Using Relative Entropy for Authorship Attribution

Ying Zhao, Justin Zobel, and Phil Vines

School of Computer Science and Information Technology, RMIT University
GPO Box 2476V, Melbourne, Australia
{yizhao, jz, phil}@cs.rmit.edu.au

Abstract. Authorship attribution is the task of deciding who wrote a particular document. Several attribution approaches have been proposed in recent research, but none of these approaches is particularly satisfactory; some of them are ad hoc and most have defects in terms of scalability, effectiveness, and efficiency. In this paper, we propose a principled approach motivated from information theory to identify authors based on elements of writing style. We make use of the Kullback-Leibler divergence, a measure of how different two distributions are, and explore several different approaches to tokenizing documents to extract style markers. We use several data collections to examine the performance of our approach. We have found that our proposed approach is as effective as the best existing attribution methods for two class attribution, and is superior for multi-class attribution. It has lower computational cost and is cheaper to train. Finally, our results suggest this approach is a promising alternative for other categorization problems.

1 Introduction

Authorship attribution (AA) is the problem of identifying who wrote a particular document. AA techniques, which are a form of document classification, rely on collections of documents of known authorship for training, and consist of three stages: preprocessing of documents, extraction of style markers, and classification based on the style markers. Applications of AA include plagiarism detection, document tracking, and forensic and literary investigations. Researchers have used attribution to analyse anonymous or disputed documents [6, 14]. The question of who wrote Shakespeare's plays is an AA problem. It could also be applied to verify the authorship of e-mails and newsgroup messages, or to identify the source of a piece of intelligence.

Broadly, there are three kinds of AA problem: binary, multi-class, and one-class attribution. In binary classification, all the documents are written by one of two authors and the task is to identify who of the two authors wrote unattributed documents. Several approaches to this problem have been described [4, 6, 10]. Multi-class classification [1,5,12], in which documents by more than two authors are provided, is empirically less effective than binary classification. In one-class

H.T. Ng et al. (Eds.): AIRS 2006, LNCS 4182, pp. 92–105, 2006.
© Springer-Verlag Berlin Heidelberg 2006

classification, also referred to as authorship verification, some documents training are written by a particular author while the authorship of the remainder is different but unknown [15]. The task is to determine whether a given document is produced by the target author. This is more difficult again, as it is easier to characterize documents as belonging to a certain class rather than to any class except the specified one.

Existing approaches use a range of methods for extracting features, most commonly style markers such as function words [1, 5, 10, 12] and grammatical elements such as part of speech [2, 21, 22]. Given these markers, AA requires use of a classification method such as support vector machines (SVMs) [5, 15] or principal component analysis [1,10,12]. However, much previous work in the area is marred by lack of use of shared benchmark data, verification on multiple data sets, or comparison between methods—each paper differs in both style markers and classification method, making it difficult to determine which element led to success of the AA method, or indeed whether AA was successful at all. It is not clear whether these methods scale, and some are ad hoc. In previous work, we compared some of these methods using common data collections [25] drawn from the TREC data [8] and other readily available sources, and found Bayesian networks and SVMs to be superior to the other approaches given function words as tokens. A secondary contribution of this new paper is extension of this previous work to grammatical style markers.

Our primary contribution is that we propose a new AA approach based on relative entropy, measured using the Kullback-Leibler divergence [17]. Language models have been successfully used in information retrieval [24], by, in effect, finding the documents whose models give the least relative entropy for the query. Here we explore whether relative entropy can provide a reliable method of categorization, where the collection of documents known to be in a category are used to derive a language model. A strong motivation for exploring such an approach is efficiency: the training process is extremely simple, consisting of identifying the distinct terms in the documents and counting their occurrences. In contrast, existing categorization methods are quadratic or exponential.

To test the proposed method, we apply it to binary and multi-class AA, using several kinds of style marker. For consistency we use the same data collections as in our previous work [25]. We observe that our method is at least as effective for binary classification as the best previous approaches, Bayesian networks and SVMs, and is more effective for multi-class classification.

