore	1	1	4.5	1
	ceremony	5	6	3.9	0

Table 1: Values of binning features are used to assign words to bins.

ing set, to compute IDFs for the remainder of our experiments.

Following the suggestion of (Umemura and Church, 2000), our second binning feature is burstiness, an idea introduced in (Katz, 1996). Burstiness takes into account that some words are likely to appear many times in a document if they appear at all. The burstiness of a word is either one or zero, depending on whether or not the average term frequency of the word in documents in which it appears is greater than some specific cutoff. It is expected that bursty words (words with a burstiness of one) will be more indicative than words which are not bursty.

The third binning feature is the number of documents in the first half of the training set that belong to the particular category being examined and contain the word. We will refer to this as the category count of the word in the specified category. All category counts above some pre-determined maximum cut-off are set equal to this maximum. It is expected that words which appear multiple times in a category in the training set will be more indicative of that category. So, for example, one bin for our first experiment consists of all bursty words that have a truncated IDF of 3 and appear in two Outdoor documents in the first half of the training set (i.e. have an Outdoor category count of 3). Table 1 shows the values of the binning features used in our first experiment for six arbitrary words, where the category counts have been determined based on the first half of the training set.

Once bins have been determined, the second half of the training set is used to empirically estimate term weights for all bins. Each bin is assigned multiple term weights corresponding to different possible occurrence counts, or term frequencies, of words in documents (this will be elaborated shortly). Given a bin related to a specific category, we estimate the probability that a word belonging to the bin will appear in an article of the same category with some specific count. The log of this probability is used as a term weight for the bin.

For the first experiment, only first sentences of captions are used to determine bins and term weights or to predict labels. (The first sentence of a caption generally describes the associated image, while the rest gives background information about the related story.) We therefore only measure probabilities for two specific occurrence counts of words in bins; namely, zero versus one or more. For the second experiment, full news articles are used, and it is often the case that a word occurs more than once. For this experiment, separate term weights are used for different possible occurrence counts. For each bin, the separate possible occurrence counts are zero, one, two, three, or four or more. It is expected that, other things being equal, the more a word occurs in a document, the more indicative it is of a category.

The above two steps are best illustrated by example. For our first experiment, the word "airplane" appears in two Outdoor documents (i.e. captions of Outdoor images) in the first half of the training set, but no Indoor documents in the first half of the training set. The IDF of "airplane" is 5.4 and the word is bursty. Therefore, "airplane" is placed in one bin representing all bursty words with a truncated IDF of 5 that appear in two Outdoor documents, and we will refer to this as the Outdoor bin of "airplane". The same word is placed in another bin representing all bursty words with a truncated IDF of 5 that appear in zero Indoor documents, and we will refer to this as the Indoor bin of "airplane". When the second half of the training set is examined, it is noted that the estimated probability that a word belonging to the same Indoor bin as "airplane" appears one or more times in an Indoor document is 2.11×10^{-4}, and so the term weight for this bin is:

$$\lambda_{indoor} = \log_2(2.11 \times 10^{-4}) = -12.21$$

It is also noted that the estimated probability that a word belonging to the same Outdoor bin as "airplane" appears one or more times in an Outdoor document is 2.90×10^{-3}, and so the term weight for this bin is:

$$\lambda_{outdoor} = \log_2(2.90 \times 10^{-3}) = -8.43$$

Intuition	Word	$\lambda_{indoor} - \lambda_{outdoor}$
Clearly Indoor	conference	5.91
	bed	4.58
Clearly Outdoor	airplane	-3.78
	earthquake	-4.86
Unclear	Gore	0.74
	ceremony	-0.32

Table 2: Term weights for these words fit intuition as to which words indicate which category.

The difference between the term weight associated with a word's Indoor bin and the term weight associated with a word's Outdoor bin is a log likelihood ratio comparing the likelihood of seeing the word (or one with the same features) in one category versus the other. In this case, $\lambda_{outdoor} - \lambda_{indoor} = -8.43 - (-12.21) = 3.78$, which means that if a word with the same features as "airplane" is seen in a document, it is about $2^{3.78} = 13.74$ more likely to be an Outdoor document than an Indoor document.

