Authorship Verification as a One-Class Classification Problem

Moshe Koppel Jonathan Schler
Dept. of Computer Science
Bar-Ilan University
Ramat-Gan, Israel

Abstract

In the authorship verification problem, we are given examples of the writing of a single author and are asked to determine if given long texts were or were not written by this author. We present a new learning-based method for adducing the "depth of difference" between two example sets and offer evidence that this method solves the authorship verification problem with very high accuracy. The underlying idea is to test the rate of degradation of the accuracy of learned models as the best features are iteratively dropped from the learning process.

1. Introduction

In the *authorship attribution* problem, we are given examples of the writing of a number of authors and are asked to determine which of them authored given anonymous texts (Mosteller & Wallace 1964, Holmes 1998). If it can be assumed for each test document that one of the specified authors is indeed the actual author, the problem fits the standard paradigm of a text categorization problem (Sebastiani 2003).

In the *authorship verification* problem, we are given examples of the writing of a single author and are asked to determine if given texts were or were not written by this author. As a categorization problem, verification is significantly more difficult than attribution and little, if any, work has been performed on it in the learning community. If, for example, all we wished to do is to determine if a text was written by Shakespeare or Marlowe, it would be sufficient to use their respective known writings, to construct a model distinguishing them, and to test the unknown text against the model. If, on the other hand, we need to determine if a text was written by Shakespeare or not, it is very difficult – if not impossible – to assemble an exhaustive, or even representative, sample of not-Shakespeare. The situation in which we suspect that a given author may have written some text but do not have an exhaustive list of alternative candidates is a common one. The problem is complicated by the fact that a single author may consciously vary his or her style from text to text for many reasons or may unconsciously drift stylistically over time. Thus researchers must learn to somehow distinguish between relatively shallow differences that reflect conscious or unconscious changes in an author's style and deeper differences that reflect styles of different authors.

Verification is thus essentially a one-class problem. A fair amount of work has been done on one-class problems, especially using support vector machines (Manevitz & Yousef 2001, Scholkopf et al 2001, Tax 2001). There are, however, two important points that must be noted. First, in authorship verification we do not actually lack for negative examples. The world is replete with texts that were, for example, not written by Shakespeare. However, these negative examples are not representative of the entire class. Thus, for example, the fact that a particular text may be deemed more similar to the known works of Shakespeare than to those of some given set of other authors does not by any means constitute solid evidence that the text was authored by Shakespeare rather than by some other author not considered at all. We will consider how to make proper limited use of whatever partial negative information is available for the authorship verification problem.

A second distinction between authorship verification and some one-class problems is that if the text we wish to attribute is long – and in this paper we will consider only long texts – then we can chunk the text so that we effectively have multiple examples which are known to be either all written by the author or all not written by the author. Thus, a better way to think about authorship verification is that we are given two example sets and are asked whether these sets were generated by a single generating process (author) or by two different processes.

The central novelty of this paper is a new method for adducing the depth of difference between two example sets, a method that may have far-reaching consequences for determining the reliability of classification models.

The underlying idea is to test the rate of degradation of the accuracy of learned models as the best features are iteratively dropped from the learning process.

We will show that this method provides an extremely robust solution to the authorship verification problem that is independent of language, period and genre. This solution has already been used to settle at least one outstanding literary problem.

2. Authorship Attribution with Two Candidates

Since our methods will build upon the standard methods for handling authorship attribution between two candidate authors, let us begin by briefly reviewing those standard methods.

We begin by choosing a feature set consisting of the kinds of features that might be used consistently by a single author over a variety of writings. Typically, these features might include frequencies of function words (Mosteller and Wallace 1964), syntactic structures (Baayen et al 1996, Stamatatos et al 2001), parts-of-speech n-grams (Koppel et al 2002), complexity and richness measures (such as sentence length, word length, type/token ratio) (Yule 1938, Tweedie and Baayen 1998, de Vel et al 2002) or syntactic and orthographic idiosyncrasies (Koppel and Schler 2003). Note that these feature types contrast sharply with the content words commonly used in text categorization by topic.

