Chapter 7

USING COMPRESSION-BASED LANGUAGE MODELS FOR TEXT CATEGORIZATION

William J. Teahan
School of Informatics
University of Wales, Bangor
Bangor, Gwynedd LL57 1UT, Wales, UK
wjt@informatics.bangor.ac.uk

David J. Harper
School of Computing
The Robert Gordon University
Aberdeen AB25 1HG, Scotland, UK
djh@scms.rgu.ac.uk

Abstract Text compression models are firmly grounded in information theory, and we exploit this theoretical underpinning in applying text compression to text categorization. Category models are constructed using the Prediction by Partial Matching (PPM) text compression scheme, specifically using character-based rather than word-based contexts. Two approaches to compression-based categorization are presented, one based on ranking by document cross entropy (average bits per coded symbol) with respect to a category model, and the other based on document cross entropy difference between category and complement of category models. Formally, we show the equivalence of the latter approach to two-class Bayes classification, and propose a method for performing feature selection within our compression-based categorization framework.

An extensive set of experiments on a range of classification tasks is reported. These tasks in increasing order of difficulty are language and dialect identification, authorship ascription, genre classification and topic classification. The results show that text categorization based on PPM is extremely effective for language and dialect identification, and for authorship ascription. PPM-based categorization is very competitive for genre and topic categorization compared with other reported approaches.

W.B. Croft and J. Lafferty (eds.), Language Modeling for Information Retrieval, 141–165.

Keywords: text categorization, language modeling, text compression

1. Background

Text categorization is the problem of assigning text to any of a set of prespecified categories. It is useful in indexing documents for later retrieval, as a stage in natural language processing systems, for content analysis, and in many other roles (Lewis, 1994).

The traditional machine learning approach to text categorization is based on a four step process (Dumais et al., 1998): first, performing word and sentence segmentation on the training files; second, performing feature selection on the word counts; third, applying a machine learning algorithm; and last, applying the learned model to the test data produced after performing the same segmentation and feature feature selection process that was applied in steps one and two. A wide variety of machine learning algorithms have been applied based on this approach, including: probabilistic Bayesian classifiers, decision trees, neural networks, multivariate regression models, symbolic rule learning and support vector machines (Lewis and Ringuette, 1994; Apte et. al., 1994; Schutze et al., 1995; Weiner et al., 1995; Cohen and Singer, 1996; Yang and Pederson, 1997; 1998; Joachims, 1998; Yang, 1999; Dumais et al., 1998).

Traditional approaches to text categorization share several problems: the need to perform feature extraction before processing; the need to define where word boundaries occur; and the need to deal uniformly with morphological variants of words (Frank at al., 2000). The problem of word and sentence segmentation is an especially vexing one, and indeed for some languages, the Western notion of a word may be inappropriate (Teahan et al., 2000).

We wish to use language models developed for text compression as the basis of a text categorization scheme and potentially for other applications in Information Retrieval (although the latter will not be discussed here). The motivation for using these models is that they are well grounded in information theory, and that they also have proven performance at many Information Retrieval related tasks such as word segmentation (Teahan et al., 2000), text mining (Witten et al., 1999), language/dialect identification and authorship attribution (Teahan, 2000).

For text categorization, our compression-based language modeling approach has several advantages. By using models that are character-based, not word-based, we can avoid the segmentation issue altogether. Additionally, we can sidestep to some extent the issue of which features to use by examining potential features within the framework of standard Markov-based approximations commonly used in language modeling.

Anecdotal evidence indicates that the idea of using compression for text categorization has been re-invented many times (Frank at al., 2000). However,

compression-based approaches to date have produced inferior results compared to state-of-the-art schemes based on traditional machine learning techniques more reliant on feature extraction (Frank at al., 2000; Teahan, 2000). Indeed, Frank et al. state that it is their belief that compression-based approaches will not be capable of competing with the machine learning approaches. In this paper, we seek to allay this mis-apprehension.

The main contribution of this work is threefold. Firstly, we demonstrate that compression-based language models work well for a number of categorization tasks. Secondly, in contrast to Frank et al., we show how a variation of the compression-based approach can in fact produce competitive results for genre and topic categorization compared with the traditional machine learning methods. And thirdly, we propose a method for performing feature selection within a compression-based framework, and show how it can lead to improved performance in certain circumstances.

This paper is organized as follows. The next two sections provide the theoretical background to our approach showing its basis in information theory and Bayesian classification. Section 4 briefly describes the particular text compression scheme (PPM) that we use in our experiments with the results discussed in Section 5. Further discussion is provided in the final section.

2. Compression models

As stated, we wish to use compression-based language models as the basis of a text categorization scheme. The crux of our reasoning hinges on the notion of *entropy* as a measure of the "information content" of a message (Shannon, 1948), and importantly that text compression can be used directly to estimate an upper bound to it (Brown et. al., 1992). Various treatments of entropy are given in (Shannon, 1948; Brown et. al., 1992; Manning and Schutze, 1999) and these are drawn upon in the following discussion.

Formally, suppose we have a language defined by an alphabet of k symbols s_i, and a probability distribution over these symbols $p(s)$. The fundamental coding theorem (Shannon, 1948) states that the lower bound on the average number of bits per symbol needed to encode messages of the language (i.e. sequences of the symbols) is given by the entropy of the probability distribution (and often referred to as the "entropy of the language"):

$$H(language) = H(p) = -\sum_{i=1}^{k} p(s_i) \log_2 p(s_i). \tag{7.1}$$

The information content of a particular symbol is given by $-\log_2 p(s_i)$, and is the number of bits required to encode that symbol. Thus, $H(p)$ can be viewed as an expectation over the number of bits per symbol. Hence, we can obtain minimum code lengths, or alternatively maximal compression rates, by encoding symbols in accordance with the actual probability distribution $p(s)$.

Generally, we do not have access to the actual probability distribution underlying a language, but rather some model of that language. Suppose we have a language, L, in which sequences of symbols $x_1, x_2, \ldots, x_n$ (denoted x_{1n} hereafter) are generated according to probability distribution $p(x)$, for which we have constructed a model $p_M(x)$. The *cross-entropy* of the language with respect to the model provides a measure of the extent to which the model departs from the actual distribution, and is given by:

$$H(L, p_M) = -\lim_{n\to\infty} \frac{1}{n} \sum_{x_{1n}} p(x_{1n}) \log_2 p_M(x_{1n}). \tag{7.2}$$

The cross-entropy is the average number of bits per symbol required to encode language L using the model p_M, and thus $H(L, p_M)$ is an upper bound on the entropy of the language, and $H(L) \leq H(L, p_M)$. $H(L, p_M)$ can be approximated by observing that Equation 7.2 is simply the expected value of $-\log_2 p_M(x_{1n})$ over an infinite sample of language examples, and we approximate it using one very large sample as follows:

$$H(L, p_M) \approx -\frac{1}{n} \log_2 p_M(x_{1n}), \quad n\ very\ large. \tag{7.3}$$

For a given document, D, we can compute the *document cross-entropy*, the average number of bits per symbol to encode the document using the model as:

$$H(L, p_M, D) = -\frac{1}{n} \log_2 p_M(D), \quad D = x_{1n}. \tag{7.4}$$

Suppose that p_M is a good model for L, then the lower the document cross-entropy, the more likely it is that the document D is generated by the language. This key observation lies at the heart of using text compression for text categorization. Categories are viewed as languages, and documents are compressed using models constructed over the categories.

