Journal of Machine Learning Research 8 (2007) 1261-1276 Submitted 3/06; Revised 10/06; Published 6/07

Measuring Differentiability: Unmasking Pseudonymous Authors

Moshe Koppel KOPPEL@CS.BIU.AC.IL
Jonathan Schler SCHLERJ@CS.BIU.AC.IL
Elisheva Bonchek-Dokow LE7.BONCHEK.DOKOW@GMAIL.COM
Department of Computer Sciences
Bar-Ilan University
Ramat-Gan 52900, Israel

Editor: Michael Collins

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.

Keywords: authorship attribution, one-class learning, unmasking

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 and 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 2002).

In the *authorship verification* problem (Van Halteren 2004), 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 and Yousef 2001; Schölkopf 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 (SVM) (Joachims 1998; De Vel et al., 2002; Diederich et al., 2003).

This general framework has been used to convincingly solve a number of real world authorship attribution problems (e.g., Mosteller and Wallace 1964; Matthews and 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 book X was written by author A without being offered a closed set of alternatives? Let us begin by considering three naïve approaches to the problem. Although none of them will prove satisfactory, each will contribute to our understanding of the problem.

Approach 1: Lining up impostors – One possibility that suggests itself is to assemble a representative collection of works by other authors and to use a two-class learner, such as SVM, to learn a model for A vs. not-A. Then we chunk the mystery work X and run the chunks through the learned model. If the preponderance of chunks of X are classed as A, then X is deemed to have been written by A; otherwise it is deemed to have been written by someone else. This method is straightforward but it suffers from a conceptual flaw. While it is indeed reasonable to conclude that A is not the author if most chunks are attributed to not-A, the converse is not true. Any author who is neither A nor represented in the sample not-A, but who happens to have a style more similar to A than to not-A, will be falsely determined by this method to be A. Despite this flaw, we will see later that this approach can be used to augment other methods.

Approach 2: One-class learning – Another plausible approach would be to try to handle the problem with *no* negative examples at all. The most straightforward way to do this is to apply a one-class learner, such as a one-class support vector machine (Chang and Lin 2001), that finds an optimal boundary that circumscribes all positive examples of A. Then we ascribe X to A if a sufficient number of chunks of X lie inside the boundary. This approach is conceptually sound but we will see that it performs poorly in practice for our problem.

Approach 3: Comparing A directly to X – Another approach that doesn't depend on negative examples is this: learn a model for 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, that is, if we obtain high accuracy in cross-validation, then conclude that A did not write X. If we fail to correctly classify test examples, conclude that A did write X. This method does not work well at all. But since the method that we introduce in this paper is based on this method, it is worth our while to pause here and consider exactly why the method fails.

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).

The main point of this paper is to show how this problem can be overcome by determining not only if A is distinguishable from X but also how great is the depth of difference between A and X. To do this we introduce 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. Thus by taking into account the depth of difference between two works, we can determine if they were authored by the same person or two different people.

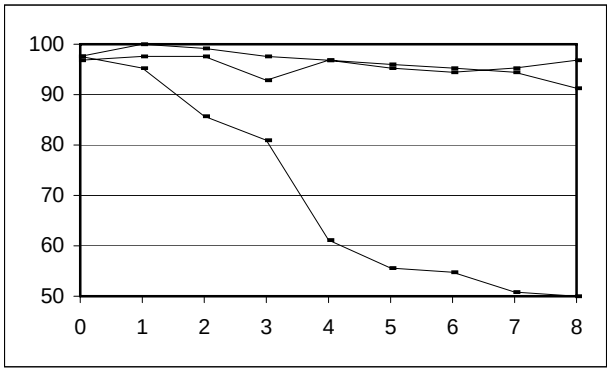

Figure 1. Ten-fold cross-validation accuracy of models distinguishing *House of Seven Gables* from each of Hawthorne, Melville and Cooper. The *x*-axis represents the number of iterations of eliminating best features at previous iteration. The curve well below the others is that of Hawthorne, the actual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

5 Experimental Results

In this section we consider an experimental framework for systematic testing of the effectiveness of the unmasking algorithm

5.1 Corpus

We consider a collection of 21 nineteenth century English books written by 10 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).