In addition, we apply our method to the standard problem of categorization of documents drawn from the Reuters newswire [16]. AA is a special case of text categorization, but it does not necessarily follow that a method that is effective for AA will be effective for categorization in general, and *vice versa*. However, these preliminary experiments have found that KLD is indeed an effective general categorization technique, with effectiveness comparable to that of SVMs. We infer that, given appropriate feature extraction methods, the same categorization techniques can be used for either problem.

2 Background

The basic processes of AA consists of three stages: text preprocessing, feature extraction, and categorization. In the first stage, the text is standardized and may be annoted with lexical information. In the second stage, features are identified in the transformed text. In the third stage, feature sets are compared to determine likely authorship. A variety of AA approaches have been proposed, differing in all three stages.

In style-based classification, both lexical and grammatical markers have been used. Function words are a lexical style marker that has been widely used [1,5, 10,12], on the basis that these words carry little content: a typical author writes on many topics, but may be consistent in the use of the function words used to structure sentences. Some researchers have included punctuation symbols, while others have experimented with n-grams [13,18,19]. Grammatical style markers have also been used for AA [2,21,22], with natural language processing techniques are used to extract features from the documents. However, the AA performance is subject to the performance of the corresponding natural-language tools that are used.

Once stylistic features have been extracted, they must be used in the way to classify documents. Several researchers have applied machine-learning techniques to AA. Diederich et al. [5] and Koppel et al. [15] have used SVMs in their experiments. Diederich et al. used a collection of newspaper articles in German, with seven authors and between 82 and 118 texts for each author. Documents with fewer than 200 words were not used, as they were considered to not have enough authorial information. Accuracies of 60% to 80% were reported. The data used by Koppel et al. consists of 21 English books by a total of 10 authors. An overall accuracy of 95.7% was reported; due to the small size of data collection, the high accuracy may not be statistically significant. In our previous work [25], we used a large data collection and tested five well-known machine learning methods. We concluded that machine learning methods are promising approaches to AA. Amongst the five methods, Bayesian networks were the most effective.

Principle component analysis (PCA) is a statistical technique that several researchers have employed for AA [1,10,12]. Baayen et al. [2] used PCA on a small data collection, consisting of material from two books. Holmes et al. [10] applied PCA to identify the authorship of unknown articles that have been tentatively attributed to Stephen Crane. The data consisted of only fourteen articles known to have been written by Crane and seventeen articles of unknown authorship. PCA has largely been used for binary classification. In our initial investigation [25], PCA appears ineffective for multi-class classification. Additionally, PCA is not easily scalable; it is based on linear algebra and uses eigenvectors to determine the principle components for measuring similarity between documents. In most cases, only the first two principle components are used for classification and other components are simply discarded. Although other components may contain less information compared to the first two components, discarding will cause information loss, which may reduce the classification effectiveness.

Compression techniques and language models are another approach to AA, including Markov chains [14, 21, 22] and n-gram models [13, 18, 19]. Khmelev and Tweedie [14] used Markov chains to identify authorship for documents in Russian. Character level n-grams are used as style markers. An accuracy of 73% was reported as the best result in multi-class classification, but in most cases there were generally only two instances of each authors' work, raising doubts as to the reliability of the results. Peng et al. [19] applied character level n-gram language models to a data collection of newswire articles in Greek. The collection contains documents by 10 authors, with 20 documents for each author. Although an average of 82% accuracy was reported, the size of collection is probably too small to draw any representative conclusions. The question of whether character-level n-grams are useful as style markers is, therefore, unclear. As the full text of the documents was retained in these experiments, it is possible that the effectiveness of topic markers rather than style markers was being measured.

Another compression-based approach is to measure the change in compressed file size when an unknown document is added to a set of documents from a single author. Benedetto et al. [3] used the standard LZ77 compression program and reported an overall accuracy of 93%. In their experiment, each unknown text is appended to every other known text and the compression program is applied to each composite file as well as to the original text. The increase in size due to the unknown text can be calculated for each case, and the author of the file with smallest increase is assumed to be the target. However, Goodman [7] failed to reproduce the original results, instead achieving accuracy of only 53%.