3. Bayes Classifiers

We would like to establish the relationship between Bayes classifiers, and using cross-entropy for categorization, and in that context discuss feature selection for compression models.

Suppose we want to classify documents such that they are assigned independently to one or more categories. Assuming we have a training collection of categorized documents, then for any given category, C, we can construct a model for that category p_C, and model for its complement p_N (hereafter the "negative" model). Bayes formula gives us a mechanism for deciding whether

a particular document, D, belongs to category C, and using the more familiar notation $P(D|C) = p_C(D)$ and $P(D|\overline{C}) = p_N(D)$, decide D belongs to C if:

$$P(D|C)P(C) > P(D|\overline{C})P(\overline{C}) \tag{7.5}$$

where $P(C)$ is the prior probability of a document belonging to C. Equivalently, the documents can be ranked with respect to the category using the following expression:

$$\log_2 \frac{P(D|C)}{P(D|\overline{C})} \tag{7.6}$$

and a cutoff is selected, which gives optimal categorization, according to some measure of categorization goodness.

Equation 7.6 has a dual formulation in the cross-entropy and compression world, namely

$$\begin{aligned} &\log_2(P(D|C)/P(D|\overline{C})) \\ &\quad \equiv -\log_2 P(D|\overline{C}) - (-\log_2 P(D|C)) \\ &\quad \equiv -\log_2 p_N(D) - (-\log_2 p_C(D)) \\ &\quad \equiv H(\overline{C}, p_N, D) - H(C, p_C, D). \end{aligned} \tag{7.7}$$

The inverse relationship between cross-entropy, and probability of a given document, is well-known. Thus, maximizing the cross entropy difference (i.e. the code length difference) between a document encoded using the negative model and category model respectively, is equivalent to constructing an independent Bayes classifier for each category. Frank et al. provided a similar analysis in (Frank at al., 2000). In applying text compression to categorization, we will determine an optimal cutoff for the cross entropy difference computed in Equation 7.7 on a category-by-category basis, which will be referred to as the *category cutoff*.

More interestingly, this relationship suggests ways in which feature selection might be incorporated within text compression models when applied to categorization. The need for feature selection was highlighted by Frank et al.—they found that in some cases the occurrence of just a few words determines whether an article belongs to a category or not. Although one might expect an information-theoretic measure to recognize and discount features with poor predictive value, Frank et al. did not propose doing feature selection within a compression framework.

Our method for performing feature selection is as follows. Features are selected for Bayes classification on the basis of their discrimination power, which

is computed based on the relative density of the feature in the category compared with its density in the complement (or similar). Likewise, we propose selecting features that exceed some cross entropy difference threshold. Hence, features will be ranked based on their potential contribution to the overall cross entropy difference (as given by Equation 7.7). Again, we will apply feature selection on an individual category basis, and the resultant cutoff will be referred to as the *feature cutoff*.

A benefit of performing thresholding (i.e. category and feature cutoff in our case) noted in (Frank at al., 2000) is that it automatically adjusts for the fact that the models p_C and p_N are formed using different amounts of training data. This is a potential problem since in general, one expects to achieve better compression with more training data.

4. PPM-based language models

The Prediction by Partial Matching (PPM) text compression scheme has consistently set the standard in lossless compression of text since it was originally published in 1984 by (Cleary and Witten, 1984). Other methods such as those based on Ziv-Lempel coding are more commonly used in practice but their attractiveness lies in their relative speed rather than any superiority in compression (Bell, Cleary and Witten, 1990). Although adequate categorization performance can be achieved by Ziv-Lempel methods in restricted circumstances (Benedetto et al., 2002), experiments show that they are significantly outperformed by other compression methods (Khmelev, 2000; Kukushkina et al., 2001).

Compression models encode a document according to a given model, and the resultant cross-entropy is given by Equation 7.4. In the case of the PPM family of models, the individual symbols are encoded within the context provided by the preceding symbols appearing in a document. Taking Equation 7.4 as a starting point, we can achieve optimal compression for the model p_M, by observing that:

$$
\begin{aligned}
& H(L, p_M, D) \\
& \quad = -\tfrac{1}{n} \log_2 p_M(D), \quad D = x_{1n} \\
& \quad = -\tfrac{1}{n} \log_2 \textstyle\prod_{i=1}^{n} p_M(x_i | context_i) \quad [by\ Chain\ Rule] \\
& \quad = \tfrac{1}{n} \textstyle\sum_{i=1}^{n} -\log_2 p_M(x_i | context_i)
\end{aligned}
$$

where $context_i = x_1, x_2, \ldots x_{i-1}$. Thus each symbol is encoded according to its *information content* within the context provided by all the preceding symbols. In practice, PPM uses a Markov approximation and assumes a fixed order context. A fixed order context of five is found to perform well, and increasing

it further does not generally improve compression (Cleary and Witten, 1984, Teahan, 1998).

PPM uses an approximate blending technique called *exclusion* based on the *escape* mechanism to exclude lower order predictions from the final probability estimate. This includes an order 0 model which predicts symbols based on their unconditioned probabilities, and a default model which ensures that a finite probability (however small) is assigned to all possible symbols. Prediction probabilities for each context in the model are calculated from frequency counts, and the symbol that actually occurs is encoded relative to its predicted distribution.

The PPM method is character-based, although the blending mechanism can equally be applied to other classes of symbols such as words and parts of speech. Compression experiments with a wide range of English text (Teahan, 1998) show that word-based models consistently outperform the character-based methods, although the difference in performance is only a few percent. A fuller description of the PPM algorithm can be found in (Bell, Cleary and Witten, 1990); two variants of the algorithm that perform well in compression experiments are PPMC (Moffat, 1990) and PPMD (Howard, 1993). These methods differ in the way they estimate the "escape" probability when a model uses a lower order context to make a prediction. PPMC uses the number of times a novel symbol has occurred before as the basis for this probability. PPMD is a slight variation of PPMC that usually provides a small but noticeable improvement in prediction in many cases.

Feature selection In contrast to previous compression-based approaches to text categorization, we seek to investigate the possibility of incorporating feature selection in some manner. In our approach, when we apply feature selection using PPM models, we determine on a category-by-category basis an optimal cutoff by only selecting features that exceed some code length difference threshold $h_N - h_C$ where the feature code lengths are $h_N = -log_2 p_N(x_i|context_i)$ and $h_C = -log_2 p_C(x_i|context_i)$.