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

Table 1 The list of books used in our experiments.

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 consist 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.

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 and Lin 2001) on the 250 most frequent words in A_X to build a model for A_X.

(Although, as discussed above, many more interesting feature sets are possible, we use this feature set here for simplicity and universal applicability.) 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 classified, 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.

A second possible baseline is the "lining up impostors" method mentioned in Section 3 above. We will discuss this method in some detail in Section 6.

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 *n* 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. (If one of the sets, A_X or X, includes more chunks than the other, we randomly discard the surplus. Accuracy results are the average of five runs of ten-fold cross-validation in which we discard randomly for each run.)
2. For the model obtained in each fold, eliminate the *k* most strongly weighted positive features and the k most strongly weighted negative features.
3. Go to step 1.

In this way, we construct degradation curves for each pair $<A_X,X>$. In Figure 2, we show such curves (using *n*=250 and *k*=3) for *An Ideal Husband* against each of ten authors, including Oscar Wilde.

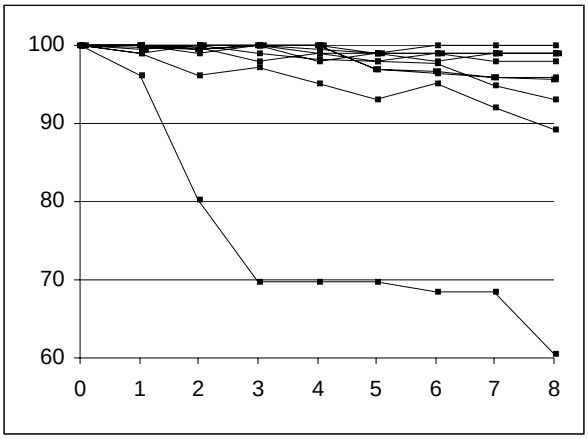

Figure 2. Unmasking *An Ideal Husband* against each of the ten authors (n=250, k=3). The curve below all the authors is that of Oscar Wilde, the actual author. (Several curves are indistinguishable.)

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 $i = 0,\ldots,m$:

- 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. (Note that although we have 20 *same-author* pairs, we really only have 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 *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 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 $\langle A_X,X\rangle$ (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 $\langle A_B,B\rangle$ 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. (It should be noted that some of the 8 misclassified *different-author* pairs result in a single book being attributed to two authors, which is obviously impossible. Nevertheless, since

each of our author/book pairs is regarded as an independent experiment, we do not leverage this information.)

Note that the algorithm includes three parameters: *n*, the size of the initial feature set; *k*, the number of eliminated features from each extreme in each iteration; *m*, the number of iterations we consider. The results reported above are based on experiments using *n*=250, *k*=3, and *m*=10, the settings used in initial experiments first reported in Koppel and Schler (2004). We chose n=250 because experimentation indicated that this was a reasonable rough boundary between common words and words tightly tied to a particular work. Nevertheless, it might be asked how robust our results are vis-à-vis these parameters. To test this, we ran our leave-one-book-out experiment for a variety of parameter settings. The results are shown in Table 2.

Features (n)	Features eliminated (k)	Iterations (m)	Correctly classified *same-author* (out of 20)	Correctly classified *different-author* (out of 189)	F1 (*macro average*)
250	3	5	16	183	0.868
		10	19	181	0.892
		20	20	180	0.896
	6	5	20	182	0.916
		10	20	180	0.896
		20	20	181	0.906
	10	5	20	180	0.896
		10	20	179	0.886
500	3	5	14	189	0.904
		10	12	186	0.828
		20	16	180	0.838
	6	5	13	184	0.826
		10	18	180	0.868
		20	19	179	0.873
	10	5	16	181	0.848
		10	18	180	0.868
		20	20	177	0.868
1000	3	5	11	189	0.843
		10	11	188	0.831
		20	12	183	0.797
	6	5	12	188	0.852
		10	14	184	0.844
		20	17	181	0.863
	10	5	15	184	0.862
		10	16	182	0.857
		20	16	177	0.812

