Language Independent Authorship Attribution using Character Level Language Models

Fuchun Peng[†] **Dale Schuurmans**[†] **Vlado Keselj**[‡] **Shaojun Wang**[†]
[†]School of Computer Science, University of Waterloo, Canada
{f3peng, dale, sjwang}@cs.uwaterloo.ca
[‡]Faculty of Computing Science, Dalhousie University, Canada
vlado@cs.dal.ca

Abstract

We present a method for computer-assisted authorship attribution based on character-level n-gram language models. Our approach is based on simple information theoretic principles, and achieves improved performance across a variety of languages without requiring extensive pre-processing or feature selection. To demonstrate the effectiveness and language independence of our approach, we present experimental results on Greek, English, and Chinese data. We show that our approach achieves state of the art performance in each of these cases. In particular, we obtain a 18% accuracy improvement over the best published results for a Greek data set, while using a far simpler technique than previous investigations.

1 Introduction

Automated authorship attribution is the problem of identifying the author of an anonymous text, or text whose authorship is in doubt (Love, 2002). A famous example is the *Federalist Papers*, of which twelve are claimed to have been written both by Alexander Hamilton and James Madison (Holmes and Forsyth, 1995). Recently, vast repositories of electronic text have become available on the Internet, making the problem of managing large text collections increasingly important. Automated text categorization (TC) is a useful way to organize a large document collection by imposing a desired categorization scheme. For example, categorizing documents by their author is an important case that has become increasingly useful, but also increasingly difficult in the age of web-documents that can be easily copied, translated and edited. Author attribution is becoming an important application in web information management, and is beginning to play a role in areas such as information retrieval, information extraction and question answering.

Many algorithms have been invented for assessing the authorship of given text. These algorithms rely on the fact that authors use linguistic devices at every level—semantic, syntactic, lexicographic, orthographic and morphological (Ephratt, 1997)—to produce their text. Typically, such devices are applied unconsciously by the author, and thus provide a useful basis for unambiguously determining authorship. The most common approach to determining authorship is to use a stylistic analysis that proceeds in two steps: first, specific *style markers* are extracted, and second, a *classification procedure* is applied to the resulting description. These methods are usually based on calculating lexical measures that represent the richness of the author's vocabulary and the frequency of common word use (Stamatatos et al., 2001). Style marker extraction is usually accomplished by some form of non-trivial NLP analysis, such as tagging, parsing and morphological analysis. A classifier is then constructed, usually by first performing a non-trivial feature selection step that employs mutual information or Chi-square testing to determine relevant features.

There are several problems with this standard approach however. First, techniques used for style marker extraction are almost always language dependent, and in fact differ dramatically from language to language. For example, an English parser usually cannot be applied to German or Chinese. Second, feature selection is not a trivial process, and usually involves setting thresholds to eliminate uninformative features (Scott and Matwin, 1999). These decisions can be extremely subtle, because although rare features contribute less signal than common features, they can still have an important cumulative effect (Aizawa, 2001). Third, current authorship attribution systems invariably perform their analysis at the *word level*. However, although word level analysis seems to be intuitive, it ignores the fact that morphological features can also play an important role, and moreover that many Asian languages such as Chinese and Japanese do not have word boundaries explicitly identified in text. In fact, word segmentation itself is a difficult problem in Asian languages, which creates an extra level of difficulty in coping with the errors this process introduces.

In this paper, we propose a simple method that avoids each of these problems. Our approach is based on building a character-level n-gram model of an author's writing, using techniques from statistical language modeling. Language modeling is concerned with capturing regularities of natural language—for example, semantic, syntactic, lexicographic and morphological patterns—that can be used to make predictions. Many of the features considered in language modeling coincide with those used in authorship attribution, and it is therefore natural to apply language modeling concepts to this problem. To perform authorship attribution, we build a character-level n-gram language model for each author. This approach exploits morphological features while avoiding the need for explicitly segmented words. By considering all possible character n-grams as potential features, we also avoid the need to run sophisticated NLP tools, such as parsers and taggers, to produce candidate features. Finally, we avoid feature selection entirely by including every feature in the model, but use estimation methods from statistical language modeling to avoid over-fitting a sparse set of training data. The result is a surprisingly simple, effective approach to authorship attribution that is completely language independent.