5. Experimental results

Experimental results are given for different categorization tasks in the next sections.[1] The tasks (language identification, authorship attribution, genre categorization and topic categorization) are arranged in order of increasing difficulty. The first three tasks use cross-entropy to rank the documents, whereas the last task uses the cross-entropy difference measure defined in Equation 7.7.

[1] Part of the work described here has already been published as an extended abstract in (Teahan, 2000).

The results show that the PPM-based categorizer is extremely effective both for language identification and for authorship attribution. Experiments with the latter show that, perhaps surprisingly, results are excellent even when testing for authorship of similar styles. Very competitive results are also obtained for the more difficult tasks of genre and topic categorization.

We wish to stress that for all of these experiments *no* lexical preprocessing was done to the texts beforehand (this includes no case conversion or whitespace substitution). An exception was that, in some cases, certain header information was removed from the documents (such as headers in Usenet news articles) when it was deemed that the information might unfairly affect categorization performance. The technique we adopt is purely character-based and does not require any a-priori knowledge about the representation of the documents—the methods would work just as well, for example, with UTF-encoded Unicode symbols.

5.1 Language identification

Language identification concerns the problem of identifying from examples of written or spoken language which language is being used. Dunning (1994) describes several statistical methods for identifying the language of small pieces of text (such as *e pruebas bioquimica* and *faits se sont produits*). Ganeson and Sherman (1993) adopt various statistical modelling methods to perform various language recognition problems such as: recognizing a known language; distinguishing a known language from uniform noise; and detecting a non-uniform unknown language. Cross-entropy calculated using PPM-based models may equally be applied to each of these problems as it provides the means to rank the performance of statistical models. The text being analysed is compressed using PPM character based language models trained on representative texts written in various languages. The most likely language is the one used to train the model that has the best compression performance.

In an experiment, a selection of Web documents were collected for sixteen different languages using the search engine `www.google.com`. A diversified pool of text of at least 100,000 bytes was obtained for each language by a three step process: firstly, restricting the search to the specified language; secondly, entering a general query (for example, containing frequently-used terms such as "one" for English or "oui" for French); and thirdly, concatenating a representative selection of the highest ranked documents returned by the search engine after first eliminating those (relatively few) documents whose language was designated incorrectly (for example, when the document contained a mixture of languages). These texts were then split into two 50,000 byte texts, one used for training and the other for testing. The testing text was further split into different sized segments, each from 50 bytes up to 500 bytes. The training text

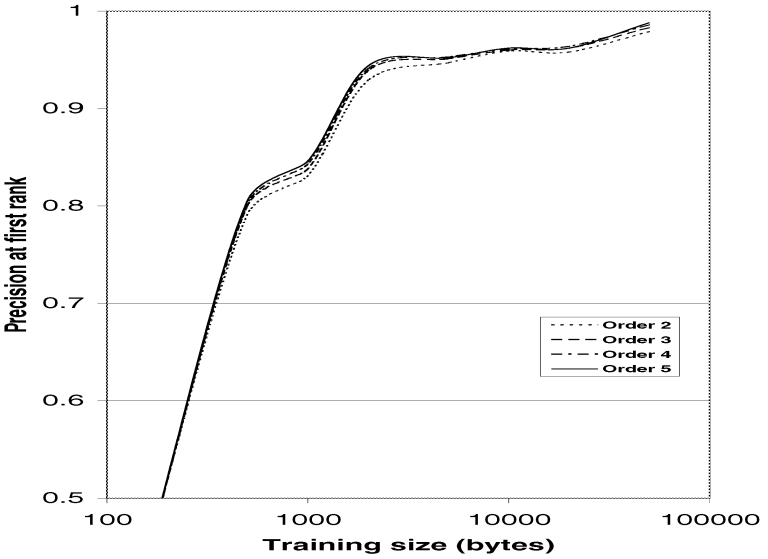

Figure 7.1. Precision at first rank at identifying the language of Web documents for different order PPM models trained on various sized training texts.

was used to train PPM models for each of the different languages, and each segment of testing test was then ranked (in ascending order) by cross-entropy with respect to these models.

Results for different order PPM models on the 500 bytes segments are graphed in Figure 7.1. The vertical axis plots the precision of the first-ranked documents at determining the correct language, whereas the horizontal axis plots on a log scale the size of the training text in bytes ranging from just over 100 bytes up to the full 50,000 bytes that was available. The graph shows that surprisingly very little training text is required (around 2,500 bytes) to achieve a high degree of precision (over 0.95) and that different order models do not have a significant affect on the precision.

Table 7.1 provides further details of results for the different languages for varying sized testing segments (50, 100, 200 and 500 bytes). The rightmost column lists the languages that were ranked second for the 500 byte case. For example, Polish was chosen as the 2nd ranking for the Czech 500-byte test segments in 78% of cases. The results provide an indication of which languages are similar to each other. For example, English is most similar to German (57% of English segments have German as their 2nd ranking), but German is most similar to Dutch (85% of German segments have Dutch as their 2nd ranking). The table shows that the best results are obtained on 500-byte segments with 50% of the languages achieving a precision at first rank value of 1.0, but very

Language	Precision at first rank				Most frequent 2nd-ranked languages (with percentages)
Length of testing text in bytes	50	100	200	500	500
Czech	0.976	0.994	0.996	1.000	Polish (78), Italian (9)
Danish	0.905	0.950	0.972	0.990	Norwegian (99)
Dutch	0.988	0.994	1.000	1.000	Norwegian (61), German (28)
English	0.933	0.980	0.992	1.000	German (57), Portuguese (30)
Finnish	0.935	0.952	0.960	0.970	Swedish (82), German (9)
French	0.937	0.966	0.976	0.990	Italian (56), Portuguese (34)
German	0.959	0.984	0.992	1.000	Dutch (85), Norwegian (10)
Hungarian	0.936	0.960	0.976	0.990	Portuguese (59), Czech (29)
Icelandic	0.939	0.960	0.968	0.970	Norwegian (75), Swedish (2)
Italian	0.984	0.996	0.996	1.000	Portuguese (59), Spanish (15)
Norwegian	0.829	0.910	0.936	0.920	Danish (87), Swedish (4)
Polish	0.999	1.000	1.000	1.000	Czech (100)
Portuguese	0.945	0.974	0.984	1.000	Spanish (92), Italian (5)
Rumanian	1.000	1.000	1.000	1.000	Italian (63), Portuguese (36)
Spanish	0.796	0.876	0.920	0.960	Portuguese (96)
Swedish	0.959	0.978	0.988	0.990	Norwegian (93), Danish (5)
Average	0.939	0.967	0.979	0.986	

Table 7.1. Various results for PPM-based language identifier on Web documents.

good results are also possible even with 50-byte segments with only Spanish and Norwegian having a precision less than 0.9.

