Towards Scalable Data-Driven Authorship Attribution

Marc-Allen Cartright and **Michael Bendersky**
Center for Intelligent Information Retrieval
University of Massachusetts, Amherst
Amherst, MA 01003
{irmarc,bemike}@cs.umass.edu

Abstract

Traditional authorship attribution approaches have made attempts at capturing features that were designed heuristically – researchers guessed at which aspects of language would best separate one author from another and then performed experiments to see how valid their assumptions were. While this approach has met some success, it also proves to be unscalable – most test collections to date have been on the size of 10 or less authors, which in the age of internet-style publication is an unrealistically low quantity. We believe that this approach to feature selection for authorship attribution adds unnecessary complexity to what the task really seems to be: a multi-class classification problem, and one where the most useful features can be easily discovered using a standard dimensionality reduction technique. We demonstrate the use of such a technique to dramatically reduce the number of used features for authorship attribution using an implementation of Support Vector Machines.

1 Introduction

The task of *authorship attribution* (henceforth, *AA*) is assigning an author to a particular document or work. While the number of plausible authors has ballooned due to the advent of the internet, the collections used for *AA* research have more or less remained static. Multiple approaches to *AA* have been developed, and many have been shown to be experimentally successful, however relatively little work has been done towards demonstrating the effectiveness of these approaches to large-scale data sets. In this work we test the plausibility of using statistical inference to automatically select a sufficient subset of all possible features to use in a standard machine-learning approach to *AA*. We use a corpus of works from various time periods and multiple languages to provide a realistic, heterogeneous representation of the *AA* problem.

2 Previous Work

Throughout most of its history, researchers typically approach the task of *AA* by manually selecting a subset of stylometric features such as function words (Zhao et al., 2006), POS counts (Stamatatos et al., 2001), punctuation and sentence length (Stamatatos et al., 2001; Holmes, 1998) or letter n-grams (Khmelev and Tweedie, 2003; Keselj et al., 2003). Despite several studies that attempt to evaluate these kinds of features (Stamatatos et al., 2001; Holmes, 1998; Koppel and Schler, 2003), the evaluation has been mainly performed on collections that pale in comparison to what we would consider standard-sized collections today.

More recent work such as (Diederich et al., 2000) and (Koppel et al., 2006) has provided us with some empirical basis for our investigation; the former work makes a compelling argument for SVMs being the most appropriate technique to approach this task, while the latter introduces what we feel to be the most realistic setting for the *AA* task to date.

3 Our Approach

3.1 Corpus

Our source data comes from Project Gutenberg[1], an online source of free electronic books. Currently the project contains over $17,000$ books in various languages. We selected 100 authors with multiple works contained in the collection. Table 1 shows some of the statistics of the collection. For our experiments, we partitioned the collection into 10 overlapping data sets, ranging from 10 to 100 authors.

# authors	# books per author	# unique terms/author	# terms per book
100	30	5,367	58,240

Table 1: Collection statistics. Average counts of books per authors and terms per book are presented.

In addition to the size of the collection, which distinguishes it from collections previously used for *AA*, there are several interesting observations that differentiate the data sets studied here from other data sets that are typically used for text classification:

- **High dimensionality.** The richness of language in our setting leads to even larger number of unique terms than is usual in text classification. A classical *Reuters-21578* set contains $27,658$ distinct terms, almost 3 times less than the smallest of our data sets ($73,743$ unique terms for 10 authors).

- **Small number of large documents.** Collections used for text classification evaluation typically consist of large number of relatively short documents (e.g., newswire, academic paper abstracts, etc.). Our data sets, on the other hand consist of a small number of very long documents (books). Again, compared to *Reuters-21578* that contains above $21,578$ documents, our largest data set contains only $3,003$ documents. When considering the semi-supervised classification task, this leads to an intriguing setting, where train examples are scarce, but each example is semantically rich.

- **Non-topical class relations.** Typically, document classification is based on document topicality. In our data sets, we instead assume that each author can be associated with a distinct term distribution, but this distribution is not necessarily topic-driven.

3.2 Data Normalization and Preparation

To normalize the data for feature extraction the following steps were taken:

- **Anonymization.** Any mention of the author's name was removed from the text of the book. In addition, the book header (first 50 lines of the book) was removed in order to prevent any metadata from influencing the classification process.

- **Stemming and Stopwords removal.** We used the standard INQUERY (Allan et al., 2000) stopwords list and the well-known Porter stemming algorithm[2].

- **Indexing.** To facilitate efficient feature extraction, the normalized book collection was indexed using the INDRI search engine[3] .