2 n-Gram Language Modeling

The dominant motivation for language modeling has traditionally come from speech recognition, but language models have recently become widely used in many other application areas, such as information retrieval, machine translation, optical character recognition, spelling correction, document classification, and bio-informatics.

The goal of language modeling is to predict the probability of naturally occurring word sequences, $s = w_1 w_2 ... w_N$; or more simply, to put high probability on word sequences that actually occur (and low probability on word sequences that never occur). Given a word sequence $w_1 w_2 ... w_N$ to be used as a test corpus, the quality of a language model can be measured by the empirical perplexity and entropy scores on this corpus

$$\begin{aligned} \textit{Perplexity} &= \sqrt[N]{\prod_{i=1}^{N} \frac{1}{\Pr(w_i|w_1...w_{i-1})}} \\ \textit{Entropy} &= \log_2 \textit{Perplexity} \end{aligned}$$

where the goal is to minimize these measures.

The simplest and most successful approach to language modeling is still based on the n-gram model. By the chain rule of probability we can write the probability of any word sequence as

$$\Pr(w_1 w_2 ... w_N) = \prod_{i=1}^{N} \Pr(w_i|w_1...w_{i-1}) \quad (1)$$

An n-gram model approximates this probability by assuming that the only words relevant to predicting $\Pr(w_i|w_1...w_{i-1})$ are the previous $n-1$ words; that is, it assumes

$$\Pr(w_i|w_1...w_{i-1}) = \Pr(w_i|w_{i-n+1}...w_{i-1})$$

A straightforward maximum likelihood estimate of n-gram probabilities from a corpus is given by the observed frequency of each of the patterns

$$\Pr(w_i|w_{i-n+1}...w_{i-1}) = \frac{\#(w_{i-n+1}...w_i)}{\#(w_{i-n+1}...w_{i-1})} \quad (2)$$

where #(.) denotes the number of occurrences of a specified gram in the training corpus. Although one could attempt to use simple n-gram models to capture long range dependencies in language, attempting to do so directly immediately creates sparse data problems: Using grams of length up to n entails estimating the probability of W^n events, where W is the size of the word vocabulary. This quickly overwhelms modern computational and data resources for even modest choices of n (beyond 3 to 6). Also, because of the heavy tailed nature of language (i.e. Zipf's law) one is likely to encounter novel n-grams that were never witnessed during training in any test corpus, and therefore some mechanism for assigning non-zero probability to novel n-grams is a central and unavoidable issue in statistical language modeling. One standard approach to smoothing probability estimates to cope with sparse data problems (and to cope with potentially missing n-grams) is to use some sort of back-off estimator.

$$\Pr(w_i|w_{i-n+1}...w_{i-1})$$
$$= \begin{cases} \hat{\Pr}(w_i|w_{i-n+1}...w_{i-1}), \\ \qquad \text{if } \#(w_{i-n+1}...w_i) > 0 \\ \beta(w_{i-n+1}...w_{i-1}) \times \Pr(w_i|w_{i-n+2}...w_{i-1}), \\ \qquad \text{otherwise} \end{cases} \tag{3}$$

where

$$\hat{\Pr}(w_i|w_{i-n+1}...w_{i-1}) = \frac{\textit{discount}\,\#(w_{i-n+1}...w_i)}{\#(w_{i-n+1}...w_{i-1})} \tag{4}$$

is the discounted probability and $\beta(w_{i-n+1}...w_{i-1})$ is a normalization constant

$$\frac{1 - \sum_{x\in(w_{i-n+1}...w_{i-1}x)} \hat{P}(x|w_{i-n+1}...w_{i-1})}{1 - \sum_{x\in(w_{i-n+1}...w_{i-1}x)} \hat{P}(x|w_{i-n+2}...w_{i-1})} \tag{5}$$

