Determining If Two Documents Are Written by the Same Author

Article *in* Journal of the Association for Information Science and Technology · January 2014

DOI: 10.1002/asi.22954

Determining If Two Documents Are Written by the Same Author

Moshe Koppel and Yaron Winter
Department of Computer Science, Bar-Ilan University, Ramat-Gan, Israel 52900.
E-mail: {moishk, yaron.winter}@gmail.com

Almost any conceivable authorship attribution problem can be reduced to one fundamental problem: whether a pair of (possibly short) documents were written by the same author. In this article, we offer an (almost) unsupervised method for solving this problem with surprisingly high accuracy. The main idea is to use repeated feature subsampling methods to determine if one document of the pair allows us to select the other from among a background set of "impostors" in a sufficiently robust manner.

Introduction

The Internet is replete with documents that are written pseudonymously or anonymously and it is often of considerable financial or legal importance to determine if two such documents were in fact written by a single author. For example, one may want to know if several tendentious product reviews were written by the same writer or, portentously, if two threatening letters were written by the same individual. In this article, we propose a solution to the authorship verification problem: determining whether two documents were written by the same author. Importantly, we consider cases in which the two input documents are not necessarily long.

Note that authorship verification is an open-set problem: we ask if an anonymous document was written by a given candidate author or a different individual. It is not hard to see that virtually all standard closed-set authorship attribution problems are reducible to the authorship verification problem, whereas the reverse is not true. In the standard case, we are faced with a closed set of candidate authors for each of whom we have writing samples and are asked to determine which of them is the actual author of an anonymous text. Plainly, if we can determine if any two documents are written by the same author, we can solve any such standard authorship attribution problem, regardless of the number of candidates. All we need to do is ask if the anonymous text was written by each of the respective candidates; we will get a positive answer for the true author and a negative answer for all the others. On the other hand, the verification problem is strictly harder than the attribution problem: the fact that we solve a closed-set attribution problem offers no guarantees that we can solve an open-set verification problem. It is thus not surprising that, with a single limited exception (see below), no satisfactory solution has previously been offered for the verification problem.

The outline of our solution is as follows: Suppose we are asked to determine if the documents *X* and *Y* were written by the same author. We systematically produce a set of "impostor" documents and—in a matter reminiscent of a police lineup—ask if *X* is sufficiently more similar to *Y* than to any of the generated impostors. The trick is using the proper methods to select the impostors and, more important, to measure document similarity. Our measurement of document similarity involves randomly selecting subsets of features that serve as the basis for comparing documents, as explained below. We see that when executed correctly, this method gives surprisingly strong results for the verification problem, even when the documents in question contain no more than 500 words.

In the following section we briefly review previous related work. In the Experimental Setup section we describe and offer two simplistic baseline methods. In The Many-Candidates Problem section we outline its solution. In The Impostors Method section we offer a method for reducing the authorship verification problem to the many-candidates problem, and in the final section we offer the results of this work.

Related Work

There has been limited research on the open-set authorship verification problem. Koppel and Schler (2004) introduced the "unmasking" method in which the two input documents are chunked and the effectiveness of machine learning methods at distinguishing them is measured via cross-validation on the chunks. Because chunks of text must

Received November 26, 2012; revised February 17, 2013; accepted February 18, 2013

 DOI: 10.1002/asi.22954

be reasonably long (at least a few hundred words) to gain any kind of statistical representativeness, unmasking requires that the input documents be very long. In fact, empirical studies (Sanderson & Guenter, 2006) have shown that unmasking is ineffective for short input documents (less than 10,000 words).

Novak, Raghavan, and Tomkins (2004) considered the case in which the respective writings of 100 authors were each split into two and then needed to be properly matched. They found that using certain feature sets to represent the texts and clustering into 100 pairs yields very strong results. However, in their formulation of the problem it is known in advance that each document has some match in the set. Furthermore, all 100 matching problems are solved dependently so that each yields information about the other. Thus, that version of the problem is considerably easier than the one we wish to solve here. Similar work includes that of Juola and Baayen (2005) and Abbasi and Chen (2008) on what they call the "similarity problem."

There has also been some work on intrinsic plagiarism detection (Meyer zu Eissen, Stein, & Kulig, 2007) that is similar, although not identical, to the authorship verification problem.

In this work, we use large sets of impostor candidates. Previous work on large candidate sets for authorship attribution is somewhat limited. Madigan et al. (2005) considered 114 authors, Luyckx and Daelemans (2008) considered 145 authors, and Koppel, Schler, and Argamon (2011) considered thousands of authors. Most recently, Narayanan et al. (2012) considered as many as 100,000 authors. A few words about authorship attribution with large sets are in order. The standard authorship attribution in which we need to assign an anonymous document to one of a small closed set of candidates is well understood and has been summarized in several surveys (Juola, 2008; Stamatatos, 2009). As a rule, automated techniques for authorship attribution can be divided into two main types. In machine-learning methods, the known writings of each candidate author (considered as a set of distinct training documents) are used to construct a classifier that can then be used to classify anonymous documents (Abbasi & Chen, 2008; Koppel, Schler, & Argamon, 2008; Zhao & Zobel, 2005; Zheng, Li, Chen, & Huang, 2006). In similarity-based methods, a metric is used to measure the distance between two documents and an anonymous document is attributed to that author to whose known writing (considered collectively as a single document) it is most similar (Abbasi & Chen, 2008; Argamon, 2007; Brennan & Greenstadt, 2009; Burrows, 2002; Hoover, 2003; Malyutov, 2006; Uzuner & Katz, 2006). When there are tens, or possibly even hundreds or thousands, of candidate authors, standard machine-learning methods—designed for small numbers of classes—are not easily usable. (In principle, one could use machine-learning methods to find a separate binary classifier for each candidate author, but this is both unwieldy and would in any case require some method for choosing from among multiple positive answers.) In such cases, similarity-based methods are more natural than machine-learning methods (Stamatatos, 2009). We use similarity-based methods in this article.