Having constructed the appropriate feature vectors, we continue, precisely as in topic-based text categorization, by using a learning algorithm to construct a distinguishing model. Although many learning methods have been applied to the problem, including multivariate analysis, decision trees and neural nets, a good number of studies have shown that linear separators work well for text categorization (Yang 1999). Linear methods that have been used for text categorization include Naïve Bayes (Mosteller and Wallace 1964, Peng et al 2004), which for the two-class case is a linear separator, Winnow and Exponential Gradient (Lewis et al 1996, Dagan et al 1997, Koppel et al 2002) and linear support vector machines (Joachims 1999, de Vel et al 2002, Diederich et al 2003).

Finally, the effectiveness of the methods used is assessed using k-fold cross-validation or bootstrapping.

This general framework has been used to convincingly solve a number of real world authorship attribution problems (e.g. Mosteller & Wallace 1964, Matthews & Merriam 1993, Holmes et al 2001, Binongo 2003).

3. Authorship Verification: Naïve Approaches

Is there some way to leverage these methods to solve the verification problem in which we are asked if some mystery book was written by author A without being offered a closed set of alternatives? One possibility that suggests itself is to concoct a mishmash of works by other authors and to learn a model for A vs. not-A. This approach is problematic, though not entirely useless, and we will come back to it later. For now, let us first try to handle the problem with *no* negative examples at all. One approach that sounds at first blush as if it might work is this: let's chunk the known works of A and the mystery work X to generate some sufficiently large set of examples of each. Now we can learn A vs. X and assess the extent of the difference between A and X using cross-validation. If it is easy to distinguish between them, i.e. if we obtain high accuracy in cross-validation, then presumably we could conclude that A and X are by different authors.

This method does not work well at all. To understand why, let's consider a real-world example. We are given known works by three 19th century American novelists, Herman Melville, James Fenimore Cooper and Nathaniel Hawthorne. For each of the three authors, we are asked if that author was or was not also the author of *The House of Seven Gables* (henceforth: *Gables*). Using the method just described and using a feature set consisting of the 250 most frequently used words in A and X (details below), we find that we can distinguish *Gables* from the works of each author with cross-validation accuracy of above 98%. If we were to conclude, therefore, that none of these authors wrote *Gables*, we would be wrong: Hawthorne wrote it.

4. A New Approach: Unmasking

If we look closely at the models that successfully distinguish *Gables* from Hawthorne's other work (in this case, *The Scarlet Letter*), we find that a small number of features are doing all the work of distinguishing between them. These features include *he* (more frequent in *The Scarlet Letter*) and *she* (more frequent in *Gables*). The situation in which an author will use a small number of features in a consistently different way between works is typical. These differences might result from thematic differences between the works, from differences in genre or purpose, from chronological stylistic drift, or from deliberate attempts by the author to mask his or her identity (as we shall see below).

This problem can be overcome using a new technique we call "unmasking". The intuitive idea of unmasking is to iteratively remove those features that are most useful for distinguishing between A and X and to gauge the speed with which cross-validation accuracy degrades as more features are removed. Our main

hypothesis is that if A and X are by the same author, then whatever differences there are between them will be reflected in only a relatively small number of features, despite possible differences in theme, genre and the like.

In Figure 1, we show the result of unmasking when comparing *Gables* to known works of Melville, Cooper and Hawthorne. This graph illustrates our hypothesis: when comparing *Gables* to works by other authors the degradation as we remove distinguishing features from consideration is slow and smooth but when comparing it to another work by Hawthorne, the degradation is sudden and dramatic. This illustrates that once a small number of distinguishing markers are removed, the two works by Hawthorne become increasingly hard to distinguish from each other. In the following section, we will show how the suddenness of the degradation can be quantified in a fashion optimal for this task and we will see that the phenomenon illustrated in the *Gables* example holds very generally.

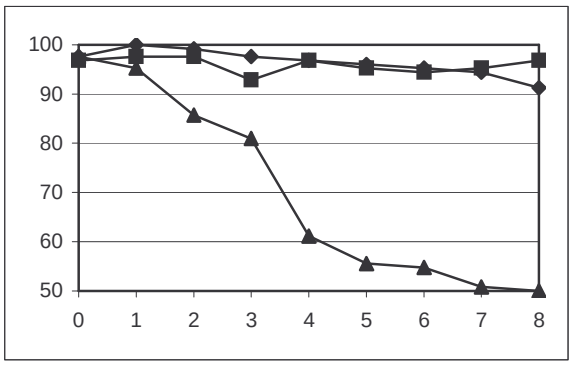

Figure 1. Ten-fold cross-validation accuracy of models distinguishing *House of Seven Gables* from each of Hawthorne, Melville and Cooper. *x*-axis represents number of iterations of eliminating best features at previous iteration. The curve well below the others is that of Hawthorne, the actual author.