This example should help to illustrate why the training set is divided into two halves. The first half of the training set determines bins, and the second half is used to calibrate bin probabilities. Let's say that "airplane" does not occur in any Indoor documents in the training set. This does not mean that if the word is seen in a test document, it is necessarily an Outdoor document. Since "airplane" is placed in an Indoor bin with other words that occur in no Indoor documents in the first half of the training set, and some of these words may appear in some Indoor documents in the second half of the training set, we are able to estimate a probability for the bin as a whole of seeing a word from the bin in an Indoor document.

Table 2 shows the results of subtracting the Outdoor term weights from the Indoor term weights for the bins of the the same six words shown in Table 1. As expected, words such as "conference" and "bed" are good Indoor indicators, while words such as "airplane" and "earthquake" are good Outdoor indicators. The words "Gore" and "ceremony" each show a slight preference for one category over the other, but according to term weights, each is less than twice as likely in the favored category.

Once term weights for all bins have been calculated, we loop through the test documents to predict categories. For each test document, we iterate through its words, summing the appropriate term weights for each category. For example, for our second experiment, we generate five sums, one for each category. The first is the sum of the term weights for the Struggle bins associated with all words in the document (taking into account the specific occurrence counts of the words), the second is the sum of the term weights for the Politics bins, etc. Whichever category has the highest sum is considered the most likely and is therefore predicted. The likelihood of bins are assumed to be independent. This is similar to the independence assumption of words in Naive Bayes, and in fact the two methods are virtually identical if every word is assigned its own bin.

Figure 3 shows pseudo-code summarizing the algorithm. In actuality, we use a script consisting of approximately 650 lines of Perl code. All IDFs and burstiness values have been precomputed and are stored in pre-existing files. The first part of the script (corresponding to lines 1 through 16 of the pseudo-code) loads the IDFs and burstiness values, uses the first half of the training set to determine category counts of words, maps words to bins, and computes the size of each bin. Next (corresponding to lines 17 through 44 of the pseudo-code), the second half of the training set is used to estimate probabilities of all possible (bin, occurrence count) pairs, and corresponding term weights are computed. Some extra code is needed to compute counts for all instances of occurrence counts of 0 (see the comment at line 27 of the pseudo-code), and there are also checks throughout the code to avoid infinities. Finally (corresponding to lines 46 thorugh 60 of the pseudo-code), the test set is examined. For every test document, every word is mapped to its appropriate bins (one for each category), and the score for each category is incremented by the appropriate term weight. Each test document is assigned to the category with the highest score. The script generally completes a full execution in a matter of minutes.

4 Evaluation

For the first experiments, by construction, the categories are mutually exclusive, so the system outputs one and only one category label for each test document. The main evaluation measure is overall accuracy, which is the percentage of such predictions that are correct. In addition, we also report the F_1 measure (van Rijsbergen, 1979) for each category. The F_1 measure combines precision and recall into a single measurement according to the following for-

System	Overall Accuracy %	Indoor F_1 %	Outdoor F_1 %
Bins	84.9	73.3	89.5
Naive Bayes (Rainbow)	85.4	73.5	89.9
Rocchio/TF*IDF (Rainbow)	84.5	73.2	89.1
K-Nearest Neighbor (Rainbow)	77.8	65.3	83.6
Probabilistic Indexing (Rainbow)	**86.3**	**78.1**	90.0
SVMs (Rainbow)	82.0	66.9	87.7
Maximum Entropy (Rainbow)	84.5	70.9	89.4
Rocchio/TF*IDF (Columbia)	80.8	69.9	85.7
Density Estimation (Columbia)	86.1	73.7	**90.5**

Table 3: Our bin-based system outperforms five out of eight alternatives for Indoor versus Outdoor categorization.

System	Overall Accuracy %	Struggle F_1 %	Politics F_1 %	Disaster F_1 %	Crime F_1 %	Other F_1 %
Bins	88.5	87.5	88.0	97.2	**89.6**	58.5
Naive Bayes (Rainbow)	87.6	86.2	86.3	96.7	89.1	61.5
Rocchio/TF*IDF (Rainbow)	87.4	81.1	85.3	97.7	88.4	**68.3**
K-Nearest Neighbor (Rainbow)	81.9	80.0	79.7	95.6	75.6	63.2
Probabilistic Indexing (Rainbow)	86.5	83.6	84.8	97.2	89.4	65.0
SVMs (Rainbow)	**88.7**	**88.1**	**89.2**	96.2	87.0	57.9
Maximum Entropy (Rainbow)	88.3	**88.1**	87.9	95.7	87.9	55.6
Rocchio/TF*IDF (Columbia)	87.1	85.0	88.4	**98.8**	79.2	60.0
Density Estimation (Columbia)	84.9	83.7	86.0	97.3	80.0	34.3