The discounted probability (4) can be computed with different smoothing techniques, including linear smoothing, absolute smoothing, Good-Turing smoothing and Witten-Bell smoothing. Although this choice has been found to make a difference in practice, Good-Turing smoothing is normally a reasonable choice (Chen and Goodman, 1998). The details of the smoothing techniques are omitted here for simplicity.

The language models described above use individual words as the basic unit, although one can easily consider models that use individual *characters* as the basic unit instead. The rest of the details remain the same in this case. The only difference is that the character vocabulary is always much smaller than the word vocabulary, which means that one can normally use a much larger context n in a character-level model (although the text spanned by a character model is still usually less than that spanned by a word model). The benefits of the character-level model in the context of author attribution are that it avoids the need for explicit word segmentation in the case of Asian languages, it captures important morphological properties of an author's writing, it can still discover useful inter-word and inter-phrase features, and it greatly reduces the sparse data problems associated with large vocabulary models. We experiment with character-level models below.

3 Language Models for Author Attribution

Our approach to applying language models to author attribution is to use Bayesian decision theory. Assume we wish to classify a text D into an author category $c \in C = \{c_1, ..., c_{|C|}\}$. A natural choice is to pick the category c that has the largest posterior probability given the text. That is,

$$c^* = \arg\max_{c\in C}\{\Pr(c|D)\} \tag{6}$$

Using Bayes rule, this can be rewritten as

$$c^* = \arg\max_{c\in C}\{\Pr(D|c)\Pr(c)\} \tag{7}$$
$$= \arg\max_{c\in C}\{\Pr(D|c)\} \tag{8}$$

where deducing Eq. (8) from Eq. (7) assumes uniformly weighted categories (since we have no other prior knowledge in this case). Here, $\Pr(D|c)$ is the likelihood of D under category c, which can be computed by Eq. (1). Therefore, our approach is to learn a separate language model for each author, by training on a data set from that author. Then, to categorize a new text D, we supply D to each language model, evaluate the likelihood of D under the model, and pick the winning author category according to Eq. (8).

4 Experimental Results

In this section, we present experimental results for our approach on three different languages. We first describe the performance measures used in our experiments, and then present results on Greek data, English data and Chinese data, in Sections 4.2, 4.3 and 4.4 respectively.

4.1 Performance Measures

We assume each text document we are classifying is written by a single author. (Although it is interesting to consider the problem of classifying multiply authored texts, we do not discuss this case in this paper.) In the case where each text only belongs to one author, the performance of an authorship classifier can be naturally measured by its *overall accuracy*: the number of correctly classified texts divided by the number of texts classified overall. However, overall accuracy does not characterize how the classifier performs on each individual category. Therefore, to measure classifier performance for each category we use the *precision*, *recall*, and *macro-average F-measure* scores. For a given category c, *precision* is the number of correctly classified texts in c, divided by the number of all texts classified to be in c. *Recall* is the number of correctly classified texts in c, divided by the number of all texts that truly belong to c. F-measure is a combination of precision and recall defined as $\frac{2 \times precision \times recall}{precision + recall}$. A macro-average F-measure is computed by averaging over all individual F-measures for each category. We now proceed to present our results for the three different languages.

4.2 Greek data

We have experimented with two Greek data sets, A and B, used in the studies of (Stamatatos et al., 1999) and (Stamatatos et al., 2000). Both sets were originally downloaded from the website of the Modern Greek Weekly Newspaper TO BHMA. Each of the two data sets consists of 200 singly-authored documents written by 10 different authors, with 20 different documents written by each author. In our experiments we used 10 of each authors' documents as training data and 10 as test instances. The specific authors that appear are shown in Table 1.