3.3 Feature Construction and Selection

We take the *bag of words* approach, which was proven to be successful in both information retrieval (Salton et al., 1975) and document classification (Joachims, 2002) settings. In other words, we represent each book as a vector of length-normalized term counts. As term vectors are highly-dimensional and sparse, classification using all features becomes intractable for large datasets. Instead, we consider feature selection based on mutual information.

Mutual Information. Feature selection based on mutual information of the features is a common procedure in document classification (Joachims, 2002). In our case, term t is ranked by mutual information with the author class variable a, or formally

$$MI(t) = \sum_{a \in \mathbf{A}} P(a)P(t|a) \log \frac{P(t|a)}{P(t)}, \quad (1)$$

[1] http://www.gutenberg.org/wiki/Main_Page

[2] http://www.tartarus. org/martin/PorterStemmer

[3] http://www.lemurproject.org/indri/

where $\mathbf{A}$ is the set of all possible author classes, and probabilities are computed using maximum-likelihood estimates.

Number of features. We initially used a standard approach, where the number of features to use in the learning algorithm is set a priori to some constant C. However, we observe that as the number of possible classes (authors) grows, so does the perplexity of a distribution over the terms in the collection (note that each new author adds on average more than $5,000$ unique terms to the collection – see Table 1). If we assume Zipf's Law (Zipf, 1949) holds for our collection, the number of potential features will monotonically increase sublinearly as the number of authors increases. Hence, we assume that number of features to use in the learning algorithm should increase log-linearly with the number of potential authors. That is, for collection containing n_a authors, the number of features used in the learning algorithm will be

$$\nu(n_a) = C(1 + \alpha \log n_a), \qquad (2)$$

where $\alpha \in [0, 1]$ is a damping factor.

4 Experimental Results

As described above, in our experiments we consider 10 overlapping data sets, the smallest containing 162 books from 10 authors, and the largest containing $3,003$ books from 100 authors, all normalized according to the procedure outlined in Section 3.2.

All authorship attribution experiments are done using $SVM^{multiclass}$. For a linear kernel, used in our experiments, $SVM^{multiclass}$ is quite efficient, and its runtime scales linearly with the number of training examples[4]. We use a 5-fold cross-validation to evaluate an average accuracy of the *AA* task on the various data sets.

We run three types of classification experiments: *SVM[all]*, *SVM[C]* and *SVM[$\nu(n_a)$]*. *SVM[all]* uses all features (terms) to perform the *AA* task, while *SVM[C]* and *SVM[$\nu(n_a)$]* use a subset of C and $\nu(n_a)$ (Equation 2) features, respectively, selected by the mutual information metric (Equation 1). We set $C = 500$, $\alpha = 0.64$ in our experiments.

[4]http://svmlight.joachims.org/

Runtime cpu/sec			
# authors	10	20	30
SVM[all]	93.72	434.16	871.38
SVM[C]	9.99	28.32	148.48
SVM[$\nu(n_a)$]	10.65	46.28	248.80

% Accuracy			
# authors	10	20	30
SVM[all]	97.54	93.22	93.59
SVM[C]	92.58	90.25	88.17
SVM[$\nu(n_a)$]	95.07	92.79	92.31

Table 2: Comparison of *AA* task performance for 10, 20 and 30 author data sets. The top table compares runtime in seconds/cpu, while the bottom table compares average accuracy.

SVM[all]* vs. *SVM[C]* and *SVM[$\nu(n_a)$] Table 2 illustrates the performance in terms of runtime and accuracy for all three classification methods on the three of our smallest datasets. We note that the results of our experiments seem to support the hypothesis that a limited number of features helps to sustain a reasonable accuracy across growing data sets, while dramatically reducing the runtime. Encouraged by these results we turn to examine the performance of *SVM[C]* and *SVM[$\nu(n_a)$]* on the larger data sets.

SVM[C]* vs. *SVM[$\nu(n_a)$] In terms of average accuracy (right-hand graph at Figure 1), the performance of both *SVM[C]* and *SVM[$\nu(n_a)$]* remains quite stable as the number of authors goes up: in both cases, accuracy drops only 10%, when moving from 10 to 100 authors data set. When juxtaposing the relative performance of *SVM[C]* vs. *SVM[$\nu(n_a)$]*, we note that adding more features uniformly improves accuracy, as expected. Comparing the accuracy results vectors for these two methods using Wilcoxon rank-sum test, shows that this improvement is statistically significant ($p < 0.02$).

The left-hand graph at Figure 1 demonstrates that both for *SVM[C]* and *SVM[$\nu(n_a)$]* the runtime of the classification procedure does not scale too well as the number of possible classes (authors) grows. *SVM[$\nu(n_a)$]* seems to be especially sensitive to this — note the sharp runtime increase when number of authors grows from 70 to 80.