More fundamentally, the approach is based on two poor premises. One is that the full text of the data is used, so that topic as well as style information is contributing to the outcomes; document formatting is a further confounding factor. The other premiss is that compression is an unreliable substitute for modelling. Compression techniques build a model of the data, then a coding technique uses the model to produce a compact representation. Typical coding techniques used in practice have ad hoc compromises and heuristics to allow coding to proceed at a reasonable speed, and thus may not provide a good indication of properties of the underlying model. By using off-the-shelf compression rather than examining properties of the underlying model, much accuracy may be lost, and nothing is learnt about which aspects of the modelling are successful in AA. In the next section we explore how models can be directly applied to AA in a principled manner.

For classification tasks in general, two of the most effective methods are SVMs and Bayesian networks. SVMs [20] have been successfully used in applications such as categorization and handwriting recognition. The basic principle is to find values for parameters α_i for data points that maximize

$$\sum_i \alpha_i - \frac{1}{2} \sum \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)$$

These values define a hyperplane, where the dimensions correspond to features. Whether an item is in or out of a class depends on which side of the hyperplane it lies. However, the computational complexity of SVM is a drawback. Even the best algorithm gives $O(n^2)$ computational cost, for n training samples.

A Bayesian network structure [9] is an acyclic directed graph in which there is one node in the graph for each feature and each node has a table of transition probabilities for estimating probabilistic relationships between nodes based on conditional probabilities. There are two learning steps in Bayesian networks, learning of the network structure and learning of the conditional probability tables. The structure is determined by identifying which attributes have the strongest dependencies between them. The nodes, links, and probability distributions are the structure of the network, which describe the conditional dependencies. However a major drawback of this approach is that asymptotic cost is exponential, prohibiting use of Bayesian networks in many applications.

3 Entropy and Divergence

Entropy measures the average uncertainty of a random variable X. In the case of English texts, each $x \in X$ could be a token such as a character or word. The entropy is given by:

$$H\left(X\right) = -\sum_{x \in X} p\left(x\right) \log_2 p\left(x\right)$$

where $p\left(x\right)$ is the probability mass function of a random character or word. $H(X)$ represents the average number of bits required to represent each symbol in X. The better the model, the smaller the number of bits.

For example, we could build a model for a collection of documents by identifying the set W of distinct words w, the frequency f_w with which each w occurs, and the total number $n = \sum_w f_w$ of word occurrences. This model is context free, as no use is made of word order. The probability $p(w) = f_w/n$ is the maximum likelihood for w, and

$$n \times \left(-\sum_w p(w) \log_2 p\left(w\right) \right) = -\sum_w f_w \log_2 \frac{f_w}{n}$$

is the minimum number of bits required to represent the collection under this model. The compression-based AA techniques considered above can be regarded as attempting to identify the collections whose models yield the lowest entropy for a new document, where however the precise modelling technique is unknown and the model is arbitrarily altered to achieve faster processing.

A difficulty in using direct entropy measurements on new documents is that the document may contain a new word w that is absent from the original model, leading to $p(w) = 0$ and undefined $\log_2 p(w)$. We examine this issue below.

Another way to use entropy is to compare two models, that is, to measure the difference between two random variables. A mechanism for this measurement of relative entropy is the *Kullback-Leibler divergence* (KLD) [17], given by:

$$KLD\left(p||q\right) = \Sigma_{x \in X} \; p\left(x\right) \log_2 \frac{p\left(x\right)}{q\left(x\right)}$$

where $p\left(x\right)$ and $q\left(x\right)$ are two probability mass functions.

In this paper, we propose the use of KLD as a categorization technique. If a document with probability mass function p is closer to q than to q'—that is, has a smaller relative entropy—then, we hypothesise, the document belongs in the category corresponding to q. The method is presented in detail later.

We use simple language modelling techniques to estimate the probability mass function for each document and category. Language models provide a principle for quantifying properties of natural language. In the context of using language models for AA, we assume that the act of writing is a process of generating natural language. The author can be considered as having a model generating documents of a certain style. Therefore, the problem is to quantify how different the authors' models are.

Given a token sequence $c_1 c_2 \ldots c_n$ representing a document we need to estimate a language model for the document. In an ideal model, we would have enough data to use context to estimate a high $p(c_i|c_1 \ldots c_{i-1})$ should be obtained for each token occurrence. However, in common with most use of language models in information retrieval, we use a unigram model; for example, if the tokens are words, there are simply not enough word sequences to estimate multigram probabilities, and thus we wish only to estimate each $p(c_i)$ independently.