5. Experimental Results

5.1 Corpus

We consider a collection of twenty-one 19th century English books written by ten different authors and spanning a variety of genres. We chose all electronically available books by these authors that were sufficiently large (above 500K) and that presented no special formatting challenges. The full list is shown in Table 1. Our objective is to run 209 independent authorship verification experiments representing all possible author/book pairs (21 books × 10 authors but excluding the pair Emily Bronte/*Wuthering Heights* which can't be tested since it is the author's only work).

For the sake of all the experiments that follow, we chunk each book into approximately equal sections of at least 500 words without breaking up paragraphs. For each author A and each book X, let A_X consists of all the works by A in the corpus unless X is in fact written by A, in which case A_X consists of all works by A except X. Our objective is to assign to each pair $<A_X,X>$ the value *same-author* if X is by A and the value *different-author* otherwise.

Table 1. The list of books used in our experiments.

Group	Author	Book	#Chunks
American Novelists	Hawthorne	Dr. Grimshawe's Secret	75
		House of Seven Gables	63
	Melville	Redburn	51
		Moby Dick	88
	Cooper	The Last of the Mohicans	49
		The Spy	63
		Water Witch	80
American Essayists	Thoreau	Walden	49
		A Week on Concord	50
	Emerson	Conduct Of Life	47
		English Traits	52
British Playwrights	Shaw	Pygmalion	44
		Misalliance	43
		Getting Married	51
	Wilde	An Ideal Husband	51
		Woman of no Importance	38
Bronte Sisters	Anne	Agnes Grey	45
		Tenant Of Wildfell Hall	84
	Charlotte	The Professor	51
		Jane Eyre	84
	Emily	Wuthering Heights	65

5.2 Baseline: One-class SVM

In order to establish a baseline, for each author A in the corpus and each book X, we use a one-class SVM (Chang & Lin 2001) on the 250 most frequent words in A_X to build a model for A_X. We then test each book X

against the model of each A_X. We assign the pair $<A_X,X>$ the value *same-author* if more than half the chunks of X are assigned to A_X. This method performs very poorly. Of the 20 pairs that should have been assigned the value *same-author*, only 6 are correctly categorized, while 46 of the 189 pairs that should be assigned the value *different-author* are incorrectly classified. These results hold using an RBF kernel; using other kernels or using a threshold other than half (the number of chunks assigned to the class) only degrades results.

5.3 Unmasking Applied

Now let us introduce the details of our new method based on our observations above regarding iterative elimination of features. We choose as an initial feature set the 250 words with highest average frequency in A_X and X (that is, the average of the frequency in A_X and the frequency in X, giving equal weight to A_X and X). Using an SVM with linear kernel we run the following unmasking scheme:

1. Determine the accuracy results of a ten-fold cross-validation experiment for A_X against X.
2. For the model obtained in each fold, eliminate the 3 most strongly-weighted positive features and the 3 most strongly-weighted negative features.
3. Go to step 1.

In this way, we construct degradation curves, as in Figure 2, for each pair $<A_X,X>$.

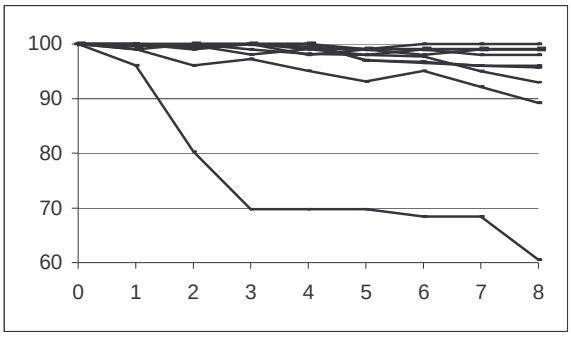

Figure 2. Unmasking *An Ideal Husband* against each of the ten authors. The curve below all the authors is that of Oscar Wilde, the actual author. (Several curves are indistinguishable.)