Dialect identification Another interesting possibility is the problem of identifying the dialect for a sample text. This is useful in many applications; in information retrieval, for example, we could index "English" text using an appropriate stemmer (English or American), which each then produces standardized stems. The Brown (Francis and Kucera, 1982) and LOB (Johansson, 1986) corpora represent diverse samples from two different dialects, American and British English; as such, they provide a means for rigorously testing how well PPM performs at dialect identification. In an experiment, equal-size partitions of 100,000 characters were successively extracted from the two corpora, and then compressed using order 5 PPMD models trained on the remaining text from each (i.e. all the text in the corpora with just the partition deleted).

The Brown-trained model compressed the Brown corpus partitions up to 3.6% better than did the LOB-trained model. Correspondingly, the LOB-trained model compressed the LOB-partitions up to 3.1% better than did the Brown-trained model. There were no cases in which a model performed worse on its respective partitions, and in only three cases was the performance the same for a particular partition. Consequently, these results show it is possible

to classify "English" text as American or British English using PPM models with a high degree of accuracy.

5.2 Authorship attribution

A similar PPM-based process to determine the cross-entropy is possible for the task of authorship attribution. It is widely known that style and statistical properties differ markedly for different authors (Boggess, Argawal and David, 1991). This can be exploited by training models using text written by each of the disputed authors, and then compressing each text in question using all these models. The hope is that the style of each author will be sufficiently different to clearly distinguish between them. Experiments with English texts reported here show that this is true in most cases.

The following two problems are investigated in this section—determining known authorship; and determining disputed authorship. The known authorship experiments described in this section use the Reuters-21578 collection. This collection is publicly available[2] and contains Reuters news articles on various topics. It is a standard data set used for text categorization research. Each article has had specific topic labels (such as `earnings` and `trade`) assigned to it manually by Reuters employees. This collection is usually used for research on what we term "topic categorization", and indeed, we use the collection for just such a purpose in section 5.4. In this section, however, we describe a quite different (and novel) use for the collection. Many of the articles in the collection also have author labels assigned to them (such as `Rich Miller` and `Patti Domm`). Various information concerning the most frequently found of these labels are listed in Table 7.2. The table lists information only for articles of single authorship—articles with multiple authorship were not considered for these experiments. The second column in the table lists the number of documents that were found for each author. For example, `Peter Torday` was designated as the author of 24 articles in the collection. The last column in the table provides a list of topics that were assigned to the same articles. As you would expect, authors write articles on similar topics, and this is reflected in the number of multiple occurrences shown in the list. Nine articles authored by `Cal Mankowski`, for example, primarily concern the topic `acq` (acquisitions), and only one concerns `earn` (earnings). (Many of the authored articles have no topic assigned to them which explains why the number of assigned topics is less than the document frequency listed in the second column.)

We use this data to determine how well our PPM-based categorizer does at the problem of determining known authorship. In an experiment, pools of

[2] `http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html`

Author	Doc. freq.	Text size (bytes)	List of assigned topics (with number of occurrences if > 1 shown in brackets)
Peter Torday	24	98550	dlr (4), gnp, interest (2), money-fx (5), trade (2), yen
Rich Miller	23	71064	gnp (2), interest (2), money-fx (2), trade (7), yen
Patti Domm	23	83060	acq (18), crude
Chaitanya Kalbag	23	74546	coconut, coconut-oil, copra-cake, earn, gnp (2), meal-feed, money-fx, palm-oil, veg-oil, sugar
Cal Mankowski	18	60453	acq (9), earn
Mark O'Neill	15	62073	bop, grain (2), sugar (2), trade
Alan Wheatley	15	60207	money-fx, money-supply (2), interest

Table 7.2. Various information concerning the most frequent authors found in the Reuters-21578 collection.

text were obtained by concatenating articles written by the same author together. The size of these texts is shown in column three of Table 7.2. These were then split into segments of 250 lines (approximately 14000 bytes each) and alternately used for testing and/or training purposes on a rotating basis for cross-validation (ensuring that testing and training data were disjoint in all cases). PPM models were constructed for each author from the training data, and used to categorize each of the testing segments using the cross-entropy ranking method as before. Out of the resulting 42 tests that were made, just two were incorrectly labelled, a precision at first rank of 95%. One segment written by `Alan Wheatley` was mis-categorized as belonging to `Peter Torday`, and one by `Cal Mankowski` was mis-labelled as `Patti Domm`'s—in both these cases, the list of assigned topics between the two authors noticeably overlap (for example, articles by both `Patti Domm` and `Cal Mankowski` concentrate on the acquisition topic).

Another interesting possibility concerns the problem of determining unknown or disputed authorship. A famous problem in authorship attribution concerns the Federalist Papers written by Alexander Hamilton, John Jay and James Madison (Mosteller and Wallace, 1984). Between the years 1787–1788, eighty-five short essays addressed to the citizens of New York on the U.S. Constitution were published in newspapers under the pseudonym "Publius". Of these papers, it is generally agreed that Jay wrote 5, Hamilton 43, Madison 14 with the authorship of the remaining 12 being in flat dispute between Hamilton and Madison.

Mosteller and Wallace (1984) determine odds of authorship for the papers by applying Bayes' theorem to evidence based upon 30 common words (such as *as* and *upon*). Their work supports the claim made by historians (and by Madison himself) that Madison wrote the disputed papers.

Disputed papers

No.	Madison (bpc)	Hamilton (bpc)	No.	Madison (bpc)	Hamilton (bpc)
49	**1.79**	1.93	55	**1.86**	1.97
50	**1.92**	2.07	56	**1.70**	1.85
51	**1.72**	1.88	57	**1.85**	1.98
52	**1.78**	1.94	58	**1.80**	1.93
53	**1.79**	1.93	62	**1.83**	1.84
54	**1.73**	1.89	63	**1.82**	**1.82**

Papers known to have been written by Madison			*Papers known to have been written by Hamilton*		
No.	Madison (bpc)	Hamilton (bpc)	No.	Madison (bpc)	Hamilton (bpc)
44	**1.76**	1.90	59	**1.82**	1.88
45	**1.67**	1.81	60	1.78	**1.71**
46	**1.78**	1.85	61	1.78	**1.73**
47	**1.70**	1.81	65	1.87	**1.82**
48	**1.85**	1.99	66	1.80	**1.75**

Table 7.3. Ascribing authorship to the Federalist Papers.

Experiments with PPM character based computer models also supports this claim.[3] Each of the disputed papers was compressed using computer models trained on text taken from the non-disputed papers. The same models were also used to test papers with known authorship. To minimize the fluctuations due to differing periods, the papers tested are all in consecutive order from paper number 45 through to 66 (excluding paper number 64 written by John Jay). Table 7.3 lists the results obtained. Compression ratios (listed in bits per character or "bpc") for Madison are much lower for the disputed papers (excluding 62 and 63), supporting the claim that Madison was the primary author for those papers. For the non-disputed papers, only one paper (59) has a result which is contradictory. Mosteller and Wallace state that historians feel least settled about papers 62 and 63, which is reflected in the results. They draw attention to the difficulty of the task confronting researchers—both Hamilton and Madison had remarkably similar prose styles, which was no unique phenomenon as for most educated Americans of the late eighteenth century they employed "the same stylistic devices, the same standard phrases, and remarkably similar sentence structure." These results are also surprising in that only a relatively small amount of training text (a few hundred kilobytes) was required to train the models.