Therefore, the task is to find out a probability function to measure the probability of each component that occurs in the document. The most straightforward estimation in language modelling is the maximum likelihood estimate, in which the probability of each component is given by the frequency normalized by the total number of components in that document d (or, equivalently, category C):

$$p_d\left(c\right) = \frac{f_{c,d}}{|d|}$$

where $f_{c,d}$ is the frequency of c in d and $|d| = \sum_{c' \in d} f_{c',d}$. We then can determine the KLD between a document d and category C as

$$\begin{aligned} KLD\left(p_d||p_C\right) &= \sum_{c \in C \cup d} p_d\left(c\right) \log_2 \frac{p_d\left(c\right)}{p_C\left(c\right)} \\ &= \sum_{c \in C} \frac{f_{c,d}}{|d|} \log_2 \frac{f_{c,d} \cdot |C|}{f_{c,d} \cdot |d|} \end{aligned} \qquad (1)$$

KLD as a Classifier for Authorship Attribution

Given author candidates $A = \{a_1 \ldots a_j\}$, it is straightforward to build a model for each author by aggregating the training documents. We can build a model for an unattributed document in the same way. We can then determine the author model that is most similar to the model of the unknown document, by calculating KLD values between author models and unknown documents to identify the target author for which the KLD value is the smallest.

However, it is usually the case that some components are missing in either the training documents or the documents to be attributed. This generates an undefined value in equation 1, and thus a KLD value cannot be computed. This is a standard problem with such models, and other researchers have explored a variety of smoothing techniques [24] to calculate the probability of missing components.

The Dirichlet prior is an effective smoothing technique for text-based applications, in particular information retrieval. We use Dirichlet smoothing to remove these zero probabilities, under which the probability of component c in document d (or equivalently, category C) is:

$$\begin{aligned} p'_d(c) &= \frac{|d|}{\lambda + |d|}\frac{f_{c,d}}{|d|} + \frac{\lambda}{\lambda + |d|} p_B\left(c\right) \\ &= \frac{f_{c,d}}{\lambda + |d|} + \frac{\lambda}{\lambda + |d|} p_B\left(c\right) \end{aligned}$$

where λ is a smoothing parameter and $p_B\left(c\right)$ is the probability of component c in a *background model*. For short documents, the background probabilities dominate, on the principle that the evidence for the in-document probabilities is weak. As document length grows, the influence of the background model diminishes. Choice of an appropriate value for λ is a tuning stage in the use of language models.

In principle the background model could be any source of typical statistics for components. Intuitively it makes sense to derive the model from other documents of similar type; in attributing newswire articles, for example, a background model derived from poetry seems unlikely to be appropriate. As background model, we use the aggregate of all known documents, including training and test, as this gives the largest available sample of material. There is no reason why a background model could not be formed this way in practice.

In estimating KLD, the same background model is used for documents and categories, so KLD is computed as

$$\begin{aligned} &KLD\left(p_d || p_C\right) \;= \\ &\sum_{c \in C \cup d} \left[\left(\frac{f_{c,d}}{\lambda + |d|} + \frac{\lambda}{\lambda + |d|} p_B(c) \right) \times \log_2 \frac{\frac{f_{c,d}}{\lambda+|d|} + \frac{\lambda}{\lambda+|d|} p_B(c)}{\frac{f_{c,C}}{\lambda+|C|} + \frac{\lambda}{\lambda+|C|} p_B(c)} \right] \end{aligned} \tag{2}$$

By construction of the background model, $d \subset C$, so there are no zeroes in the computation.

4 Feature Types

Function words are an obvious choice of feature for authorship attribution, as they are independent of the content but do represent style. A related choice of feature is punctuation, though the limited number of punctuation symbols mean that their discrimination power must be low.

An alternative is to use lexical elements. We explored the use of parts of speech, that is, lexical categories. Linguists recognize four major categories of words in English: nouns, verbs, adjectives, and adverbs. Each of these types can be further classified according to morphology. Most part-of-speech tag sets make use of the same basic categories; however, tag sets differ in how finely words are divided into categories, and in how categories are defined.