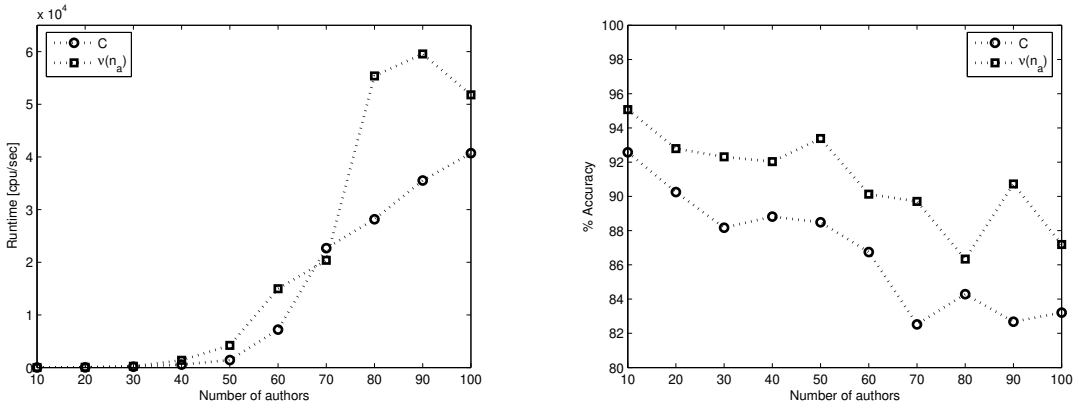

Figure 1: Comparison of *SVM[C]* and *SVM[$\nu(n_a)$]* performance.

5 Conclusions

We have shown that a machine learning approach to *AA* need not fall victim to the curse of dimensionality when encountering large data sets. A simple feature selection procedure using techniques from information theory can reduce the number of possible features by orders of magnitude while still maintaining classification accuracy to within a few percent of using all of the available features.

We believe that the future direction of feature selection for *AA* will be necessarily driven by methods that can adjust to the changing characteristics of the data, and therefore require methods that emphasize certain features using statistical information present in those data sets.

Another important direction in *AA* research is tackling the problem of scaling the performance of exisitng multi-class categorization models for large-scale corpora with hundreds or thousands of candidate classes.

Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval and in part by NSF grant #IIS-0534383. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsor.

References

J. Allan, M.E. Connell, W.B. Croft, F.F. Feng, D. Fisher, and X. Li. 2000. INQUERY and TREC-9. *Proceedings of the Ninth Text Retrieval Conference (TREC-9)*, pages 551–562.

J. Diederich, J. Kindermann, E. Leopold, and G. Paass. 2000. Authorship attribution with support vector machines. *Applied Intelligence*, pages 109–123.

D. I. Holmes. 1998. The evolution of stylometry in humanities computing. *Literary and Linguistic Computing*, 13(3):111–7.

T. Joachims. 2002. *Learning to Classify Text Using Support Vector Machines*. Kluwer Academic Publishers.

Vlado Keselj, Fuchun Peng, Nick Cercone, , and Calvin Thomas. 2003. N-gram-based author profiles for authorship attribution. In *PACLING'03: Proceedings of the Conference Pacific Association for Computational Linguistics*, pages 255–264. Pacific Association for Computational Linguistics.

D. V. Khmelev and F. J. Tweedie. 2003. Using Markov Chains for Identification of Writer. *Literary and Linguistic Computing*, 16(3):299–307.

M. Koppel and J. Schler. 2003. Exploiting stylistic idiosyncrasies for authorship attribution. In *Proceedings of IJCAI'03 Workshop on Computational Approaches to Style Analysis and Synthesis*. International Joint Conferences on Artificial Intelligence.

Moshe Koppel, Jonathan Schler, Shlomo Argamon, and Eran Messeri. 2006. Authorship attribution with thousands of candidate authors. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 659–660, New York, NY, USA. ACM.

G. Salton, A. Wong, and C. S. Yang. 1975. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620.

E. Stamatatos, N. Fakotakis, and G. Kokkinakis. 2001. Computer-Based Authorship Attribution Without Lexical Measures. *Computers and the Humanities*, 35(2):193–214.

Y. Zhao, J. Zobel, and P. Vines. 2006. Using relative entropy for authorship attribution. *Proc. 3rd AIRS Asian Information Retrieval Symposium, Springer*, pages 92–105.

George K. Zipf. 1949. *Human Behavior and the Principle of Least Effort*. Addison-Wesley, Cambridge, Massachusetts.