Table 2 Accuracy results on the 21 book experiment for a variety of parameter setting

As is evident, results are somewhat robust with regard to choice of *k* and *m* (in fact, some parameter choices turn out to be better than our initial choice), but the recall results for *same-author* degrades considerably as the size of the initial feature set increases. Apparently, what is important is that at some stage a sufficiently small feature set is reached. A related pattern that is evident in the data is that as the maximum number of features eliminated (i.e., k*m) increases, the total number of example pairs classified as *same-author* increases. (This is reflected by increasing *same-author* recall and decreasing *different-author* recall.) This is because it is the behavior of significantly stripped-down feature sets that permits meta-learning to effectively distinguish between *same-author* curves and *different-author* curves. In the absence of such information, support vector machines tend to err in the direction of majority class, which in this case is *different-author*.

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.

Let us recall the "lining up impostors" method suggested in Section 3 above. Suppose, 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, as we noted above, if it turns out that many, or even all, examples of X are assigned to Melville, we would be hard-pressed to conclude that Melville wrote the book since it may very well be that the works of other authors, say Shaw or Bronte, happen to be more similar to Melville than to Hawthorne or Cooper.

Nevertheless, it is instructive to try this method. Formally, we proceed as follows. For each author A, choose other authors of the same type ("impostors") – let's call them A1,…,An – 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 Ai against not-Ai. 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 Ai(X) analogously. Then if A(X) is not greater than Ai(X) for all i, conclude that A is not the author of X. Otherwise conclude that A is the author of X.

Applying this impostors method to our 209 book-author experiments, we find that 19 of 20 *same-author* pairs are correctly classified but only125 of 189 *different-author* pairs are correctly classified. If we try to remedy the tendency of the method to over-assign to the class *same-author* by adding the constraint that X not be assigned to A unless A(X) exceeds some threshold, θ, we at best obtain only very slightly improved performance. For example, for θ=1/2, 19 of 20 *same-author* pairs and 127 of 189 *different-author* pairs are correctly classified. For θ=.8, 13 of 20 *same-author* pairs and 140 of 189 *different-author* pairs are correctly classified.

In short, the basic impostors method often wrongly concludes that a given author wrote a given book, but when it concludes that a given author did *not* write a given book (because some impostor looks more plausible), it is almost always correct. Thus, although the impostors method is obviously not as effective as unmasking as a stand-alone method, it can be used to augment unmasking. We simply conclude that A is the author of X if and only if both the unmasking

method and the impostors method indicate that A is the author of X. If either of them indicates that A is not the author of X, we conclude that A is not the author of X.

```
Given: anonymous book X, works of suspect author A,
       (optionally)impostors {A1,…,An}

Step 1 – Impostors method(optional)

if impostors {A1,…,An} are given then
{
    Build model M for classifying A vs. all impostors
    Test each chunk of X with built model M
    foreach impostor Ai
    {
         Build model Mi for classifying Ai vs. {A ∪ all other
                                                   impostors}
         Test each chunk of X with built model Mi
     }
     If for some Ai 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 m iterations
      {
            Build a model for A against X
            Evaluate accuracy results
            Add accuracy number to degradation curve <A,X>
            Remove k top contributing features (in each
                                    direction) from data
       }
}
```

Figure 3: Overview of the authorship verification algorithm.

In our experiment, using the augmenting unmasking with the impostors method 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 *different-author* pairs previously misclassified as *same-author* were corrected. Overall, then, the augmented method classed all 189 *different-author* pairs and 18 of 20 *same-author* pairs correctly.

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

7 Effects of Topic Variability on Unmasking

It might well be wondered how robust unmasking is to variability in topic. Specifically, can we successfully identify two works as being by a single author even if they are on different topics and, conversely, can we identify two works as being by two different authors even if they are on the same topic?

An ideal corpus for testing this question is that of rabbinic legal responsa. This corpus of Hebrew-Aramaic letters in response to legal queries is divided by author and typically sub-divided by general topic: ritual law, family law and business law. For our purposes, we chose three prolific authors all of whom worked in the second half of the 20th century. Table 3 shows the number of responsa written by each author in each of three different topic areas.