In this paper, we propose the following approach to use of parts of speech in authorship attribution. We applied NLTK (a Natural Language ToolKit)[1] to extract the part-of-speech tags from each original document. The part-of-speech tag set we used to tag our data collection in text preprocessing is the "brown" tag set. For simplicity, and to ensure that our feature space was not too sparse, we condensed the number of distinct tags from 116 to 27, giving basic word classes whose statistical properties could be analysed.

A further refinement is to combine the classes. We explore combinations of function words, parts of speech, and punctuation as features in our experiments.

5 Experiments

We used experiments on a range of data sources to examine effectiveness and scalability of KLD for attribution. In preliminary experiments, we also examined the effectiveness of KLD for other types of classification problems. Several data collections were used in our experiments, including newswire articles from the Associated Press (AP) collection [8], English literature from the Gutenberg Project, and the Reuters-21578 test collection [16]. The first two data collections are used for AA. The Reuters-21578 test collection was used to examine the applicability of KLD for general categorization.

AP. From the AP newswire collection we have selected seven authors who each contributed over 800 documents. The average document length is 724 words. These documents are splitted into training and testing groups. The number of documents used for training was varied to examine the scalability of the methods. This collection was used in our previous work [25].

Gutenberg project. We wanted to test our technique on literary works, and thus selected the works of five well known authors from the Gutenberg project[2]: *Haggard*, *Hardy*, *Tolstoy*, *Trollope*, and *Twain*. Each book is divided into chapters and splitted for training and testing. Our collection consists of 137 books containing 4335 chapters. The number of chapters from each author ranges from 492 to 1174, and the average chapter length is 3177 words. In our experiments, the number of chapters used for training is randomly selected and varied.

Reuters-21578. These documents are from the Reuters newswire in 1987, and have been used as a benchmark for general text categorization tasks. There are 21578 documents. We use the Modapte split [16] to group documents for training and testing. The top eight categories are selected as the target classes; these are *acq*, *crude*, *earn*, *grain*, *interest*, *money-fx*, *ship*, and *trade*.

[1] Available from `http://nltk.sourceforge.net/index.html`.

[2] `www.gutenberg.org`

Table 1. Effectiveness (percentage of test documents correctly attributed) for Bayesian networks, SVMs, and KLD attribution on two-class classification. The data is the AP collection, with function words as features. Best results in each case are shown in **bold**.

Docs per	Bayes	KLD	KLD	KLD	KLD	SVM
author	network	$\lambda = 10$	$\lambda = 10^2$	$\lambda = 10^3$	$\lambda = 10^4$	
50	78.90	89.24	**89.98**	89.67	77.83	85.81
100	81.55	90.93	**91.19**	91.17	82.10	89.38
200	84.18	91.74	**91.81**	91.67	87.38	91.12
400	84.82	92.05	92.19	92.19	89.86	**92.40**
600	84.46	92.17	92.14	92.24	90.74	**92.86**

We used the KLD method in a variety of ways to examine robustness and scalability of classification. We first conducted experiments for *two-class classification*, that is, to discriminate between two known authors. In this context, all the documents used for training and testing are written by either one of these two candidates. *Multi-class classification*, also called n-class classification for any $n \geq 2$, is the extension of two-class classification to arbitrary numbers of authors.

We applied KLD classification to all three data collections for both binary classification and n-class classification. In all experiments, we compared our proposed KLD language model method to Bayesian networks, which was the most effective and scalable classification method in our previous work [25]. In addition, we have made the first comparison between a KLD classifier with SVM, a successful machine learning method for classification. We used leave-one-out validation method to avoid the overfitting problem and estimate the true error rate for classification. The linear kernel was selected as most text categorization problems are linear separable [11]. More complex kernel functions have not been shown to significantly increase the classification rate [20, 23]. The package used in our experiments is *SVM-light.*[3]

We also investigated the significance of different types of features that can be used to mark authorial structure of a particular document. As discussed above, we have used function words, parts of speech, and punctuation as features; these were used both separately and in combination.