5.4 Meta-learning: Identifying Same-Author Curves

We wish now to quantify the difference between *same-author* curves and *different-author* curves. To do so, we first represent each curve as a numerical vector in terms of its essential features. These features include, for i = 0,…,9:

- accuracy after i elimination rounds
- accuracy difference between round i and i+1
- accuracy difference between round i and i+2
- i^{th} highest accuracy drop in one iteration
- i^{th} highest accuracy drop in two iterations

We sort these vectors into two subsets: those in which A_X and X are the by same author and those in which A_X and X are by different authors. We then apply a meta-learning scheme in which we use learners to determine what role to assign to various features of the curves.

Technically speaking, we have 189 distinct *different-author* curves but only 13 distinct *same-author* curves, since for authors with exactly two works in our corpus, the comparison of A_X with X is identical for each of the two books. To illustrate the ease with which *same-author* curves can be distinguished from *different-author* curves, we note that for all 13 distinct *same-author* curves, it holds that:

- accuracy after 6 elimination rounds is lower than 89% *and*
- the second highest accuracy drop in two iterations is greater than 16%.

These two conditions hold for only 5 of the 189 *different-author* curves.

5.5 Accuracy Results: Leave-one-book-out Tests

In order to honestly assess the accuracy of the method, we use the following cross-validation methodology. For each book B in our corpus, we run a trial in which B is completely eliminated from consideration. We use unmasking to construct curves for all author/book pairs $<A_X,X>$ (where B does not appear in A_X and is not X) and then we use a linear SVM to meta-learn to distinguish *same-author* curves from *different-author* curves. Then, for each author A in the corpus, we use unmasking to construct a curve for the pair $<A_B,B>$ and use the meta-learned model to determine if the curve is a *same-author* curve or a *different-author* curve.

Using this testing protocol, we obtain the following results: All but one of the twenty *same-author* pairs are correctly classified. The single exception is *Pygmalion* by George Bernard Shaw. In addition, 181 of 189 *different-author* pairs were correctly classified. Among the exceptions were the attributions of *The Professor* by Charlotte Bronte to each of her sisters. Thus, we obtain overall accuracy of 95.7% with errors almost identically distributed between false positives and false negatives.

6. Extension: Using Negative Examples

Until now we have not used any examples of non-A writing to help us construct a model of author A. Of course, plenty of examples of non-A writing exist; they are simply neither exhaustive nor representative. We now consider how use can be made of such data. Suppose, then, that we have available the works of several authors roughly filling the same profile as A in terms of geography, chronology, culture and genre. To make matters concrete, suppose we are considering whether some book X was written by Melville. We could use the works of Hawthorne and Cooper as examples of non-Melville writing and learn a model for Melville against non-Melville. Assuming we can do so successfully, we can then test each example of X to see if it assigned by the model to Melville or to not-Melville. If many are assigned to not-A, it might be reasonable to conclude that X is not by the same author as A. But if it turns out that many, or even all, examples of X are assigned to Melville, could we reasonably conclude that Melville wrote the book? Certainly not, since if the book were, for example, *Uncle Tom's Cabin*, we would be led astray. Harriet Beecher Stowe's writing may be more similar in some respects to Melville than to Hawthorne and Cooper, but she is not Melville.

Still we can exploit this method to possibly eliminate some false positives from the unmasking phase. Our elimination method works as follows. For each author A, choose other authors of the same type – let's call them $A_1,\ldots,A_n$ – and allow them to collectively represent the class not-A. In our corpus, we consider four types as indicated in Table 1. We learn a model for A against not-A and we learn models for each A_i against not-A_i. Assuming that k-fold cross-validation results for each of these models are satisfactory, we test all the examples in X against each one of these models. Let A(X) be the percentage of examples of X classed as A rather than not A; define $A_i(X)$ analogously. Then if A(X) is not greater than $A_i(X)$ for all i, conclude that A is not the author of X. Otherwise conclude nothing.

We use this elimination method to augment unmasking. For those cases where unmasking concludes that A authored X, we allow the elimination method to overrule it if the elimination method indicates that A was not the author of X. The elimination method is not used at all for cases where unmasking concludes that A did not author X.