[3]Similar results using compression models to tackle the Federalist Papers authorship problem have been reported by Juola (1998) who used a different method of calculating the cross-entropy.

5.3 Genre categorization

Genre categorization is the problem of identifying the subject domain of a document. Some sample genres are: politics, religion, history, sport and science. A number of studies (Biber (1988); Kessler et al., 1997; Wolters and Kirsten, 1999; Stamatatos et al., 2001) have investigated this problem usually by adapting methods found suitable for the related problem of topic categorization (see next section). The terms "domain categorization" and "genre categorization" are often used confusedly in the literature—most take the approach we adopt here of defining a "genre" to be a domain subject area such as history and sport, although Lee and Myaeng (2002) investigate the slightly different problem of identifying the style of the text that characterizes the purpose for which the document has been written (such as a research article, novel, poem, news article, editorial, home page, advertisement, manual etc.). Furthermore, many other researchers have not distinguished between genre and topic categorization (see for example, McCallum & Nigam, 1998), although clearly each task may present its own problems that need to be investigated independently. As a case in point, the task of identifying *sentiment*, or the overall opinion that the author of a document has to its subject matter, has highlighted the need for more sophisticated techniques than the traditional machine learning methods adopted for topic categorization which are found to perform poorly (Pang and Lee, 2002).

An investigation into using PPM for genre categorization was performed using the `Newsgroups` data set (McCallum & Nigam, 1998). This collection contains 20,000 articles evenly divided among 20 Usenet discussion groups. This data set is particularly interesting as many of the newsgroup categories fall into confusable clusters: for example, three discuss religion (alt.atheism, soc.religion.christian, talk.religion.misc), three discuss politics (from the talk.politics.* hierarchy) and five are from the comp.* hierarchy. In an initial experiment, 20 separate order 5 PPMD models were trained on the text contained in the first 80% of the articles, and the remaining 20% of the articles were compressed against each of them to see which model performed best. The Usenet headers (including the subject, followup and newsgroup lines) were removed from each of the articles beforehand. By assigning the newsgroup of the top-ranked model as the newsgroup of the article, the newsgroup category was correctly deduced in 82.1% of the articles, which compares favourably with the results reported by McCallum & Nigam (1998) who achieved a precision at first rank accuracy level of 74% using a multivariate Bernoulli event model, and 85% using a multinomial model using the same training/testing split of the data.

A confusion matrix of the errors made by the PPM-based classifier for ten of the newsgroups is shown in Table 7.4. The columns show results for

	aa	*sc*	*se*	*sm*	*ss*	*src*	*tpg*	*tpme*	*tpmi*	*trm*
aa	**149**				2	4		1	2	40
sc		**177**	3	3	1	1	4		2	
se	2	3	**136**	13	4			1	3	
sm	2	1	4	**165**	3	1	2		6	2
ss		1	3	1	**172**		1	1	8	5
src						**191**	1			3
tpg	1			2	1	2	**171**		11	9
tpme	1					5		**185**	6	1
tpmi	1				2	1	19	12	**139**	26
trm	35	1			2	9	5	1	6	**141**

Table 7.4. Cross-entropy confusion matrix for the **Newsgroups** data set. The columns show results for each model trained on text taken from the respective newsgroup; rows show test results for each newsgroup based on articles not included in the training data. Only 10 (of the 20) newsgroups are shown. These are: alt.atheism (i.e.*aa*), sci.scrypt, sci.electronics, sci.med, sci.space, soc.religion.christian, talk.politics.guns, talk.politics.mideast, talk.politics.misc and talk.religion.misc.

each model trained on text in the respective genres; rows show test results for each genre based on text not included in the training data. As is typical for a PPM-based classifier, the leading diagonal correctly accounts for most of the categorizations. For most of the newsgroups, there are only a few outliers, the major exception between the confusion between the alt.atheism and talk.religion.misc newsgroups, which seems a reasonable confusion to make, and which accounts for 11% of the overall mis-categorizations. For many of the mis-categorizations, the correct newsgroup category has been ranked second or third-best, with only a small difference in cross-entropy from the best. Choosing the best two improves accuracy to 91.5%, and the best three to 94.2%. However, each test article is now multiply classified, a penalty that some retrieval-based applications may be willing to pay for the increased accuracy. This penalty can, however, be significantly reduced by choosing to discard the second or third best categories if the difference in cross-entropy from the best is above some threshold. For example, examining the best three categories and choosing a difference threshold of as little as 0.2 bits per character in cross-entropy will discard nearly 62% of the multiple categorizations while still ensuring a very high accuracy level of 90.9%.

5.4 Topic categorization

The task of topic categorization concerns the problem of assigning one or more categories to a document from a list of pre-defined categories where the categories reflect the "topics" or subjects the document is concerned with. The

Category	Number Training Documents	Number Testing Documents	Training size (bytes)	Avg. size of document (bytes)
earn	2877	1087	1488519	517.4
acq	1650	719	1330793	806.5
money-fx	538	179	623599	1159.1
trade	369	118	579607	1570.8
crude	389	189	532919	1370.0
grain	433	149	490372	1132.5
interest	347	131	335260	966.2
wheat	212	71	239843	1131.3
ship	197	89	218093	1107.1
corn	182	56	217246	1193.7

Table 7.5. Various statistics for the Reuters-21578 collection.

categories are likely to be more fine-grained than the broad categories used for genre classification.

The topic-based experiments described in this section all use the Reuters-21578 collection that was used in section 5.2. Using the ModApte split (Dumais et al., 1998), the collection is separated into a training set, testing set and an unused set (9603, 3299 and 7634 documents respectively). This collection comes with manually assigned categories for all the documents in the training and testing sets.

For the experiments, we have further split the training set into a training subset and a validation subset (6338 and 3266 documents respectively). We use the validation subset for choosing the category and feature cutoff values that maximizes the average of recall and precision (i.e. "breakeven" point) for each category based on the cross entropy difference ranking formula defined in Equation 7.7. Once the cutoff values have been obtained, the models p_C and p_N are rebuilt to take maximum use of the full training data in the same manner as (Frank at al., 2000).