Two-class experiments. Our experiments were for the two-class classification task. The results were reported in Table 1, where outcomes are averaged across all 21 pairs of authors, because significant inconsistencies were observed from one pair of authors to another in our previous reported experiments [25]. We tested different values of λ: 10, 10^2, 10^3, and 10^4.

We observed that the best results were obtained for value of $\lambda = 10^2$ and $\lambda = 10^3$. To examine the scalability of KLD attribution, we have increased the number of documents used for training and maintained the same set of test documents. As can be seen, the accuracy of classification increases as the number of documents for training is increased, but appears to plateau. The KLD

[3] Available from `http://svmlight.joachims.org`.

Table 2. Effectiveness (percentage of test documents correctly attributed) of KLD attribution with $\lambda = 10^2$ on AP, using different feature types, for two-class classification

Docs per author	func word	POS tags	POS(punc)	combined
50	**89.98**	83.00	83.38	88.38
100	**91.19**	82.90	83.21	88.79
200	**91.67**	82.90	83.79	89.62
400	**92.19**	83.29	83.67	89.36
600	**92.14**	83.07	83.52	89.17

Table 3. Effectiveness (percentage of test documents correctly attributed) for Bayesian networks, SVMs, and KLD attribution on two-class classification. The data is the Gutenberg collection, with function words as features.

Docs per	Bayes	KLD	KLD	KLD	SVM
author	network	$\lambda = 10^2$	$\lambda = 10^3$	$\lambda = 10^4$	
50	93.50	94.70	**94.80**	84.30	91.40
100	95.10	95.80	**96.00**	88.90	94.85
200	95.10	96.10	**96.50**	93.70	**96.50**
300	95.38	96.50	96.70	95.50	**97.20**

method is markedly more effective than the Bayesian network classifier. With a small number of documents for modelling, the KLD method is more effective than SVM, while with a larger number of documents SVM is slightly superior.

As noted earlier, the computational cost of the SVM and Bayesian network methods is quadratic or exponential, whereas the KLD method is approximately linear in the number of distinct features. It is thus expected to be much more efficient; however, the diversity of the implementations we used made it difficult to meaningfully compare efficiency.

We next examined discrimination power of different feature types, using KLD classification on the two class classification task. As discussed above, we used function words, part-of-speech (POS) tags, POS with punctuation, and a combined feature set containing all previous three types of feature. Results are reported in Table 2, which shows the average effectiveness from the 21 pairs of authors. Function words were best in all cases, and so we concentrated on these in subsequent experiments. With all feature types, effectiveness improved with volume of training data, but only up to a point.

We then tested KLD attribution on the Gutenberg data we had gathered. Average effectiveness is reported in Table 3. The trends were similar to those observed on the AP collection. Again, our proposed KLD method is consistently more effective than Bayesian networks, and SVM is more effective than KLD only when a larger number of training documents is used; when SVM is superior, the difference is slight. In combination these results show that KLD attribution can be successfully used for binary attribution.

We applied the KLD approach to the Gutenberg data to examine the discrimination power of different feature types. Results are shown in Table 4. In

Table 4. Effectiveness (percentage of test documents correctly attributed) of KLD method for Gutenberg attribution, using different feature types, on two-class classification

Docs per author	functions word	POS tags	POS(punc)	combined
50	94.80	86.00	91.10	**96.10**
100	**96.00**	86.35	93.05	95.70
200	**96.50**	85.65	93.40	96.30
300	**96.70**	86.15	93.10	96.34

Table 5. Effectiveness (percentage of test documents correctly attributed) of Bayesian networks and KLD attribution for the AP data, on two- to five-class classification

Number of authors	Bayes network	KLD $\lambda = 10^2$	KLD $\lambda = 10^3$	KLD $\lambda = 10^4$
50 documents per author				
2	89.67	**92.14**	91.41	74.86
3	79.49	**84.21**	83.97	64.55
4	75.83	**81.43**	81.14	52.77
5	71.72	76.15	**76.27**	48.36
300 documents per author				
2	90.46	**94.95**	94.91	91.82
3	85.22	**88.70**	88.61	85.24
4	80.63	87.00	**87.05**	82.05
5	76.33	82.84	**83.11**	77.15

one case, the combined feature set is superior; in the remainder, the best feature type is again the function words.