Data set		Author Name	Train size (characters)
A	A0	G. Bitros	47868
	A1	K. Chalbatzakis	71889
	A2	G. Lakopoulos	77549
	A3	T. Lianos	45766
	A4	N. Marakis	59785
	A5	D. Mitropoulos	70210
	A6	D. Nikolakopoulos	75316
	A7	N. Nikolaou	51025
	A8	D. Psychogios	35886
	A9	R. Someritis	50816
B	B0	S. Alaxiotis	77295
	B1	G. Babiniotis	75965
	B2	G. Dertilis	66810
	B3	C. Kiosse	102204
	B4	A. Liakos	89519
	B5	D. Maronitis	36665
	B6	M. Ploritis	72469
	B7	T. Tasios	80267
	B8	K. Tsoukalas	104065
	B9	G. Vokos	64479

Table 1: Authors in two Greek data sets, A and B.

The main difference between the two sets is that the documents in group A are written by journalists on a variety of topics, including news reports, editorials, etc., whereas the documents in group B are written by scholars on topics in science, history, culture, etc. The result is that the documents in group A are more heterogeneous in their style, whereas the documents in group B are more homogeneous owing to the more rigid strictures of academic writing (Stamatatos et al., 2000).

In our experiments, we obtained the best performance on both group A and group B by using a 3-gram model with absolute smoothing. The best accuracy we obtained on test documents from group A is **74%**, and **90%** on group B. This compares favorably to the best accuracy reported in (Stamatatos et al., 2000) of 72% on group A and 70% on group B.[1] Thus, our accuracy improvement is 2% on group A and 18% on group B, which is surprising given the relative simplicity of our method.

[1]Note that Stamatatos et al.'s measures of *identification error* and *average error* correspond to our *recall* and *overall accuracy* measures respectively.

Results on group A											
	Computer Estimate										
True Label	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	Recall
A0	10										1.00
A1		6	2					2			0.60
A2			9			1					0.90
A3	2		2	5						1	0.50
A4					9	1					0.90
A5			1			9					0.90
A6			3				7				0.70
A7			3				1	6			0.60
A8				2				1	3	4	0.30
A9										10	1.00
Precision	0.83	1.00	0.45	0.71	1.00	0.82	0.88	0.67	1.00	0.67	
Overall Accuracy: **0.74**						Macro-average F-measure: **0.73**					

Results on group B											
	Computer Estimate										
True Label	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	Recall
B0	8								2		0.8
B1		10									1.0
B2			8				1			1	0.8
B3				10							1.0
B4					10						1.0
B5		2			3	4			1		0.4
B6							10				1.0
B7								10			1.0
B8									10		1.0
B9										10	1.0
Precision	1.00	0.83	1.00	1.00	0.77	1.00	0.91	1.00	0.77	0.91	
Overall Accuracy: **0.90**						Macro-average F-measure: **0.89**					

Table 2: Experimental results on the Greek data sets.

To compare individual author categories, Table 2 gives the confusion matrices obtained on the two data sets. Comparing Table 2 to (Stamatatos et al., 2000, Table 6), shows that we obtain better results in every category of group B, and perform slightly better on group A authors, except K. Chalbatzakis (author A1—0.6 versus 0.7 recall).

Note that our performance on group B is much stronger than group A, whereas similar results were obtained on both groups in (Stamatatos et al., 2000). This suggests that our language modeling approach is more effective at capturing author-specific idiosyncrasies in a homogeneous collection like B, but in collections like A where a given author may write of many different types of text, our method is less effective. It appears that sometimes genre might dominate authorship when it comes to style. By looking at the confusion matrices, one can see that authors A1, A3, A5, A6, A7 are mistakenly identified as A2, resulting in a low precision for A2 (0.45). Also, many of the documents produced by A8 are not correctly identified, resulting in a low recall for A8 (0.3). We are investigating whether genre is responsible for this.