Table 4: Our bin-based system outperforms seven out of eight alternatives for our Events categories.

mula:

$$F_1 = \frac{2 \times Precison \times Recall}{Precision + Recall}$$

This formula leads to a value that is closer to the smaller of precision and recall, and thus requires good results for both measurements to achieve a high F_1 score. In Section 6, we will consider a third experiment in which the categories are not mutually exclusive and therefore the system may output multiple categories for each document. Overall accuracy is considered a more useful evaluation measure in the mutually exclusive case, while evaluation measures based on F_1 are considered more useful in the non-exclusive case (this will be described in more detail in Section 6).

5 Results

Table 3 shows the results of all systems tested for our first experiment (involving the categories Indoor and Outdoor). These include the bin system described in this paper, six competing systems which are part of the publicly available Rainbow package, and two competing systems previously developed at Columbia University. All competing systems use text categorization techniques which have been previously described in literature. On this set of categories, our bin system performs somewhere in the middle, outperforming five of the eight competing systems tested. For further comparison, a baseline classifier which picks the larger category every time (Outdoor) achieves a 71.2% overall accuracy, significantly lower than all the automatic systems tested. Humans who were asked to predict labels after being shown only captions achieved an 87.6% overall accuracy, and we consider this a reasonable upper bound for how well a text categorization system might be expected to do.

Table 4 shows the performance of all systems tested for our second experiment (involving our Events categories). Note that the bin system outperforms all but one of the competing systems, based on overall accuracy, and only marginally loses to SVMs by 0.2% (which corresponds to one test document). A baseline categorizer which picks the largest category every time (which happens to be Struggle, based on the training set) would achieve only a 30.5% overall accuracy.

6 Reuters

We also performed an additional experiment using the ModApte split of the Reuters-21578 corpus (Lewis, 1997), a common benchmark for comparing methods of text categorization

method	miR	miP	miF1	maF1	error
SVM	.8120	.9137	.8599	.5251	.00365
kNN	.8339	.8807	.8567	.5242	.00385
LSF	.8507	.8489	.8498	.5008	.00414
NNet	.7842	.8785	.8287	.3765	.00447
NB	.7688	.8245	.7956	.3886	.00544
Bins	.8053	.7915	.7984	.4561	.00561

Table 5: Our bin-based system marginally outperforms Naive Bayes in terms of micro-average F_1 (miF1), and it significantly outperforms both Naive Bayes and Neural Nets in terms of macro-average F_1 (maF1).

(Joachims, 1998; Schapire and Singer, 2000; Yang and Liu, 1999). This collection can be obtained at http://www.research.att.com/~lewis/reuters21578.html. To allow for direct comparison with Yang and Liu (1999), all categories which did not contain at least one training document were eliminated, and then all unlabeled document were removed. This data set includes 90 categories, 7,770 training documents, and 3,019 test documents. Reuters-21578 is a binary categorization corpus, meaning that documents are allowed to have multiple labels, so a separate YES/NO decision is required for each (document, category) pair. The average number of categories assigned to a document is 1.24, and the most categories assigned to a single document is 15. The main purpose of this experiment was to see if our system, using bin-based smoothing, would outperform the Naive Bayes system tested by Yang and Liu. Since Naive Bayes is a similar method that estimates term weights for individual words instead of bins, and its performance was the lowest of the five systems tested by Yang and Liu (1999), we did not expect our bin-based system to do as well as the other systems.

The methodology described in Section 3, which relies on a separate set of bins for each category, is not appropriate for the Reuters data set. Although this data set contains significantly more total documents than those used for our first two experiments, it includes many small categories which do not have enough training examples to estimate term weights for per category bins. Instead, we implemented a modified methodology using only a single set of bins based on the entire training corpus. Each word is placed into a bin solely according to its IDF. The term weights computed estimate the likelihood that two documents share the same or similar categories, given that they share a word from a specific bin. (Recall that Reuters is a binary categorization corpus, and so it is possible for two documents to share some but not all of the same categories; However, it turns out that in most cases, documents share all or none of the same categories.)