	Ritual	Business	Family
Author 1 (Yosef)	328	55	143
Author 2 (Feinstein)	157	46	120
Author 3 (Halberstam)	138	70	82

Table 3. Number of responsa written by each author on each topic in legal responsa corpus.

Using the same parameter settings as above (n=250, k=3, m=10), we plotted a variety of unmasking curves. In Figure 4, we show unmasking curves for each author vs. all other authors (solid lines) where all writings are on the same topic, and unmasking curves for each author's writing on a given topic vs. that same author's writing on all other topics (dotted lines).

As is evident, for *different-author* pairs (on a single topic), accuracy remains high even as features are eliminated, while for *same-author* pairs (on different topics), accuracy degrades as features are eliminated. Evidently, among common words, the set of markers of authorial style, regardless of topic, is much larger than the quickly eliminated set of topic markers. In this sense, it is much easier to tell apart one author from another – even when the authors are writing on the same topic – than to tell apart works on different topics written by the same author. As a result, *same-author* curves and *different-author* curves do not resemble each other, despite the potentially confounding effects of topic. In fact, in this case *different-author* curves are distinguishable from *same-author* curves already at the first iteration, before any unmasking is performed. But note that as unmasking is performed, the differences become most clear at the sixth iteration just as was the case for the English literature curves.

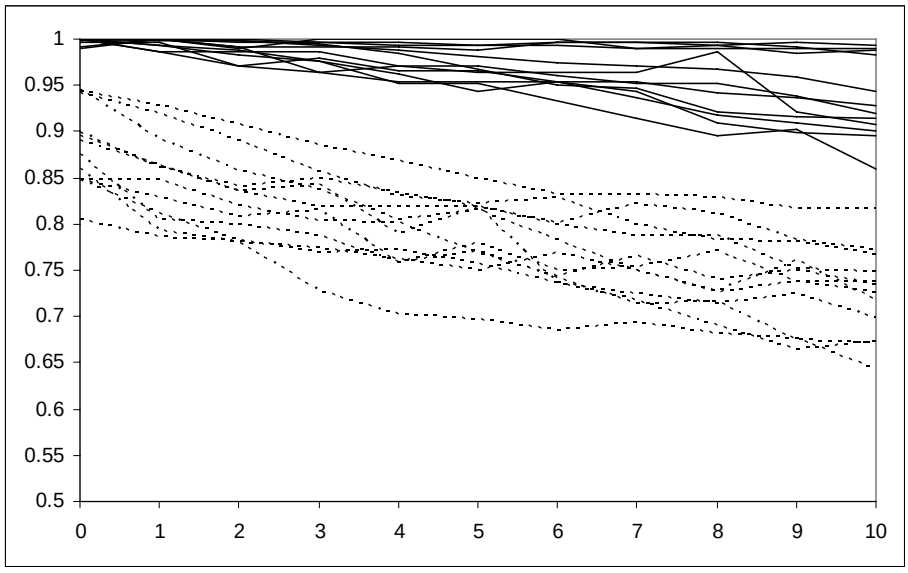

Figure 4 Unmasking of rabbinic legal responsa. Solid lines are *different-author* curves (on same topic) and dotted lines are *same-author* curves (on different topics).

7.1 **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 responsa. 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 (Ben-David, 2002).

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 impostors 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 fact, the strongest distinguishing features – and hence the first eliminated by unmasking – result from Ben Ish Chai employing different standard signoffs in *RP* and *TL*; it is hard to know whether this reflects deliberate subterfuge or mere chronological drift.)

In Figure 5, we show the results of unmasking for *TL* against Ben Ish Chai as well as, for comparison, each of the other four candidate authors. The curve for Ben Ish Chai is the one far below those of the others. This affirms the consensus among historians (Ben-David 2002) that Ben Ish Chai was indeed the author of TL. Indeed, as in our previous experiments, the differences between the curves are most clear at the sixth iteration.