Experimental Setup

We use a corpus consisting of the full output of several thousand bloggers taken from blogger.com. The average blogger in our corpus has written 38 separate blog posts over a period of several years. We consider pairs of (fragments of) blog posts, $\langle X, Y \rangle$, where X consists of the first 500 words produced by a given blogger and Y consists of the last 500 words (on the date we downloaded) produced by a given blogger (who might or might not be the same blogger). We chose the first and last words of bloggers to maximize the time gap between the documents we wish to compare; in fact, for the cases in which X and Y are taken from the same blogger, it is never the case that X and Y belong to the same blog post. We chose 500 words per blog to show that our methods are effective even for a relatively short document; later we consider texts of different lengths, both greater and less than 500 words.

We randomly generate a corpus that includes 500 such pairs; for half of them, X and Y are by the same blogger and for the other half they are not. (No single blogger appears in more than one pair $\langle X, Y \rangle$.) The task is to correctly identify a given pair as *same-author* (i.e., X and Y are by a single blogger) or *different-author* (i.e., X and Y are by two different bloggers).

Note that our problem is unsupervised in the sense that we are not supplied with labeled examples of any of the authors in the corpus.

Similarity-Based Baseline Method

We first consider two rather simplistic baseline methods for approaching the problem. Given the pair of documents $\langle X, Y \rangle$, the first method is to measure the similarity between X and Y and assign the pair to the class same-author if the similarity exceeds some threshold. This is essentially the method used by Abbasi and Chen (2008) for what they call "similarity detection," although the similarity measures they use are based on features considerably more sophisticated than ours.

To measure the similarity between the documents X and Y, we first represent each document as a numerical vector containing the respective frequencies of each space-free character 4-gram in the document. For our purposes, a space-free character 4-gram is (a) a string of characters of length four that includes no spaces or (b) a string of four or fewer characters surrounded by spaces. We select the 100,000 such features most frequently found in the corpus as our feature universe. Character n-grams have long been known to be effective for authorship attribution (Houvardas & Stamatatos, 2006; Keselj, Peng, Cercone, & Thomas, 2003) and have the advantage of being measurable in any language without specialized background knowledge. Although other feature

DOI: 10.1002/asi

sets, such as bag-of-words, function words, and parts-of-speech n-grams, are reasonable choices, recent studies (Escalante, Solorio, & Montes, 2011; Grieve, 2007; Luyckx & Daelemans, 2010; Plackias & Stamatatos, 2008) suggest that simpler feature sets such as character n-grams are at least as effective as the alternatives and often even more effective; our preliminary experiments confirmed this finding. In our case, character n-grams have an additional advantage: Our method works most naturally with a very large and homogeneous feature set, precisely what is offered by character n-grams. The use of 4-grams, specifically, was found to be effective in experiments reported in Winter (2012).

Let $\vec{X} = <x_1, \ldots, x_n>$ and $\vec{Y} = <y_1, \ldots, y_n>$ be the respective vector representations of the documents X and Y, where each x_i represents the *tf*idf* value of a character 4-gram in X and n is the total number of such 4-grams that we consider. We use two standard vector similarity measures, the cosine measure and the min-max measure:

$$\mathrm{sim}(X, Y) = \mathrm{cosine}(\vec{X}, \vec{Y}) = \vec{X} * \vec{Y} / \|\vec{X}\| * \|\vec{Y}\|$$

$$\mathrm{sim}(X, Y) = \mathrm{minmax}(\vec{X}, \vec{Y}) = \frac{\sum_{i=1}^{n} \min(x_i, y_i)}{\sum_{i=1}^{n} \max(x_i, y_i)}$$

This baseline method ignores the fact that the similarity of two documents is determined by many factors—genre, topic, and so on—other than author identity; a single uniform threshold for all pairs is not likely to work especially well. In fact, using cosine similarity, the threshold that maximizes accuracy on the development set yields accuracy of 70.6% on the test set. For the min-max similarity measure, the threshold that maximizes accuracy on the development set yields accuracy of 74.2% on the test set. We return to these numbers in the Results section.

Supervised Baseline Method

We now consider a second baseline method that makes use of a training set. Suppose that we have a training set consisting of 1,000 pairs $\langle X, Y\rangle$, each of which is labeled as a different-author pair or a same-author pair. (The set of authors that appear in this training set are disjoint from those that appear in our corpus.) We use supervised methods to learn to distinguish between same-author pairs and different-author pairs, as follows: Represent X and Y as vectors, as described previously. For a pair $\langle X, Y\rangle$, define $\mathrm{diff}(X, Y) = \langle |x_1 - y_1|, \ldots, |x_n - y_n|\rangle$. For each pair $\langle X, Y\rangle$ in the training set, assign the vector diff(X, Y) the label *same-author* if $\langle X, Y\rangle$ is a same-author pair and assign the vector diff(X, Y) the label *different-author* if $\langle X, Y\rangle$ is a different-author pair. We now use these labeled examples as training examples for supervised learning and apply the learned classifier to our test set. (Note that our classifier learns nothing about specific authors but, rather, about what differences in n-gram frequency are characteristic of same-author pairs in general.) Because this is a binary learning problem and support vector machine (SVM) has often been found to perform well for binary authorship problems (Abbasi & Chen, 2008; Zheng et al., 2006), we chose SVM as our learning algorithm.