At first, we used the same IDF values for binning as we did for the first two experiments; namely, those based on approximately one million AP news documents. It is expected that the words belonging to bins representing higher IDFs should be more indicative of shared categories, and that these bins should therefore have higher term weights. In general, this trend was seen, but there were some anomalies. A few bins had significantly lower term weights than expected. Further examination showed that in every such case, the anomaly was caused by one or two words that, due to stylistic differences between the two corpora, are much more common in Reuters than in AP. For example, the abbreviations "mln" for "million" and "pct" for "percentage" are both quite common in various Reuters categories, but extremely uncommon in AP, and therefore are assigned IDF values which are misleading. We did not want to switch entirely to IDF values based only on the Reuters training set, because that is a much smaller set than the AP corpus, and words which do not appear in the training set would have to be ignored or all placed in a single bin. So instead, we decided to use two IDF values for each word, one based on the Reuters training set and the other based on the AP corpus. These represent two different features of each word. In general, most words are assigned two similar IDF values, but the anomalies such as those just mentioned are not, and words like these are therefore filtered to their own bins, and do not affect the term weights of other bins.

For this experiment, we use the same metrics that are used by Yang and Liu (1999) to allow for direct comparison. These measures are micro-average recall (miR), micro-average precision (miP), micro-average F_1 (miF1), macro-average F_1 (miF1), and overall error (which is one minus overall accuracy). The macro-average F_1 computes the F_1 for all categories and then averages these numbers together (thus giving all categories equal weight), while the micro-average F_1 computes the F_1 once based on all binary decisions being made (thus giving all documents equal weight). These metrics are considered more important than overall accuracy (or overall error) for binary categorization, since most documents have few labels, and a trivial system which predicts no labels for every document can achieve high overall

accuracy.

Table 5 shows the results of our bin system (bottom row) and all systems tested by Yang and Liu (1999). The top five rows of the table are a reproduction of Table 1 from (Yang and Liu, 1999), showing the results of five state-of-the-art systems using methods which are commonly employed for text categorization. The five systems represented in these rows, from top to bottom, are Support Vector Machines, K-Nearest Neighbors, Least Squares Fit, Neural Nets, and Naive Bayes, all of which are described in (Yang and Liu, 1999). The two most important metrics shown are the micro-average F_1 and macro-average F_1. The bin system described in this paper has a micro-average F_1 which is marginally higher than the Naive Bayes system tested by Yang and Liu, while the macro-average F_1 is significantly better than two of the systems tested by Yang and Liu including Naive Bayes.

7 Conclusions and Future Work

This paper describes how to use bins to empirically estimate term weights for the purpose of text categorization. Using bins allows our system to estimate term weights for words which appear infrequently, or even not at all, in the training set. This smoothing effect can improve performance by avoiding inaccurate term weights for infrequent words. We have performed three complete experiments on very different data sets and compared our bin system to many competing systems using a variety of methods which have proved successful for text categorization in the past. Our results show that our bin-based system is highly competitive with other systems, and in particular, that it is usually at least as good as Naive Bayes.

In the future, we hope to test more binning features. Even without conducting full experiments, examination of term weights can determine which features are important. We hope to use query expansion, analogous to the way it was used with binning to improve information retrieval in (Umemura and Church, 2000), to boost results further. Also, we believe that it might be better to compute term weights for individual words when possible, and to back off to the bin only when necessary. In other words, when there is sufficient evidence to compute term weights for words, we should do so, but in other cases, we should back off and use bins. These changes may improve our system, which is already faring well against competitors. Additional information about this work can be found at http://www.cs.columbia.edu/~sable/bins.html, and we will continue to expand this page as our research continues.

References

F. Jelinek. 1998. *Statistical Methods for Speech Recognition.* The MIT Press, Cambridge, Massachusetts.

T. Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the European Conference on Machine Learning.*

S. Katz. 1996. Distribution of content words and phrases in text and language modelling.

D. Lewis. 1997. Reuters-21578 text categorization test collection, readme file (version 1.2).

D. Lewis. 1998. Naive (Bayes) at forty: The independence assumption in information retrieval. In *Proceedings of the European Conference on Machine Learning.*

A. McCallum. 1996. Bow: A toolkit for statistical language modeling, text retrieval, classification, and clustering. http://www.cs.cmu.edu/~mccallum/bow.