Although there are 118 categories in the entire Reuters collection, our experiments only concentrate on the ten largest categories (since by common practice results are seldom published in the literature for the smaller categories). The categories are ordered by the size of training data available, in bytes: *earn, acq, money-fx, trade, crude, grain, interest, wheat, ship* and *corn*. This cuts the validation subset and testing set down to 2152 and 3019 documents respectively. Table 7.5 lists the number of training and testing documents available for each category; also listed is the total size of the training data in bytes after concatenating all the training files together once irrelevant text (including tags) has been removed. The average size per document is shown in the last column. The figures show that the distribution of the collection is highly skewed—there

Category	Order 2	Order 3	Order 4	Order 5	Order 6
earn	0.964	0.970	0.967	0.947	*0.953*
acq	0.942	0.954	0.959	*0.954*	0.953
money-fx	0.838	0.835	*0.816*	0.834	0.810
trade	0.776	0.772	0.759	0.778	*0.772*
crude	0.837	*0.888*	0.878	0.867	0.857
grain	0.870	0.905	0.915	*0.899*	0.911
interest	0.749	0.735	0.723	0.728	*0.759*
wheat	0.721	0.717	0.706	*0.735*	0.757
ship	0.757	*0.783*	0.750	0.771	0.765
corn	0.635	0.666	0.668	0.669	*0.646*
Weighted average	0.899	0.910	0.907	0.900	*0.901*

Table 7.6. Average recall/precision breakeven points for PPM-based categorization for the Reuters-21578 collection using PPMD models with feature cutoff.

Category	Na"ve Bayes	LSVM	PPMC (Order 2)	PPMD (Order 3)	
				without feature cutoff	*with feature cutoff*
earn	0.959	**0.980**	0.963	0.968	0.970
acq	0.878	0.936	0.910	0.950	**0.954**
money-fx	0.566	0.745	0.763	0.831	**0.835**
trade	0.639	0.759	0.650	0.734	**0.772**
crude	0.795	**0.889**	0.807	0.871	0.888
grain	0.788	**0.946**	0.746	0.883	0.905
interest	0.649	**0.777**	0.604	0.685	0.735
wheat	0.697	**0.918**	0.649	0.744	0.717
ship	0.854	**0.856**	0.819	0.749	0.783
corn	0.653	**0.903**	0.542	0.667	0.666
Weighted average	0.848	**0.919**	0.863	0.902	0.910

Table 7.7. Average recall/precision values for PPM-based categorization compared with various schemes on the Reuters-21578 collection.

is a varying amount of training data available from nearly 1.5 Mb for the most frequent category *earn* down to just over 200 Kb for the smallest category *corn*; the sizes of the documents also vary noticeably, with just over 500 bytes for *earn*, to over three times that for *trade*.

Table 7.6 shows the average precision/recall breakeven points for our PPM categorizer that are achieved on the Reuters-21578 collection at the testing stage when using the category and feature cutoffs obtained from the validation stage. The italic values indicate the orders that performed the best at the validation stage. Shown at the bottom of the table is the average of the top ten categories weighted according to their frequency of occurrence in the testing set. From Table 7.6, we can see that no one context order is particularly predominate at achieving the best average precision/recall, with the best values

scattered across all orders. These results show that each category has features that make a particular context order most effective for it. This is not too surprising when we consider the language used in some of these categories. Considering *earn, acq, money-fx* and *trade*, these categories quite often use acronyms that are only two or three characters long, or contain numerical data, and therefore are best served with models that have a context order different from the order 5 models found most effective for general English text. Also, higher orders (4 to 6) feature strongly, which closely match well-documented results from standard compression experiments on text-like data (Bell, Cleary and Witten, 1990; Teahan, 1998). The overall results, however, are sufficiently similar for orders ≥ 3 that we can regard it as stable, and arbitrarily pick one—we have chosen order 3 for subsequent experiments.

Table 7.7 compares the order 3 model results from Table 7.6 (reproduced in the last column) with results published in (Dumais et al., 1998) for Na"ve Bayes and linear support vector machines (LSVMs), and with previously published results for PPM. Results in the last two columns are new results using our models. (The best result for each category is shown in bold font). The results show that our approach is uniformly better than Na"ve Bayes (except for *ship*), and performs better than LSVMs in three out of the ten categories (*acq*, *money-fx* and *trade*) and are competitive in a further four categories (*crude*, *earn*, *grain* and *interest*). Lack of training data clearly affects performance for the three poorest performing categories, *corn*, *ship* and *wheat*, which correspond to the three smallest categories from Table 7.5.

Shown in the third column of Table 7.7 are the results reported by Frank et al. who used order 2 PPMC models (Frank at al., 2000). In their method, they used a single category cutoff for optimizing precision/recall; this is clearly improved across nearly all categories using the order 3 PPMD models and the feature cutoff approach that we adopt. They also found that models higher than order 2 achieved degraded performance in almost all cases. They attributed this to the amount of training data being insufficient to justify more complex models. This is in direct contrast to the results that we have obtained—we have found that higher order models perform as well (and in some cases better). Unfortunately, Frank et al. did not include results for higher order models in their paper, so it is difficult to speculate as to the reasons for the variance between our results and theirs.

Results for order 3 PPMD models without feature cutoff are shown in the second to last column of Table 7.7. They show a consistent but slight improvement across all categories when using feature cutoff, except for *corn* (a very slight decrease) and *wheat*. Results for other orders show that feature cutoff leads to better performance for two categories in particular, *interest* and *money-fx*, but to inconsistent results for other categories (sometimes better and sometimes worse than when just applying category cutoff without feature cut-

		Relevant Documents (Percentage of occurrences)					*Non-relevant documents* (Percentage of occurrences)				
	$h_C \rightarrow$	< 1	1–2	2–4	4–8	≥ 8	< 1	1–2	2–4	4–8	≥ 8
interest	< 1	*43.91*	3.94	2.98	**0.91**	**0.07**	*33.95*	6.33	7.43	**5.01**	**1.25**
	1–2	*3.96*	4.23	3.01	**0.95**	**0.06**	*1.67*	3.11	3.90	**2.30**	**0.45**
	2–4	*2.26*	3.13	3.01	4.53	0.29	*0.68*	1.85	3.90	7.16	0.95
	4–8	*0.67*	0.75	3.01	9.03	1.18	*0.10*	0.30	3.90	9.30	2.15
	≥ 8	*0.06*	0.05	3.01	0.73	0.76	*0.00*	0.01	3.90	0.84	1.12
wheat	< 1	*37.56*	4.09	3.07	**1.05**	**0.07**	*32.15*	7.52	8.08	**5.63**	**0.88**
	1–2	*3.50*	3.10	2.84	**0.95**	**0.05**	*1.59*	3.05	4.05	**2.32**	**0.49**
	2–4	*2.28*	3.01	2.84	4.82	0.37	*0.54*	1.69	4.05	7.06	0.84
	4–8	*1.16*	1.24	2.84	11.83	1.79	*0.10*	0.31	4.05	9.15	1.84
	≥ 8	*0.06*	0.10	2.84	1.76	1.71	*0.00*	0.01	4.05	0.90	0.97
	$h_N \uparrow$										

Table 7.8. Percentage of occurrences within various ranges for symbol code lengths obtained when coding documents for the *interest* and *wheat* categories in the Reuters-21578 testing set.

off). However, these results are encouraging, and show that the approach is worthy of further investigation.