Learning a linear SVM classifier on the development set, exactly as described, we obtain an accuracy of 79.8% on our test set. (This is the strongest result we obtained using a variety of kernels and parameter setting and various feature sets, including bag-of-words, function words, and others; thus, this is the most competitive version of the baseline method against which we compare our algorithm.) We see below that although our method is almost unsupervised, it performs better than this supervised method.

The Many-Candidates Problem

We now consider a new approach to the verification problem. Our approach is based on the solution to a closely related problem: Given a large set of candidate authors, determine which, if any, of them is the author of a given anonymous document. We call this problem the *many-candidates problem*; it is sometimes called the *open-set identification problem*. If we can solve the many-candidates problem, we can convert the verification problem into the many-candidates problem by generating a large set of impostor candidates. (Technically speaking, the open-set identification problem is also reducible to the open-set verification problem, so these two problems are equivalent. As we noted earlier, closed-set identification is reducible to open-set verification; obviously, it is also reducible to open-set identification.)

Next we consider how the many-candidates problem can be effectively approached (Koppel et al., 2011).

In keeping with our experimental setup, suppose that we have a candidate set consisting of 5,000 bloggers for each of whom we have the first 500 words of their blog. Now we are given the last 500 words (which we'll call a *snippet*) of some unspecified blog and are asked to determine which, if any, of the 5,000 candidates is the author of this snippet.

Many-Candidates Method

We start with a somewhat naïve information-retrieval approach to assign an author to a given snippet. Using the feature set and min-max proximity measure defined previously, we assert that the author of the snippet is the blogger in the candidate set whose text is most similar to the snippet vector. (Note that we use min-max rather than cosine as our proximity measure, because it yielded better results in our similarity-based baseline method above.) The number of snippets correctly assigned will depend on the length of the snippets and the number of candidate authors. In Figure 1 we show the accuracy obtained for a variety of snippet lengths and sizes of candidate sets. (Each datapoint represents accuracy obtained for 1,000 test snippets.) Thus, for example, we find that when there are 5,000 author candidates, each consists of 500 words, and 32.5% of the snippets are correctly assigned.

DOI: 10.1002/asi

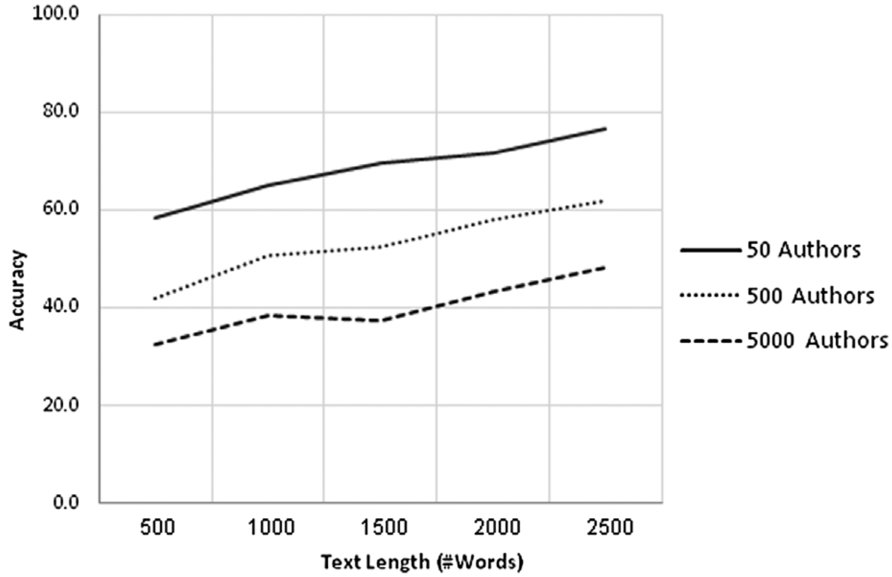

FIG. 1. Accuracy of the naïve information retrieval algorithm for various sized texts and author sets.

We note that although 32.5% is perhaps surprisingly high (because the baseline is 0.02%), it is inadequate for most applications. Moreover, this method necessarily assigns every snippet to some author in the candidate set despite the fact that it may be the case that none of the authors in the candidate set is the actual author. What is required is some criterion by which it can be determined if the best candidate is the actual author of the snippet.

As our earlier baseline results indicate, simply requiring that similarity between the best candidate and the snippet exceeds some threshold will not work. Rather, the crucial idea is to vary the feature sets used in representing the texts. If a particular candidate blogger's known text is more similar to the snippet than any other candidate for many different feature set representations of the texts, then that candidate is very likely the author of the snippet. Another candidate's text might happen to be the most similar for one or a few specific feature sets, but it is highly unlikely to be consistently so over many different feature sets.