C. Sable and V. Hatzivassiloglou. 2000. Text-based approaches for non-topical image categorization. *International Journal of Digital Libraries*, **3**(3):261–275.

C. Sable. 2000. Categorizing multimedia documents using associated text (thesis proposal). http://www.cs.columbia.edu/~sable/proposal.

G. Salton and C. Buckley. 1988. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, **24**(5):513–523.

G. Salton. 1989. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer.* Addison-Wesley, Reading, Massachusetts.

R. Schapire and Y. Singer. 2000. Boostexter: A boosting-based system for text categorization. *Machine Learning*, **39**(2):135–168.

K. Umemura and K. W. Church. 2000. Empirical term weighting and expansion frequency. In *Joint SIGDAT Conference on Empircal Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000).*

C. J. van Rijsbergen. 1979. *Information Retrieval.* Butterworths, London, 2nd edition.

Y. Yang and X. Liu. 1999. A re-examination of text categorization methods. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-99).*

PONTA DELGADA, PORTUGAL, 1-NOV-1997: Rescue workers remove the body of a man from mud, October 31, following a landslide in Ribeira Quente, Azores. Ten people died and dozens are still missing inside their destroyed houses after a mass of rocks fell on a group of homes. The disaster may have been caused by heavy rain which has battered the Azores. [Photo by AFP]

Figure 1: This image was manually labeled as Outdoor, and the associated document was labeled as a Disaster document.

TIRANA, ALBANIA, 29-JUN-97: Albanian President Sali Berisha casts his vote at a central Tirana polling station on, June 29. Albania holds its general elections three months after the collapsed pyramid investment schemes drove the country into armed turmoil. [Photo by Petr Josek, Reuters]

Figure 2: This image was manually labeled as Indoor, and the associated document was labeled as a Politics document.

```
 1 # Determine category counts of words in corpus.
 2 for every document in first half of training set {
 3     current_category <-- category[document];
 4     for every word in document {
 5         count[word, current_category] += 1;
 6     }
 7 }
 8 # Determine bins for all known words (not just those in corpus).
 9 for every known word {
10     for every category {
11         category_count <-- count[word, category];
12         current_bin <-- bin(IDF[word], burstiness[word], category_count);
13         # Increment size of current bin.
14         size[current_bin] += 1;
15     }
16 }
17 # Determine count of every (bin, occurrence count) pair.
18 for every document in second half of training set {
19     current_category <-- category[document];
20     for every word in document {
21         category_count <-- count[word, current_category];
22         current_bin <-- bin(IDF[word], burstiness[word], category_count);
23         occurrence_count <-- term_frequency[word, document];
24         # Increment count for this word's (current_bin, occurrence count) pair.
25         count[current_bin, occurrence_count] += 1;
26     }
27     # Fix counts for occurrence counts of 0 (uses bin sizes) here.
28 }
29 # Estimate probability of every (bin, occurrence count) pair.
30 for every bin {
31     total = 0;
32     for every possible occurrence_count (0 to MAX) {
33         total += count[bin, occurrence_count];
34     }
35     for every possible occurrence_count (0 to MAX) {
36         probability[bin, occurrence_count] = count[bin, occurrence_count] / total;
37     }
38 }
39 # Calculate term weights.
40 for every bin {
41     for every possible occurrence_count (0 to MAX) {
42         lambda[bin, occurrence_count] = log (probability[bin, occurrence_count]);
43     }
44 }
45 # Loop through test set.
46 for every document in test set {
47     for every possible category {
48         score[category] = 0;
49     }
50     for every word in document {
51         occurrence_count <-- term_frequency[word, document];
52         for every possible category {
53             category_count <-- count[word, category];
54             current_bin <-- bin(IDF[word], burstiness[word], category_count);
55             score[category] += lambda[current_bin, occurrence_count];
56         }
57     }
58     # Assign document to category with highest score here.
59 }
60 # Various results are computed and displayed here.
```

Figure 3: This pseudo-code represents the algorithm used to conduct an entire experiment, including training (lines 1 - 44) and testing (lines 45 - 60).