To investigate the effect of applying feature cutoff further, we analyzed the percentage of occurrences of the symbol (i.e. character) code length differences that were observed during coding of the relevant (category) and non-relevant (non-category) documents in the testing set. Table 7.8 summarizes the results for the two categories *interest* and *wheat* (these were chosen because they had the biggest positive and negative difference in values between the last two columns shown in Table 7.7). The columns in the table correspond to the category model's code length ranges ($h_C < 1; 1 \leq h_C < 2; \ldots$), and the rows to the negative model's code length ranges ($h_N < 1; 1 \leq h_N < 2; \ldots$) where h_C and h_N are the character code lengths (in bits) obtained when coding using the category and negative models respectively. Similar results were found for all categories, and not just the two shown in the table.

The results highlighted in bold and italics font in the table show a marked difference between the relevant and non-relevant documents in the observed code lengths. The results in italics are when the category models are performing very well (i.e. $h_C < 1$)—as shown, this occurs much more frequently for the relevant documents. The figures in bold font show a similar contrast—these occur when the category models are performing poorly (h_C is high) and the negative models are doing relatively well (h_N is low). This is likely to occur due to lack of training when a feature is unobserved in the relevant training documents but is common in the non-relevant ones (such as a sequence of text that occurs frequently in standard English, say).

These results are likely to have a significant impact on the overall decision, even though the number of occurrences of these cases ranges between 5 and

10% in total. This is because these code length values are high, whereas for a substantial proportion of the values, the difference in code lengths is small (i.e. the table shows that 30 to 40% of the values occur when both h_C and h_N are < 1). Of particular interest is the over one per cent difference in percentage for *interest* observed for the case when $h_C \geq 8$ and $h_N < 1$. In general, we found that the feature cutoff values chosen for each category varied between -4 and -8 (i.e. when the category model was performing very poorly compared to the negative model). The primary effect of this, as evidenced by the results in the table, is to discard features which are "suspiciously" rare in the non-relevant documents.

6. Discussion

In this paper, we have presented the results of a study in the use of text compression models for text categorization. Compression models are firmly grounded in information theory, and we have exploited this theoretical underpinning in applying text compression to categorization. We have argued that document cross-entropy (the average number of bits per symbol to encode the document using the model) provides an excellent ranking measure for many categorization tasks. We have demonstrated that another compression-based approach, namely maximization of the recall/precision average based on the cross entropy difference between category and complement of category models, leads to improved performance for the specific task of topic categorization. We have also shown that this second method is equivalent to a two class (category, complement of category) Bayes classification.

We have argued that the PPM-family of compression models, and in particular the character-based variants, have a number of advantages over word-based approaches. Problems of text segmentation (determining word boundaries), normalization such as stemming, and to a lesser extent word sense disambiguation, can be largely avoided. Instead, the PPM model discovers the important character contexts, which capture word boundary, morphology and word sense. Furthermore, we can apply our approach to languages where word segmentation is far more problematic. And, models can be constructed over very diverse texts, which may include coded and numeric data.

We have performed a comprehensive set of experiments on the use of compression based language models for the following categorization tasks: language identification, authorship and genre categorization. The experiments demonstrate high levels of categorization performance that show these techniques are useable in operational environments. For the language identification experiments, suitable Web training and testing data for 16 different languages were obtained using a popular search engine, and our PPM-based cross-entropy measure was able to identify the correct language in most cases, even

for small samples of testing text (50 bytes). We were also able to accurately distinguish between dialects (e.g. American and British English) using this method.

Two tasks were investigated for the authorship experiments—that of determining known and disputed authorship. We were able to make novel use of the Reuters-21578 collection to test how well our cross-entropy based method performed at the first task, and we found that the method was able to accurately predict the Reuters authors in all but two cases, the contrary results occuring for authors who had substantial overlap in their topics of interest. For the second task, we applied the cross-entropy method to a famous problem in authorship attribution—that of determining the authorship of the Federalist papers—and we arrived at the same conclusions as other studies on the subject.

Experiments were performed using the `Newsgroups` data set for the more difficult problem of classifying by genre. The performance of the PPM-based categorizor compares favourably with the results reported by McCallum & Nigam (1998) who used a multi-variate Bernoulli event model and a multinomial model. Biber (1988) uses factor analysis to identify six dimensions of variation on the basis of linguistic co-occurrence patterns across 23 spoken and written genres. An interesting possibility for future work in this area is to compare the effectiveness of the PPM based classifier at distinguishing these variations.

We have also conducted preliminary experiments in topic categorization. For this task, we applied the cross-entropy difference measure. Our results indicate that PPM character-context compression models perform significantly better than word-based Na"ve Bayes classifiers, and approach the performance of the linear vector support machine text classifiers. Unlike earlier work presented in (Frank at al., 2000), we have shown that the performance of various PPM n-order models are comparable (for $n \geq 2$), although overall order 3 models are best.

Feature selection in the context of compression models has shown small improvements (see the final two columns of Table 7.7). We based our feature selection on selecting features which will maximize recall/precision based on the code length difference. Our analysis of the distribution of feature code lengths over relevant (category) and non-relevant (non-category) documents indicates that our thresholding strategy was effective in removing some evidently poor features, but that many features are retained with poor discrimination values. Currently, features are ranked and selected based on the "local" code length difference for an individual symbol (and context). We conjecture that ranking features by the likely "global" effect they would have on code length difference may result in improved feature selection.

An important issue is the size of the training data. We have found that for certain applications such as language identification and authorship ascription,

a relatively small amount of training data yields excellent results (see Sections 5.1 and 5.2). However, our topic-based categorization experiments show that lack of training data can have an affect on performance and possibly this will remain an impediment to compression-based approaches in future. However, it may be possible to use a hierarchy of models, whose purpose is to progressively select features (and hence text), for subsequent compression-based categorization. For example, one could imagine a hierarchy of models for this application comprising: English model, Reuter's model (and we have evidence to suggest such a model exits), and Category models. Alternatively, one could combine genre categorization, which provides a broader classification of the data, with topic categorization in a two-tiered approach.

The idea of using text compression models is not new. Text compression models have been used for various categorization problems: language identification, dialect identification, and authorship attribution, to name but some. Recently, they have been applied to the arguably more demanding problems of subject/genre categorization, as exemplified by the Reuter's categorization task. However, unlike (Frank at al., 2000), we remain optimistic about compression-based text categorization, both because of the ease of applying the approach, and because our results indicate performance is comparable to other well-performed approaches, and capable of being still further improved. For example, some compression-based research we have conducted into combining multiple models through switching methods (Teahan and Harper, 2001) has shown significant improvements in compression over single model approaches, and these gains might transfer across into the text categorization arena. Lastly, it seems highly likely that new applications will emerge in which the combination of text compression and text categorization, when used in tandem, will be a potent mixture, especially given the huge amounts of uncategorized text which is stored on the Internet, and on Intranets.