This observation suggests using the following algorithm (Koppel et al., 2011):

Given: a snippet to be assigned; known-texts for each of C candidates

1. **Repeat** k times
   a. Randomly choose half of the features in the full feature set.
   b. Find top known-text match to snippet using min-max similarity
2. **For each** candidate author A,
   a. Score(A) = proportion of times A is top match

Output: argmax_A Score(A) **if** max Score(A) > σ*; else Don't Know

The idea is to check if a given author proves to be most similar to the test snippet for many different randomly selected feature sets of fixed size. The number of iterations, k, is a tweakable parameter, but, as we will see shortly, k = 100 is sufficient. The threshold σ* serves as the minimal score an author needs to be deemed the actual author. This parameter can be varied to obtain a tradeoff between recall-precision tradeoff.

This method is similar to classifier ensemble methods in which different classifiers are learned using different subsets of features (Bryll, Gutierrez-Osuna, & Quek, 2003).

Many-Candidates Results

We applied this method to the blogger problem described previously, using 1,000 test snippets for various candidate set sizes: 50, 500, and 5,000. In Figure 2, we show recall-precision curves generated by varying the score threshold σ* (where precision is the proportion of correct attributions among all test snippets for which some attribution is given by the algorithm and recall is the proportion of test snippets for which an attribution is given by the algorithm and is correct). As expected, the results improve as the number of candidate authors diminishes. We mark on each curve the point σ* = 0.80. For example, for 500 candidates, at σ* = 0.80, we achieve 90.2% precision at 22.2% recall. (Koppel et al. [2011] reported similar results for different candidate set sizes and snippet lengths.)

For the above experiments, we used k = 100 iterations. We note that using more iterations does not appreciably change the results. For example, for the case of 500 candidate authors, recall at 90% precision is 22.3% using 100 iterations; using 1,000 iterations it is also 22.3%.

It would be a mistake to conclude, however, that the many-candidates problem is necessarily easier as the number of candidates diminishes. The above result

DOI: 10.1002/asi

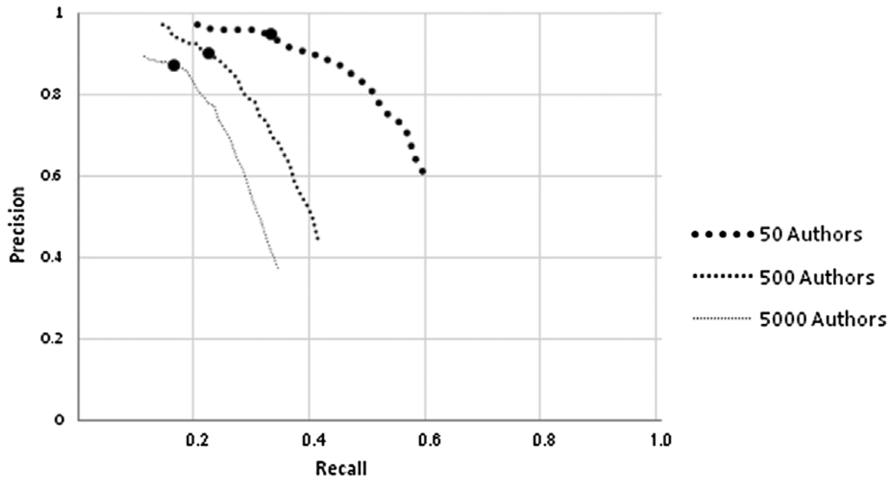

FIG. 2. Recall-precision curves for the many-candidates method for various sized candidate sets.

considered cases in which the actual author of a snippet is among the candidate authors. Consider now the possibility that none of the candidate authors is the actual author of the snippet. What we would hope to find is that in such cases the method does not attribute the snippet to any of the candidates. In fact, testing on 1,000 snippets that belong to none of the candidates, we find that at $\sigma^* = 0.80$, not many are mistakenly attributed to one of the candidate authors: 3.7% for 5,000 candidates, 5.5% for 500, and 8.4% for 50. Perhaps counterintuitively, for snippets by authors not among the candidates, having fewer candidates actually makes the problem *more* difficult since the fewer competing candidates there are, the more likely it is that there is some consistently most similar (but inevitably wrong) candidate. (To take an extreme case, when there are only two candidates, neither of whom is the author, it is plausible that one of them is more similar to the snippet than the other for the preponderance of feature sets; for 1,000 candidates, it is unlikely that one of them is consistently more similar than all the others.)

Thus, there is a tradeoff between cases with many candidates (in which case there might be many false negatives) and cases with few candidates (in which case there might be many false positives). It is important to bear this tradeoff in mind in what follows.

The Impostors Method

We return now to the verification problem. We are given a pair of documents $\langle X, Y\rangle$ and need to determine if they are by the same author. Because we have seen that we have a reasonably effective solution to the many-candidates problem, we can use impostors to reduce the verification problem to the many-candidates problem. The use of impostors as a background set is a well-established practice in the speaker-identification community (e.g., Reynolds, 1995) and has also been applied to information-retrieval problems (Zelikovitz, Cohen, & Hirsh, 2007), but, as far as we know, has not been previously used for authorship attribution.

We proceed as follows:

1. Generate a set of impostors $Y_1, \ldots, Y_m$ (as specified below).
2. Compute $score_X(Y)$ = the number of choices of feature sets (out of 100) for which $sim(X, Y) > sim(X, Y_i)$, for all $i = 1, \ldots, m$.
3. Repeat the above with impostors $X_1, \ldots, X_m$ and compute $score_Y(X)$ in an analogous manner.
4. If average($score_X(Y)$, $score_Y(X)$) is greater than a threshold σ^*, assign $\langle X, Y\rangle$ to *same-author*.