4.3 English data

The English data used in our experiments is available from the Alex Catalogue of Electronic Texts.[2] We used the 8 most prolific authors from this collection, shown in Table 3.

	Author Name	Training size word(character)
E0	Charles Dickens	1614258 (9033267)
E1	John Keats	49314 (335676)
E2	John Milton	146446 (868857)
E3	William Shakespeare	645605 (3642829)
E4	Robert L. Stevenson	1036108 (5687003)
E5	Oscar Wilde	102092 (585092)
E6	Ralph W. Emerson	384570 (2201546)
E7	Edgar Allan Poe	307710 (1785067)

Table 3: Authors appearing in the English data set.

To reduce any sparse data problems we might face, we first converted the corpus into lowercase characters and only used 30 most frequent characters in the vocabulary, which comprises over 99% of all character occurrences in the corpus. The best accuracy we obtained was **98%**, which was achieved by a 6-gram model using absolute smoothing. This is excellent performance. However, it is probably due to the distinct idiosyncratic writing styles of these famous authors. We are investigating more challenging data sets.

4.4 Chinese data

Authorship attribution in Chinese normally requires an initial word segmentation phase, followed by a feature extraction process at the word level, as in English. However, word segmentation is itself a hard problem in Chinese, and an improper segmentation may cause insurmountable problems for later prediction phases. We avoid the word segmentation problem by simply operating at the character level.

The Chinese corpus we used in our experiments was also downloaded from the Internet.[3] Eight of the most popular modern Chinese martial art novelists were included in this study, shown in Table 4. One or two novels was selected from each author to be used as training data, and 20 additional novels were used as a test set.

	Author Name	Training size (character)
C0	Gu Long	1466286
C1	Huang Yi	1860555
C2	Jin Yong	979885
C3	Liang Yusheng	1085179
C4	Wen RuiAn	536986
C5	Xiao Yi	875678
C6	Chen Qinyun	857929
C7	Wo Losheng	1554689

Table 4: Authors appearing in the Chinese data.

Note that a significant difference between Chinese and English or Greek is that the Chinese character vocabulary is much larger than the English or Greek character vocabularies. For example, the most commonly used Chinese character set contains 6763 characters. In our experiments, we encountered about 4600 distinct Chinese characters. Therefore, to reduce the sparse data problems we might encounter, we first selected the most frequent 2500 characters as our vocabulary, which comprised about 99% of all character occurrences.

The best overall accuracy we obtained was **94%**, with a 3-gram language model using Witten-Bell smoothing. Once again, this is effective performance, but possibly obtained on an easy data set. We undertake a more detailed analysis below.

5 Analysis

As discussed in Section 2, the perplexity of an n-gram language model depends on several factors, including the context length n, the smoothing technique, and the size of training corpus. Different choices of these factors will result in different perplexities in the test corpus, which could influence the final decision in using Eq. (8). To ascertain the sensitivity of our results to these factors, we assess their influence in turn.

5.1 Influence of Context Length

The context length n is a key factor in n-gram language modeling. A context n that is too small will not capture sufficient information to accurately model character dependencies. On the other hand, a context n that is too large will create sparse data

[2] http://www.infomotions.com/alex/downloads/

[3] http://chineseculture.about.com/library/chinese/blindex.htm

problems in training. Both extremes will result in a larger perplexity than an optimal length. In a word level n-gram model, n must normally be set to 2 or 3 to obtain optimal performance. However, it would seem intuitive that a longer context would be more effective at the character level. The influence of context length n in each of the three languages is illustrated in Figure 1. Here we see that