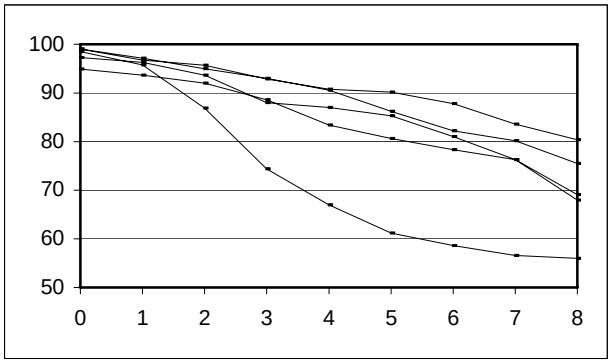

Figure 5: Unmasking *TL* against Ben Ish Chai and four impostors. The curve below the others is Ben Ish Chai.

8 An Alternative Measure of Depth of Difference

It seems plausible that there should be a direct way to determine how different two works are without actually going to the trouble of running the multiple learning experiments required for plotting unmasking curves. One sensible suggestion would be to check the number of features with significant information-gain between authors. Gabrilovich and Markovitch (2004) have used precisely this measure to suggest which learning algorithms might be most appropriate for a given problem.

In Figure 6, we plot the curve in which the *y*-axis represents the number of features with information gain greater than *x*, plotted for each multiple of 0.01 from 0 to 0.6. The ten curves represent the book *An Ideal Husband* vs. the works of each of the ten authors considered in Section 5 above. (Thus, this figure is analogous to Figure 2 above.) The idea is that we expect the curve representing *An Ideal Husband* vs. the other work of Oscar Wilde – the actual author – to show a more sudden drop than those curves comparing *An Ideal Husband* to other authors. And indeed this is the case.

The differences between the *same-author* curve and the *different-author* curves in Figure 6 are not quite as dramatic as the difference in Figure 2 where unmasking curves were used. Nevertheless, the question arises if we can use these curves in much the same way as we use unmasking curves for determining whether two books are by the same author. In order to answer this question, we ran experiments analogous to the leave-one-book-out experiments of Section 5 but with a different feature set. Whereas above we used characteristics of unmasking curves as features in a meta-learning scheme, here we use features of the information-gain curves just considered. More precisely, we record each value shown in Figure 6.

There is no doubt that these curves encode a great deal of information regarding authorship. For example, of the 210 curves generated by the authorship experiment of Section 5, there are 182 IG curves in which there are fewer than 65 features with information gain above 0.03. Of these, 179 are *different-author* curves and only 3 are *same-author* curves. Unfortunately, however, using the meta-learning approach described in Section 5 and a leave-one-out testing protocol results in correct classification of 182 out of 189 *different-author* curves, but only 11 out of 20 *same-author* curves. These results are not quite as good as those obtained using unmasking but they do suggest that information-gain curves are somewhat useful despite their simplicity.