The crucial issues that need to be dealt with are how to choose the impostor set and how many impostors to use. Intuitively, if we choose too few impostors or we choose impostors that are unconvincing—to take an extreme example, imagine X and Y are in English and the impostors are all in Turkish—we will get many false positives. Conversely, if we choose too many impostors or we choose impostors for Y that are in the same genre as X but not in the same genre as Y, we will get many false negatives.

In short, we seek an optimal combination of impostor quality, impostor quantity, and score threshold. The three are interrelated. To develop some intuition for this, consider the following three methods of generating a universe of potential impostors for Y:

- **Fixed**: Use a fixed set of impostor documents having no special relation to the document pair in question. For this purpose, we used the aggregate results of random (English) Google queries.
- **On-the-fly**: Choose a variety of small sets of random (medium-frequency) words from Y and use each such set as a Google query; aggregate the top results of the respective queries. This is a set of topically "plausible" impostors for Y. (Of course, the identical procedure is used to generate

DOI: 10.1002/asi

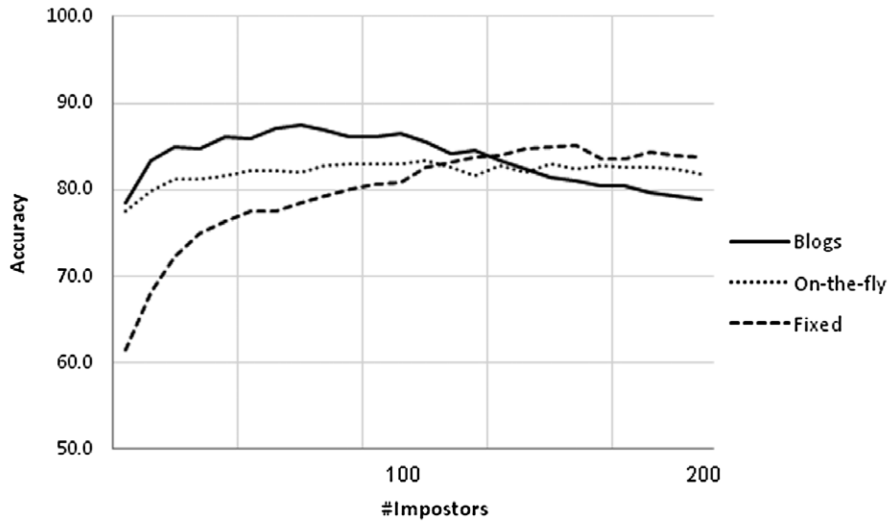

FIG. 3. Accuracy of the impostors method ($\sigma^* = 0.1$) for three impostor universes as the number of impostors increases.

impostors for X.) The motivation for this method is that it can be applied to any pair of documents on-the-fly with no prior knowledge regarding their provenance. (For our purposes, we use 50 sets of 3 to 5 random words and take the top 25 results for each; because we are satisfied with any plausible impostor set, we make no attempt to optimize these values.)

- **Blogs**: Choose texts from other bloggers. This is a set of impostors that are at least in the same genre as both X and Y. Here we assume that we at least know the shared genre of the pair of documents.

In each of these universes, all documents are 500 words in length exactly. To illustrate one key point, we show in Figure 3 the accuracy obtained using the impostors method with varying numbers of impostors drawn respectively from each of the above three universes. Results are shown for the score threshold $\sigma^* = 0.10$, but the phenomenon we wish to point out is evident for other threshold values as well.

We find that when we choose impostors that are more similar to Y either in terms of genre (Blogs) or content (On-the-fly), fewer impostors are required to achieve the same or better accuracy than just choosing random impostors. For all impostor universes, the greater the number of impostors, the more false negatives and the fewer false positives.

In our experiments, we consider the Blog universe and the On-the-fly universe. In each case, we use reasonably good impostors (to avoid false positives), although not necessarily the very best impostors (to avoid false negatives). We thus use the following protocol for generating impostors (for both On-the-fly and Blogs):

1. Compute the min-max similarity to Y of each document in the universe. Select the m most similar documents as potential impostors.
2. Randomly select n actual impostors from among the potential impostors.

We see below that results are not particularly sensitive to the choice of m and n.

Results

Blog Corpus Results

For our first experiment, we are given 500 blog pairs as described previously and we need to assign each of them to the class *same-author* or to the class *different-author*. We apply five methods, three baseline methods as well as our impostors method with each of two universes. For each method, we use a parameter to tradeoff recall and precision. Briefly, the five methods and their respective parameters are as follows:

1. Thresholding on similarity using cosine.
2. Thresholding on similarity using min-max.
3. Classifying according to an SVM classifier learned on the training set; signed distance from the boundary is the parameter.
4. The impostors method using the On-the-fly universe; the score threshold is the parameter.
5. The impostors method using Blog universe; the score threshold is the parameter.

Figure 4a shows recall and precision for all methods for the class *same-author* and Figure 4b shows recall-precision curves for the class *different-author*. As can be seen, the impostors method is quite a bit better than the baseline methods, including the supervised method. Also, the Blog universe gives better results than the On-the-fly universe for the impostors method.