Multi-class experiments. We next examined the performance of the KLD method when applied to multi-class classification. In the two-class experiments, the function words were the best at discrimination amongst different author styles; in the following experiments, then, we compared Bayesian networks and the KLD classification method using only function words as the feature set. SVMs were not used, as they cannot be directly applied to multi-class classification.

For each test, we used 50 and 300 documents from each author for training. The outcomes were again averaged from all possible author combinations, that is 21 combinations for 2 and 5 authors, and 35 combinations for 3 and 4 authors. As shown in Table 5, with appropriate λ values, the KLD approach consistently and substantially outperforms Bayesian networks. Smaller values of λ are the more effective, demonstrating that the influence of the background model should be kept low.

We then ran the corresponding experiments on the Gutenberg data, as shown in Table 6. The outcomes were the same as that on the AP data, illustrating that the method and parameter settings appear to be consistent between collections.

General text categorization. In order to determine the suitability of KLD classification for other types of classification tasks, we used the Reuters-21578

Table 6. Effectiveness (percentage of test documents correctly attributed) of Bayesian networks and KLD attribution for the Gutenberg data, on two- to five-class classification

Number of authors	Bayes network	KLD $\lambda = 10^2$	KLD $\lambda = 10^3$	KLD $\lambda = 10^4$
50 documents per author				
2	93.50	94.70	**94.80**	84.30
3	88.80	**92.33**	91.87	71.97
4	87.67	**89.80**	89.15	62.75
5	86.00	**87.60**	87.00	54.80
300 documents per author				
2	95.38	96.50	**96.70**	95.50
3	91.13	94.73	**94.90**	92.30
4	88.75	92.80	**93.00**	90.05
5	87.25	91.00	**91.20**	88.20

Table 7. Effectiveness (precision, recall, and accuracy) of KLD classification and SVM for general text categorization on the Reuters-21578 test collection

categories top 8 (1 vs. n)	relevant/irrelevant (same train/test split)	KLD($\lambda = 10^2$) rec/pre/acc	SVM rec/pre/acc
acq	668/1675	**95.81**/93.70/96.97	94.01/96.32/97.27
crude	150/2193	**96.58**/62.95/96.24	69.33/91.23/97.61
earn	1048/1295	97.23/90.02/93.94	**98.19**/98.19/98.38
grain	117/2226	**99.15**/71.17/97.95	84.62/99.00/99.19
interest	80/2263	**92.50**/45.68/95.99	37.50/93.75/97.78
money-fx	123/2224	**95.12**/54.42/95.56	69.11/80.95/97.52
ship	54/2289	**85.19**/33.58/95.78	24.07/86.67/98.16
trade	103/2240	**93.20**/52.17/95.95	67.98/87.50/98.16

collection to test topic-based classification using KLD. In the Reuters-21578 data collection, documents are often assigned to more than one category. (This is a contrast to AA, in which each document has only one class.) In our experiment, we chose the first category as the labelled class, as it is the main category for that document. In common with standard topic classification approaches we used all document terms as the classification features.

In these preliminary experiments—we do not claim to have thoroughly explored the application of KLD to general categorization—we tested n-class classification, where $n = 8$, both with and without stemming. We compared KLD classification and SVM in terms of precision, recall, and overall accuracy. Accuracy measures the number of documents correctly classified. Thus for any given category, it is calculated as the total number of documents correctly classified as belonging to that category, plus the total number of documents correctly classified as not belonging to that category, divided by the total number of documents classified. Results are shown in Table 7. KLD classification consistently achieves higher recall than SVMs, but with worse precision and slightly lower accuracy. We conclude that KLD classification is a plausible method for general text categorization, but that further exploration is required to establish how best it should be used for this problem.

6 Conclusions

We have proposed the use of relative entropy as a method for identifying authorship of unattributed documents. Simple language models have formed the basis of a recent series of developments in information retrieval, and have the advantage of simplicity and efficiency. Following simple information theoretic principles, we have shown that a basic measure of relative entropy, the Kullback-Leibler divergence, is an effective attribution method.