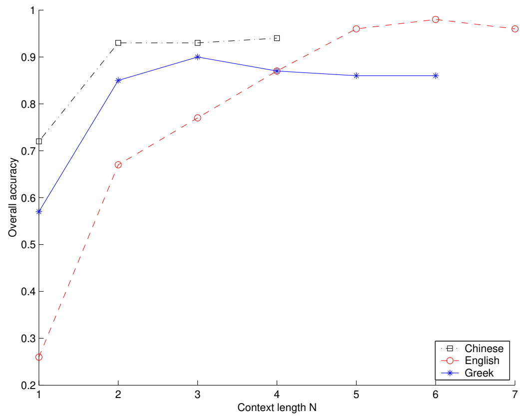

Figure 1: Influence of context length on accuracy.

authorship attribution accuracy initially increases with increasing context, but begins to degrade as sparse data problems begin to take hold. Interestingly, the optimal context length is different in each of the three languages. In each case, the results are not overly sensitive after a minimal context length has been reached.

5.2 Influence of Smoothing Technique

Another key factor affecting the performance of a language model is the smoothing technique used. It has been found that Good-Turing smoothing performs effectively in many contexts (Chen and Goodman, 1998). However, our goal is to make a final decision based on the *ranking* of perplexities, not just their absolute values, which means that the best smoothing technique for language modeling in the sense of perplexity may not be the best choice for text categorization. Indeed, we find that the language models using Good-Turing smoothing do not always perform best in our experiments. The effects of smoothing in each of the three languages is illustrated in Figure 2. Figure 2 shows that in most cases the smoothing technique does not have a significant effect on authorship attribution accuracy, except in one case (Greek) where

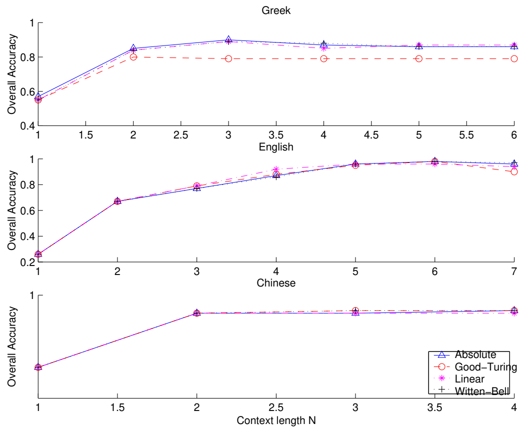

Figure 2: Influence of smoothing on accuracy.

Good-Turing smoothing is not as effective as other techniques. Nevertheless, for the most part, one can use any standard smoothing technique in this problem and obtain comparable performance, because the rankings they produce are almost always the same.

5.3 Influence of Training Size

Clearly, the size of the training corpus can affect the accuracy of a language model. Normally with a larger training corpus more reliable statistics can be recovered which leads to better prediction accuracy on test data. To test the effect of training set size we obtained an additional 10 documents from each author in the group B Greek data set. In fact, this same additional data has been used in (Stamatatos et al., 2001) to improve the accuracy of their method from 71% to 87%. Here we find that the extra training data also improves the accuracy of our method, although not so dramatically. Figure 3 shows the improvement obtained for n-gram language models using absolute smoothing.

Here we can see that indeed the extra training data uniformly improves attribution accuracy. On the augmented training data the best model (3-gram) now obtains a **92%** attribution accuracy, compared to the 90% we obtained originally. Moreover, this improves the best result obtained in (Stamatatos et al., 2001) of 87%. However, our improvement (90% to 92%) is not nearly as great as that obtained by (Stamatatos et al., 2001) (71% to 87%). However, this could be due to the fact that it is harder to reduce a small prediction error.