Note that for the impostors method using the blog universe, recall at precision = 0.9 is 82.5% for the class *same-author* and 66.0% for the class *different-author*. The score threshold σ^* for which we obtain precision of 0.9 for same-author is 0.13; for diff-author we obtain precision = 0.9 with

DOI: 10.1002/asi

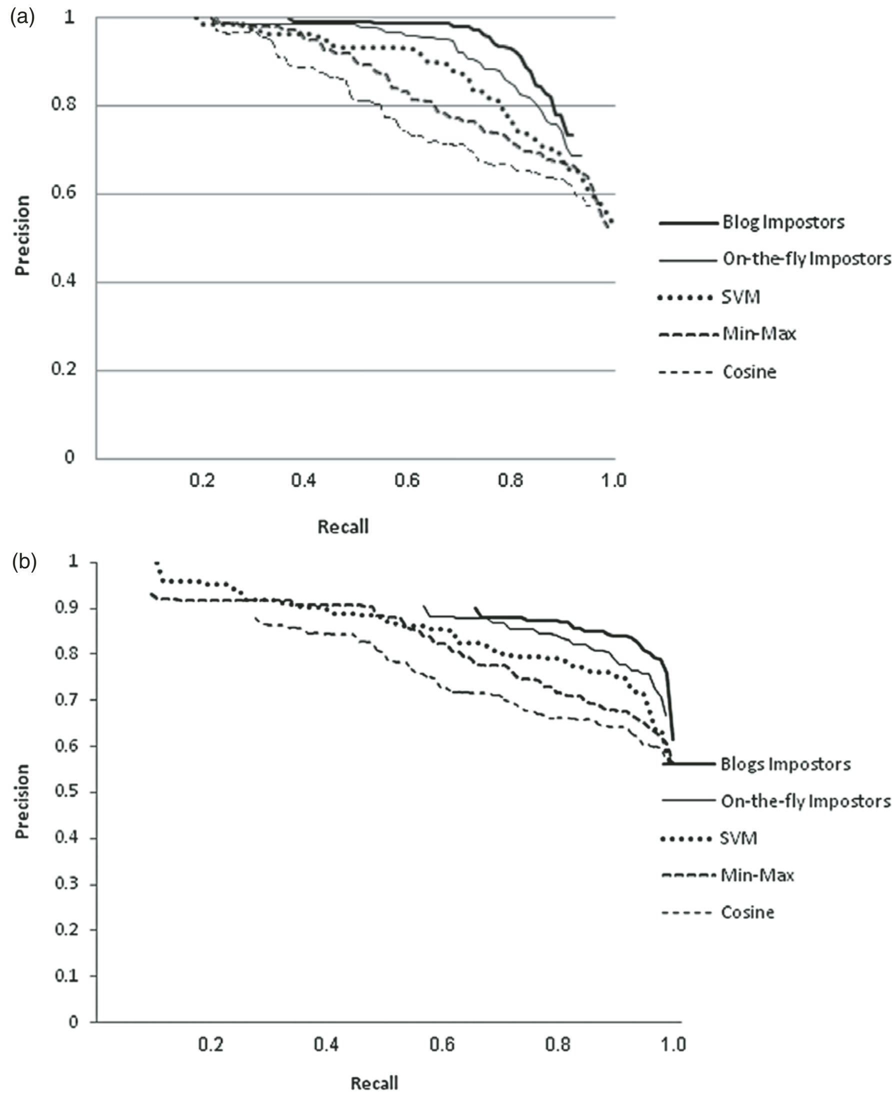

FIG. 4. (a) Recall-precision curves for the class *same-author* for each of five verification methods. (b) Recall-precision curves for the class *different-author* for each of five verification methods.

score threshold of 0.01. As a practical matter, this means that—assuming a prior probability of 0.5 that X and Y are by the same author—a score above 0.12 indicates that the chance that X and Y are by different authors is less than 10% and a score below 0.02 indicates that the chance that X and Y are by the same author is less than 10%.

In Figure 5, we show the accuracy obtained at the optimal choice of parameter value for each of our five methods. (For the impostors method, the optimal parameter values are determined on a separate development set that is constructed using the exact same methodology used to construct our test corpus, but on a disjoint set of bloggers. In this sense, our method is technically not completely unsupervised.) We obtain 87.4% accuracy for the impostors method using the Blogs universe. The fact that we obtain 83.2% for the impostors method using the On-the-fly universe is especially encouraging; it means that this method can be successfully

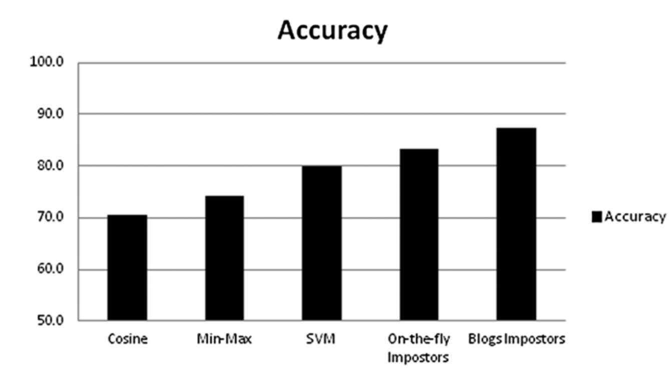

FIG. 5. Optimal accuracy (over all values of σ^*) for each of five verification methods on the Blogs corpus.

DOI: 10.1002/asi

TABLE 1. Optimal accuracy for various impostor selection configurations using Blog impostors.

Selected(n) \ Potential(m)	100	250	500	1,000
10	85.6%	87.7%	87.3%	87.2%
25	85.2%	87.4%	87.7%	87.4%
50	85.7%	87.6%	87.7%	87.5%
100	82.8%	86.9%	87.1%	86.8%

TABLE 2. Optimal accuracy for various impostor selection configurations using On-the-fly impostors.