Here and in other work we have explored alternative attribution methods based on machine learning methods. These methods are computationally expensive and, despite their sophistication, at their best can only equal relative entropy. We have also explored other feature extraction methods, but the results show that function words provide a better style marker than do tokens based on parts of speech or patterns of punctuation. Compared to these previous methods, we conclude that relative entropy, based on function word distributions, is efficient and effective for two-class and multi-class authorship attribution.

Acknowledgements. This work was supported by the Australian Research Council.

References

1. H. Baayen, H. V. Halteren, A. Neijt, and F. Tweedie. An experiment in authorship attribution. *6th JADT*, 2002.
2. H. Baayen, H. V. Halteren, and F. Tweedie. Outside the cave of shadows: Using syntactic annotation to enhance authorship attribution. *Literary and Linguistic Computing*, 11(3):121–132, 1996.
3. D. Benedetto, E. Caglioti, and V. Loreto. Language trees and zipping. *The American Physical Society*, 88(4), 2002.
4. J. N. G. Binongo. Who wrote the 15th book of oz? an application of multivariate statistics to authorship attribution. *Computational Linguistics*, 16(2):9–17, 2003.
5. J. Diederich, J. Kindermann, E. Leopold, and G. Paass. Authorship attribution with support vector machines. *Applied Intelligence*, 19(1-2):109–123, 2003.
6. G. Fung. The disputed federalist papers: Svm feature selection via concave minimization. In *Proceedings of the 2003 Conference on Diversity in Computing*, pages 42–46. ACM Press, 2003.
7. J. Goodman. Extended comment on language trees and zipping, 1995.
8. D. Harman. Overview of the second text retrieval conference (TREC-2). *Information Processing & Management*, 31(3):271–289, 1995.
9. D. Heckerman, D. Geiger, and D. Chickering. Learning bayesian networks: the combination of knowledge and statistical data. *Machine Learning*, 20:197–243, 1995.

10. D. I. Holmes, M. Robertson, and R. paez. Stephen crane and the new-york tribune: A case study in traditional and non-traditional authorship attribution. *Computers and the Humanities*, 35(3):315–331, 2001.
11. T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In C. Nédellec and C. Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, number 1398, pages 137–142, Chemnitz, DE, 1998. Springer Verlag, Heidelberg, DE.
12. P. Juola and H. Baayen. A controlled-corpus experiment in authorship identification by cross-entropy. *Literary and Linguistic Computing*, 2003.
13. V. Keselj, F. Peng, N. Cercone, and C. Thomas. N-gram-based author profiles for authorship attribution. In *Pasific Association for Computational Linguistics*, pages 256–264, 2003.
14. D. V. Khmelev and F. J. Tweedie. Using markov chains for identification of writers. *Literary and Linguistic Computing*, 16(4):229–307, 2002.
15. M. Koppel and J. Schler. Authorship verification as a one-class classification problem. In *Twenty-first International Conference on Machine Learning.* ACM Press, 2004.
16. D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. Rcv1: A new benchmark collection for text categorization research. *J. Mach. Learn. Res.*, 5:361–397, 2004.
17. Manning and H. Schze. *Foundations of Statistical Natural Language Processing.* MIT Press, Cambridge, may 1999.
18. F. Peng, D. Schuurmans, V. Keselj, and S. Wang. Language independent authorship attribution using character level language models. In *10th Conference of the European Chapter of the Association for Computational Linguistics, EACL*, 2003.
19. F. Peng, D. Schuurmans, and S. Wang. Language and task independent text categorization with simple language models. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 110–117, Morristown, NJ, USA, 2003. Association for Computational Linguistics.
20. B. Scholkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond.* The MIT Press, 2002.
21. E. Stamatatos, N. Fakotakis, and G. Kokkinakis. Automatic authorship attribution. In *Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics*, pages 158–164, 1999.
22. E. Stamatatos, N. Fakotakis, and G. Kokkinakis. Computer-based authorship attribution without lexical measures. *Computers and the Humanities*, 35(2):193–214, 2001.
23. V. Vapnik and D. Wu. Support vector machine for text categorization, 1998.
24. C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, 22(2):179–214, 2004.
25. Y. Zhao and J. Zobel. Effective authorship attribution using function word. In *2nd Asian Information Retrieval Symposium*, pages 174–190. Springer, 2005.