In our experiment, using the elimination method in this way resulted in the introduction of a single new misclassification: Thoreau was incorrectly concluded not to have written *A Week on Concord*. At the same time, all of the pairs previously misclassified as *same-author* were corrected. Overall, then, the augmented method classed all 189 *different-author* pairs and 19 of 21 *same-author* pairs correctly.

In Figure 3, we summarize the entire algorithm including both unmasking and (optional) elimination. Note that although we introduced the elimination method after the unmasking method in our exposition, for purposes of the pseudo-code it is more natural to present the elimination method as a filter prior to the unmasking method.

```
Given: anonymous book X, works of suspect author A

Step 1 – Elimination (optional)

if alternative authors {A1,…,An} exist then
{
   Build model M for classifying A vs. all other
                                  alternative authors
   Test each chunk of X with built model M
   foreach alternative author Ai
   {
      Build model Mi for classifying Ai vs. {A ∪ all
                                  other alternative authors}
      Test each chunk of X with built model Mi
   }
   if number of chunks assigned to Ai > number of
   _____________chunks assigned to A then
         return different-author
}


Step 2 - Unmasking

Build degradation curve <A,X>
Represent degradation curve as feature vector (see text)
Test degradation curve vector (see text)
   if test result positive
      return same-author
   else
      return different-author


Method Build Degradation Curve:

Use 10 fold cross validation for A against X
foreach fold
{
    Do 10 iterations
    {
          Build a model for A against X
          Evaluate accuracy results
          Add accuracy number to degradation curve
          Remove 6 top contributing features from data
     }
}
```

Figure 3: Overview of the authorship verification algorithm.

7. Solution to a Literary Mystery: The Case of the Bashful Rabbi

Finally, we apply our method to an actual literary mystery. Ben Ish Chai was the leading rabbinic scholar in Baghdad in the late 19th century. Among his vast literary legacy are two collections of Hebrew-Aramaic responsa (letters written in response to legal queries). The first, *RP (Rav Pe'alim)* includes 509 documents known to have been authored by Ben Ish Chai. The second, *TL (Torah Lishmah)* includes 524 documents that Ben Ish Chai claims to have found in an archive. There is ample historical reason to believe that he in fact authored the manuscript but did not wish to take credit for it for personal reasons.

For the sake of comparison, we also have four more collections of responsa written by four other authors working in the same area during the same period. While these far from exhaust the range of possible authors, they collectively constitute a reasonable starting point. There is no reason to believe that any of these authors wrote *TL*.

In any event, the elimination method handily eliminates all candidates but Ben Ish Chai. We now wish to use unmasking to check if Ben Ish Chai is indeed the author. Unmasking is particularly pertinent here, since Ben Ish Chai did not wish to be identified as the author and there is evidence that he may have deliberately altered his style to disguise his authorship. In Figure 4, we show the results of unmasking for *TL* against Ben Ish Chai as well as, for comparison, each of the other four candidate authors.

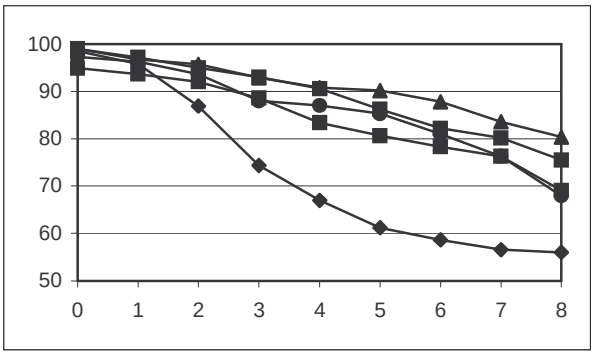

Figure 4. Unmasking TL against Ben Ish Chai and four impostors.

The curve for Ben Ish Chai is the one far below those of the others. Applying the formula learned above confirms what is suggested visually by the figure: Ben Ish Chai was indeed the author of TL. It is particularly interesting to note that the curves obtained in this experiment on Hebrew-Aramaic legal letters are quite similar to those obtained on 19th century English literature.

8. Conclusions

The essentials of two-class text categorization are fairly well understood. We have shown in this paper that by using ensembles of text-categorization results as raw material for meta-level analysis, we are able to solve a more difficult and sophisticated problem such as authorship verification. Even when we completely ignore negative examples and thus treat authorship verification as a true one-class classification problem, our methods obtain extremely high accuracy on out-of-sample author/book pairs. When we use just a bit of non-representative negative data, classification is even better.