Selected(n) \ Potential(m)	100	250	500	1,000
10	81.5%	82.5%	82.4%	82.2%
25	83.1%	83.2%	83.2%	83.0%
50	82.1%	83.1%	83.2%	82.4%
100	80.8%	83.1%	82.4%	82.1%

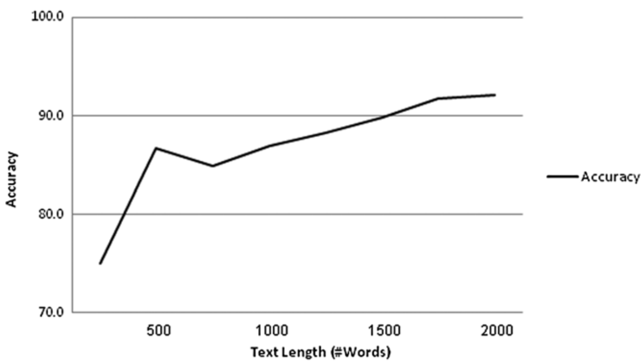

FIG. 6. Accuracy as a function of text length for the impostor method with Blog impostors.

applied even in cases where we know nothing at all about the pair of documents, not even their genre.

For simplicity, we have shown results for the impostors method using a particular choice of values for the number of potential impostors ($m = 250$) and the number of randomly chosen impostors ($n = 25$). However, as we can see in Tables 1 (Blog impostors) and 2 (On-the-fly impostors), results for both impostor sets are hardly sensitive to the choices of these parameters as long as each is sufficiently large, although it seems that it is better to randomly choose actual impostors from among the top impostors than to use all the top impostors.

Note that all our results are for pairs of documents that are of length 500. If we have longer documents, the results are even stronger. In fact, as can be seen in Figure 6, the accuracy of the impostors method increases as the length of the input documents increases. For documents of length 1,500 or greater the accuracy exceeds 90%.

Our results thus far relate to a test corpus in which the same-author pairs and different-author-pairs are evenly

TABLE 3. Macro-averaged F1 for various priors for same-author.

Prior	Macro F1	Score threshold
0.1	86.9	0.30
0.2	86.7	0.22
0.3	87.5	0.19
0.4	87.5	0.18
0.5	87.4	0.14
0.6	86.0	0.14
0.7	83.7	0.05
0.8	81.0	0.05
0.9	75.8	0.01

distributed. In the real world, however, it is typically the case that prior probability that a pair X and Y are by the same author may be very far from 0.5 (in either direction). We therefore show in Table 3 the macro-averaged F1 value obtained for various prior probabilities that X and Y are indeed by the same author. In each case, the score threshold (indicated in the third column of the table) is the one that optimizes macro-averaged F1 (in a separate development corpus) for that particular distribution of same-author and different-author pairs. As the prior probability of same-author diminishes, the score we require for a pair to be classed as a same-author pair increases. As can be seen, as long as the prior probability of same-author is not too large, macro-averaged F1 results are quite consistent.

Student Essay Results

The similarity of two documents by the same blogger across different feature sets presumably results from a common underlying writing style as well as, at least in some cases, the tendency of an individual blogger to often return to particular favorite topics. How much would the results be weakened if we systematically guarantee that there is no topical stickiness within individual authors?

To investigate this issue, we considered a corpus of student essays[1] in which each of 950 students has contributed four essays: stream of consciousness, reflections on childhood, an assessment of one's own personality, and a thematic apperception test. Our corpus includes 2,000 pairs $\langle X, Y \rangle$. (In each case, we use only the first 500 words of the essay.) The critical point, however, is that in every such pair X is chosen from one of these subgenres and Y is chosen from a different subgenre. Thus, the topic issue (and to a lesser extent, the genre issue) is completely neutralized: neither same-author pairs nor different-author pairs are ever about a single topic or in the same subgenre.

In this case, we chose our impostors for Y from among all those essays that are in the same subgenre as Y. This method of choosing impostors is akin to choosing blog

[1]We thank Jamie Pennebaker for making the Students-Essay corpus available to us.

DOI: 10.1002/asi

impostors in the previous problem; we simply leverage what we know about the common nature of a given document pair. In this case, we know only that both are student essays, so we use student essays as impostors. We also know the subgenre of *Y*, so we use impostors from its subgenre.

In Figure 7, we show the accuracy obtained at the optimal choice of parameter value (determined on a separate development set) for each of our four methods on the Student Essay corpus. (There are four methods rather than five in this case because there is only one impostors universe.) We obtain 73.1% accuracy for the impostors method using genre-based-impostors. This result is better than any of the baseline methods, but it is still considerably weaker than the results obtained for the blog corpus. This reflects the fact that same-author pairs in this corpus differ by design both in terms of subgenre and topic, leaving many fewer common features to exploit. In Figure 8 we show recall-precision curves for each of the four methods. The relative strength of the impostors method is evident.

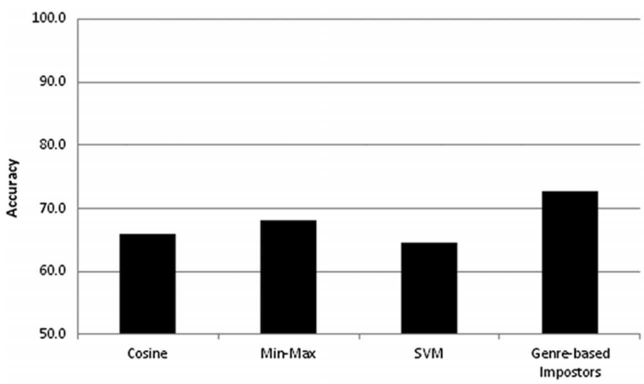

FIG. 7. Optimal accuracy (over all values of σ*) for each of the four verification methods on the Student Essay corpus.