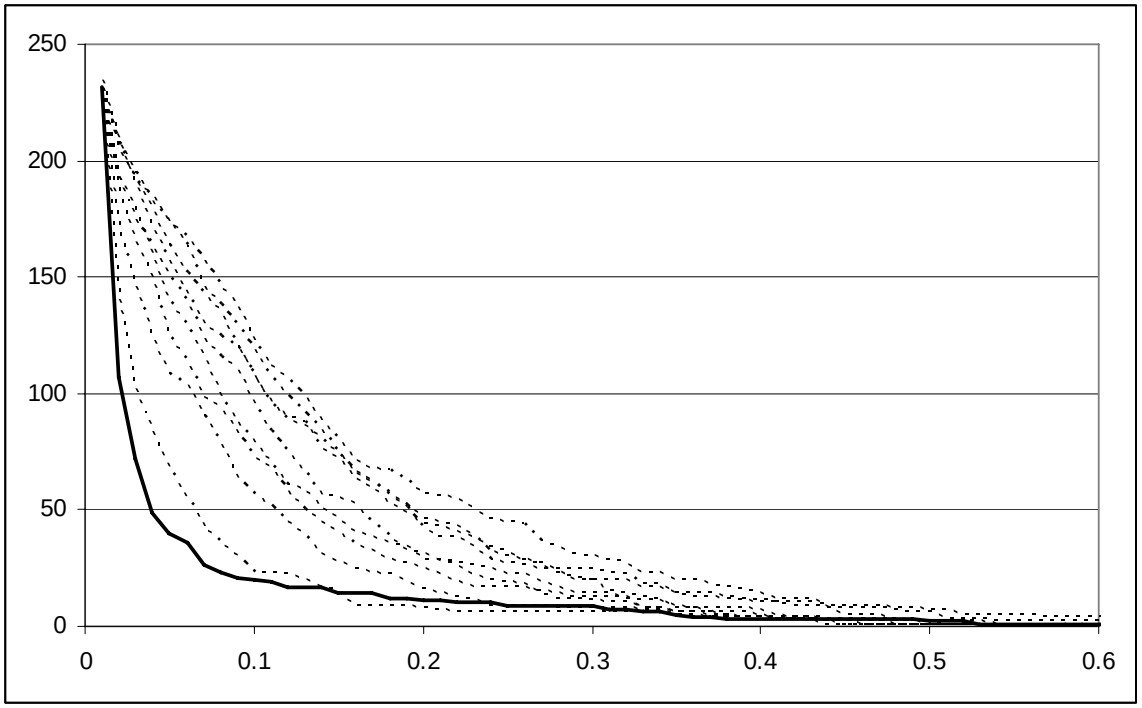

Figure 6. Information-gain curves for *An Ideal Husband* versus ten authors. The dark line is Oscar Wilde, the actual author.

9 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 evidence presented suggests that similar results are obtained as these parameters are varied. In fact, some evidence presented suggests that the method is immune to deliberate attempts to cover up authorship.

The point of the unmasking method suggested here is to measure of the true "depth of difference" between two example sets. This measure is clearly of a different type than other measures, such as margin width, that could in principle depend on a single highly differentiating feature. Although we have tested the method on a single application, it is not unreasonable to speculate that the new measure presented here ought to be applicable to other applications in which we need to determine whether given phenomena were generated by a single process.

References

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

Y.L. Ben-David, (2002), Shevet mi-Yehudah (in Hebrew), Jerusalem (no publisher listed)

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

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

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

O. De Vel, 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.

J. Diederich, J. Kindermann, E. Leopold and G. Paass (2003), Authorship attribution with support vector machines, *Applied Intelligence 19(1)*, 109-123

E. Gabrilovich and S. Markovitch (2004), Text categorization with many redundant features: using aggressive feature selection to make SVMs competitive with C4.5. In *Proc. 21st International Conference on Machine Learning (ICML) 2004,* pages. 321-328.

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

D. Holmes, L. Gordon, and C. Wilson (2001), A widow and her soldier: Stylometry and the American civil war, *Literary and Linguistic Computing 16(4)*, 403-420

T. Joachims, (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

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

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*, pages 69-72.

M. Koppel and J. Schler (2004), Authorship verification as a one-class classification problem, in *Proceedings of 21st International Conference on Machine Learning*, July 2004, Banff, Canada, pages 489-495.

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

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

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

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

F. Peng, D. Schuurmans and S. Wang (2004). Augmenting naive Bayes text classifier with statistical language models , *Information Retrieval,* 7 (3-4), 317 - 345

B. Schölkopf, 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.

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

E. Stamatatos, N. Fakotakis, and G. Kokkinakis (2001) Computer-based authorship attribution without lexical measures. *Computers and the Humanities,* 35(2), 93-214, Kluwer, 2001.

D.M.J. Tax (2001), One-Class Classification. *PhD thesis,* TU Delft, 2001.

F. J Tweedie 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.

H. Van Halteren (2004) Linguistic profiling for authorship recognition and verification, *Proc. of 42nd Conf. Of ACL*, July 2004, 199-206

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

G.U. Yule (1938). On sentence length as a statistical characteristic of style in prose with application to two cases of disputed authorship, *Biometrika*, 30, 363-390.