Nothing in our method is tied to any particular language, period or genre and some anecdotal evidence suggests that similar results can be obtained as these parameters are varied. More experiments are required to confirm this hypothesis.

The unmasking method suggested here might find more general application beyond the particular case of authorship verification considered here. Unmasking should work generally as a measure of the true "depth of difference" between two example sets.

9. Bibliography

Baayen, H., H. van Halteren, F. Tweedie (1996). Outside the cave of shadows: Using syntactic annotation to enhance authorship attribution, *Literary and Linguistic Computing*, 11, 1996.

Binongo, J.N.G. (2003). Who wrote the 15th Book of Oz? An application of multivariate analysis to authorship attribution. *Chance* 16(2), pp. 9-17.

Chang, C.C. and Lin, C. (2001) LIBSVM: a Library for Support Vector Machines (Version 2.3)

Dagan, I., Y. Karov, D. Roth (1997), Mistake-driven learning in text categorization, in *EMNLP-97: 2nd Conf. on Empirical Methods in Natural Language Processing*, 1997, pp. 55-63.

De Vel, O., M. Corney, A. Anderson and G. Mohay (2002), E-mail Authorship Attribution for Computer Forensics, in *Applications of Data Mining in Computer Security*, Barbará, D. and Jajodia, S. (eds.), Kluwer.

Diederich, J., J. Kindermann, E. Leopold and G. Paass (2003), Authorship Attribution with Support Vector Machines, *Applied Intelligence 19(1)*, pp. 109-123

Holmes, D. (1998). The evolution of stylometry in humanities scholarship, *Literary and Linguistic Computing*, 13, 3, 1998, pp. 111-117.

Holmes, D. I., L. Gordon, and C. Wilson (2001), A Widow and her Soldier: Stylometry and the American

Civil War, *Literary and Linguistic Computing 16(4)*, pp. 403-420

Joachims, T. (1998) Text categorization with support vector machines: learning with many relevant features. In *Proc. 10th European Conference on Machine Learning ECML-98*, pages 137-142

Koppel, M., S. Argamon and A. Shimoni (2002), Automatically categorizing written texts by author gender, *Literary and Linguistic Computing 17(4)*, pp. 401-412

Koppel, M. and J. Schler (2003), Exploiting Stylistic Idiosyncrasies for Authorship Attribution, in *Proceedings of IJCAI'03 Workshop on Computational Approaches to Style Analysis and Synthesis*, pp. 69-72.

Lewis, D., R. Schapire, J. Callan, R. Papka (1996). Training algorithms for text classifiers, in *Proc. 19th ACM/SIGIR Conf. on R&D in IR*, 1996, pp. 306-298.

Manevitz, L. M. and M. Yousef (2001). One-class svms for document classification,. *Journal of Machine Learning Research* 2: 139-154

Matthews, R. and Merriam, T. (1993). Neural computation in stylometry : An application to the works of Shakespeare and Fletcher. *Literary and Linguistic Computing*, 8(4), pp. 203-209.

Mosteller, F. and Wallace, D. L. (1964). *Inference and Disputed Authorship: The Federalist*. Reading, Mass. : Addison Wesley.

Peng, F. Schuurmans, D. and Wang, S. (2004). Augumenting Naive Bayes Text Classifier with Statistical Language Models , Information Retrieval, 7 (3-4), pp. 317 - 345

Schölkopf, B., J. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson (2001). Estimating the support of a high-dimensional distribution. *Neural Computation*, 13, 1443-1471.

Sebastiani, F. (2002). Machine learning in automated text categorization, *ACM Computing Surveys* 34(1), pp. 1-47.

Tax, D.M.J., *One-class classification*. PhD thesis, TU Delft, 2001.

Tweedie, F. J. and R. H. Baayen (1998). How Variable May a Constant Be? Measures of Lexical Richness in Perspective, *Computers and the Humanities*, 32 (1998), 323-352.

Yang, Y. (1999). An evaluation of statistical approaches to text categorization. *Journal of Information Retrieval*, 1 (1-2), pp 67--88.

Yule, G.U. (1938). On Sentence Length as a Statistical Characteristic of Style in Prose with Application to Two Cases of Disputed Authorship, *Biometrika*, 30, 363-390.