References

Apte, C., Damerau, F. and Weiss, S. "Text mining with decision rules and decision trees,", *ACM Transactions on Information Ssystems*, 12:3, 23-251.

Bell, T. C., Cleary, J. G. and Witten, I. H. (1990). *Text compression.* New Jersey: Prentice Hall.

Benedetto, D., Caglioti, E. and Loreto, V. (2002) "Language trees and zipping," *Physical Review Letters*, 88:4, 048702-1–4.

Biber, D. (1988). *Variations across speech and writing.* Cambridge: Cambridge University Press.

Boggess, L. Argawal, R. and Davis, R. (1991). "Disambiguation of prepositional phrases in atomatically labelled technical text," *AAAI'91*, 155–159.

Brown, P. F., Della Pietra, S. A., Della Pietra, V. J., Lai, J. C. and Mercer, R. L. (1992). "An estimate of an upper bound for the entropy of English,", *Computational Linguistics*, 18:1, 31–40.

Charniak, E. (1993). *Statistical language learning.* Cambridge, Massachusetts: MIT Press.

Cleary, J. G. and Witten, I. H. (1984). "Data compression using adaptive coding and partial string matching," *IEEE Transactions on Communications*, 32:4, 396–402.

Cohen, W.J. and Singer, Y. (1996). "Context-sensitive learning methods for text categorization," *SIGIR '96: Proceedings 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 307–315.

Dumais, S., Platt, J., Heckerman, D. and Sahami, M. (1998). "Inductive learning algorithms and representations for text categorization," *Proceedings International Conference on Information and Knowledge Management*, 148–155.

Dunning, T. (1994). "Statistical identification of language," Computing Research Laboratory, New Mexico State University: Technical Report 94–273.

Francis, W.N. and Kucera, H. (1982). *Frequency analysis of English usage: lexicon and grammar.* Boston: Houghton Mifflin.

Frank, Eibe, Chui, Chang and Witten, Ian H. (2000). "Text categorization using compression models," *Proceedings IEEE Data Compression Conference*, Snowbird, Utah.

Ganeson, R. and Sherman, A.T. (1993). "Statistical techniques for language recognition: an introduction and guide for cryptanalysts," *Cryptologia*, 17:4, 321–366.

Howard, Paul G. (1993). *The design and analysis of efficient lossless data compression systems.* Ph.D. thesis. Providence, Rhode Island: Brown University.

Joachims, T. (1998). "Text categorization with support vector machines: Learning with many relevant features," *Proceedings of the 10th European Conference on Machine Learning*. Springer-Verlag.

Johansson, S. Atwell, E., Garside, R. and Leech, G. (1986). *The tagged LOB Corpus.* Bergen: Norwegian Computing Centre for the Humanities.

Juola, P. (1998). "What Can We Do With Small Corpora? Document Categorization Via Cross-Entropy," *Proceedings of the Workshop on Similarity and Categorization.*

Kessler, B., Nunberg, G. and Schutze, H. (1997) "Automatic detection of text genre," *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics.*

Khmelev, D. V. (2000). "Disputed authorship resolution through using releative empirical entropy for Markov Chains of letters in human language text," *Journal of Quantitative Linguistics*, 7:3, 201–207.

Kukushkina, O.V., Polikarpov, A.A. and Khmelev, D. V. (2001). "Using literal and grammatical statistics for authorship attribution," letters in human language text," *Problems of Information Transmission*, 37:2, 96–108.

Lee, Y. and Myaeng, S.H. (2002). "Text genre classification with genre-revealing and subject revealing features," *SIGIR '02: Proceedings 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 145–150.

Lewis, D. D. (1994). "Guest editorial," *ACM Transactions on Information Systems,* 12:3, 231.

Lewis, D. D. and Ringuette, M. (1994). "A comparison of two learning algorithms for text categorization," *Proceedings Annual Symposium on Document Analysis and Information Retrieval,* 37–50.

Manning, C. D. and Schutze, Hinrich. (1999). *Foundations of statistical natural language processing.* Cambridge, Massachusetts: MIT Press.

McCallum, A. and Nigam, K. (1998). "A comparison of event models for Naive Bayes text classification," *AAAI-98: Workshop on learning for text categorization.*

Moffat, Alistair. (1990). "Implementing the PPM data compression scheme," *IEEE Transactions on Communications,* 38:1, 1917–1921.

Mosteller, F. and Wallace, D.L. (1984). *Applied Bayesian and classical inference: the case of the Federalist papers.* New York: Springer-Verlag.

Pang, B. and Lee, L. (2002). "Thumbs up? Sentiment classification using machine learning techniques," *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing.*

Shannon, C. E. (1948). "A mathematical theory of communication," *Bell System Technical Journal,* 27, 379-423,623-656.

Rose, T.G., Stevenson, M. and Whitehead, M. (2002). "The Reuters Corpus Volume 1—from yesterday's news to tomorrow's language resources," *Proceedings of the Third International Conference on Language Resources and Evaluation* Las Palmas de Gran Canaris, 29–31.

Schutze, H., Hull, D. and Pedersen, J.O. (1995). "A comparison of classifiers and document representations for the routing problem," *SIGIR '95: Proceedings 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* 229–237.

Stamatatos, E., Fakotakis, N. and Kokkinakis, G. (2001). "Automatics text categorization in terms of genre and author," *Computational Linguistics*, 26:4, 471–495.

Teahan, W. J. (1998). *Modelling English text.* Ph.D. thesis. Hamilton, New Zealand: Waikato University.

Teahan, W. J. (2000). "Text classification and segmentation using minimum cross-entropy," *Proceedings RIAO'2000.* Paris, France. April, 2, 943–961.

Teahan, W. J. and Harper, D.J. (2001). "Combining PPM models using a text mining approach," *Proceedings DCC'2001*. Snowbird, Utah. 153–162.

Teahan, W. J., Wen, Y. Y., McNabb, R., Witten, I. H. (2000). "Using compression models to segment Chinese text," *Computational Linguistics,* 26:3, 375–393.

Witten, I. H., Bray, Z., Mahoui, M. and Teahan, W. J. (1999). "Using language models for generic entity extraction," *ICML-99 Worskhop: Machine learning in text data analysis.*

Weiner, E., Pedersen, J.O., and Weigend, A.S. (1995) "A neural network approach to topic spotting," *Proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrieval.*

Wolters, M. and Kirsten, M. (1999). "Exploring the use of linguistic features in domain and genre classification," *Proceedings of the Meeting of the European Chapter of the Association for Computational Linguistics*, Bergen, Norway.

Yang, Y. (1999) "An evaluation of statistical approaches to text categorization," *Information Retrieval*. 1:1, 69–90.

Yang, Y. and Pederson, J.O. (1997) "A comparative study on feature selection in text categorization," *Proceedings International Conference on Machine Learning*. 412–420.