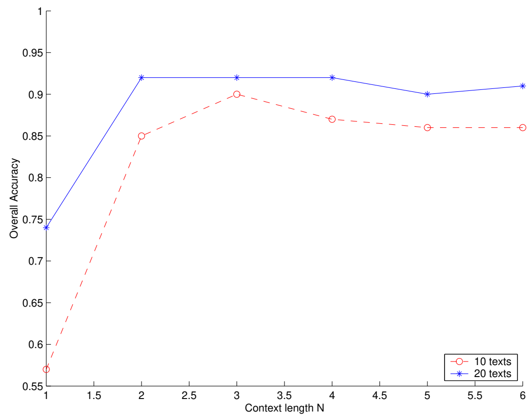

Figure 3: Influence of training size on accuracy.

6 Related Work

In principle, any language modeling technique can be used to perform authorship attribution based on Eq. (8). However, n-gram models are extremely simple and have been found to be effective in many applications. For example, character level n-gram language models can be easily applied to any language, and even non-language sequences such as DNA and music. Character level n-gram models are widely used in text compression—e.g., the PPM model (Bell et al., 1990)—and have recently been found to be effective in text mining problems as well (Witten et al., 2000). Text categorization with n-gram models has also been attempted by (Cavnar and Trenkle, 1994). However, they use n-grams as features for traditional feature selection, and then deploy classifiers based on calculating feature-vector similarities. In the domain of language independent text categorization, Apt et al.(Apt et al., 1994) have used word-based language modeling techniques for both English and German. However, their techniques do not apply to Asian languages where word segmentation remains a significant problem.

7 Conclusion

We have presented a novel approach to automated authorship attribution that is based on character level n-gram language modeling. We have demonstrated our approach on three different languages and obtained effective performance in each case. In particular, we have obtained a 18% accuracy improvement for one Greek data set (group B), over a state of the art system that uses significantly deeper NLP techniques. The simplicity of our method, however, makes it immediately applicable to any natural language and yields effective baseline performance. We have experimentally analyzed the influence of various factors that can affect the accuracy of our approach and found that, for the most part, our results are fairly robust to perturbations of the method. We are investigating alternative data sets and additional languages.

8 Acknowledgments

We thank E. Stamatatos for supplying us with the Greek data. Research supported by Bell University Labs and MITACS.

References

A. Aizawa. 2001. Linguistic Techniques to Improve the Performance of Automatic Text Categorization. In *Proceedings 6th NLP Pac. Rim Symp. NLPRS-01*.

C. Apté, F. Damerau and S. Weiss. 1994. Toward Language Independent Automated Learning of Text Categorization Models. In *Proceedings SIGIR-94*.

T. Bell, J. Cleary and I. Witten. 1990. *Text Compression*. Prentice Hall.

W. Cavnar and J. Trenkle. 1994. N-Gram-Based Text Categorization. In *Proceedings SDAIR-94*.

S. Chen and J. Goodman. 1998. An Empirical Study of Smoothing Techniques for Language Modeling. *TR-10-98*, Harvard.

M. Ephratt. 1997. Authorship Attribution - the Case of Lexical Innovations. In *Proc. ACH-ALLC-97*.

D. Holmes and R. Forsyth. 1995. The Federalist Revisited: New Directions in Authorship Attribution. In *Literary and Linguistic Computing*, 10, 111-127.

H. Love, (2002). *Attributing Authorship: An Introduction*. Cambridge University Press.

S. Scott and S. Matwin. 1999. Feature Engineering for Text Classification. In *Proceedings ICML-99*.

E. Stamatatos, N. Fakotakis and G. Kokkinakis. 1999. Automatic Authorship Attribution. In *EACL-99*

E. Stamatatos, N. Fakotakis and G. Kokkinakis. 2000. Automatic Text Categorization in Terms of Genre and Author. *Comput. Ling.*, 26 (4), pp. 471-495.

E. Stamatatos, N. Fakotakis and G. Kokkinakis. 2001. Computer-based Authorship Attribution without Lexical Measures *Computers and the Humanities*, 35, pp. 193-214.

I. Witten, Z. Bray, M. Mahoui and W. Teahan. 1999. Text mining: A New Frontier for Lossless Compression. *Proceedings IEEE Data Compression 97*