Conclusions

In this article, we have considered one of the most fundamental and difficult authorship problems—determining if a pair of short documents was written by the same author. We have found that this problem can be solved with reasonable accuracy under certain conditions. This result is of considerable practical importance because many real-life problems—for example, authentication of short documents of questionable authenticity—are of this form.

Our approach is almost unsupervised. The method works in two stages. The first stage is to generate a set of impostors that will serve as a background set. We have found that in choosing the impostors one must find the proper balance between the quality of the impostors (i.e., their similarity to the "suspect") and the number of impostors chosen: the more convincing the impostors, the fewer need be used. We have further found that best results are obtained when the impostors are selected from the same genre as the input documents, but that strong results can be obtained even when no information regarding the input documents is available. In such cases, a search engine can be used to generate impostors.

The second stage is to use feature randomization to iteratively measure the similarity between pairs of documents, as proposed in Koppel et al. (2011). If, according to this measure, a suspect is picked out from among the impostor set with sufficient salience, then we claim the suspect as the author of the disputed document.

There are a number of potential limitations of the method that require further investigation. First, as we have seen, when the two documents in question differ in genre and topic, it is considerably harder to distinguish same-author and different-author pairs.

Another potential pitfall is that we need to be fairly certain that our impostor documents were not themselves written by the author(s) of the pair of documents in question. This danger does not seem to have adversely affected the

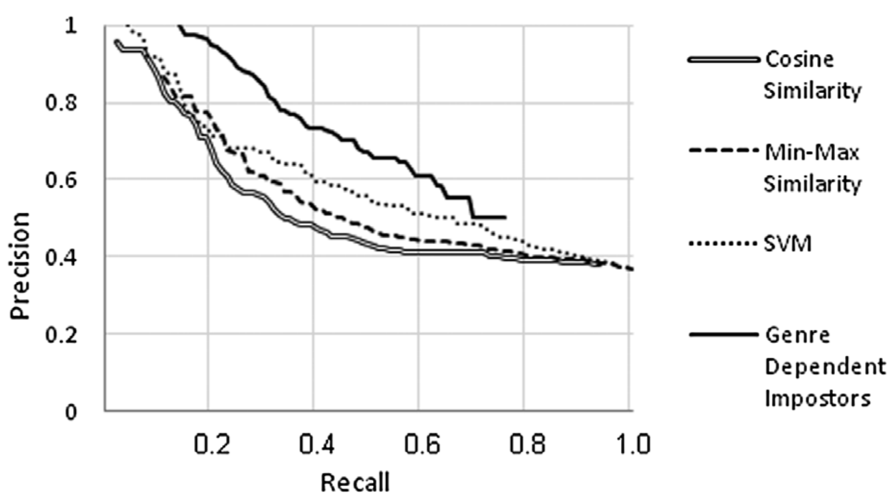

FIG. 8. Recall-precision curves for the class *same-author* for each of the four verification methods on the Student Essay corpus.

DOI: 10.1002/asi

results in the blog corpus, but it is a potential problem that should be taken into account.

References

Abbasi, A., & Chen, H. (2008). Writeprints: A stylometric approach to identity-level identification and similarity detection. ACM Transactions on Information Systems, 26(2).

Argamon, S. (2007). Interpreting Burrows's Delta: Geometric and probabilistic foundations. Literary and Linguistic Computing, 23(2), 131–147.

Brennan, M., & Greenstadt, R. (2009). Practical attacks on authorship recognition techniques. In Proceedings of the Twenty First Conference on Innovative Applications of Artificial Intelligence (IAAI), Pasadena, CA.

Bryll, R., Gutierrez-Osuna, R., & Quek, F.(2003). Attribute bagging: Improving accuracy of classifier ensembles by using random feature subsets. Pattern Recognition, 36(6), 1291–1302.

Burrows, J.F. (2002). Delta: A measure of stylistic difference and a guide to likely authorship. Literary and Linguistic Computing, 17, 267–287.

Escalante, H.J., Solorio, T., & Montes, M. (2011). Local histograms of character n-grams for authorship attribution. In Proceedings of the 49h Annual Meeting of the Association for Computational Linguistic (ACL 2011) (pp. 288–298). Portland, OR: ACL.

Grieve, J. (2007). Quantitative authorship attribution: An evaluation of techniques. Literary and Linguistic Computing, 22, 251–270.

Hoover, D.L. (2003). Multivariate analysis and the study of style variation. Literary and Linguistic Computing, 18, 341–360.

Houvardas, J., & Stamatatos, E. (2006). N-gram feature selection for author identification. In Proceedings of the 12th International Conference on Artificial Intelligence: Methodology, Systems, and Applications (ICAI 2006) (pp. 77–86). Varna, Bulgaria: ICAI.

Juola, P. (2008). Authorship attribution. Foundations and Trends in Information Retrieval, 1(3), 233–334.

Juola, P., & Baayen, R.H. (2005). A controlled-corpus experiment in authorship identification by cross-entropy. Literary and Linguistic Computing, 20, 59–67

Keselj, V., Peng, F., Cercone, N., & Thomas, C. (2003). N-gram-based author profiles for authorship attribution. In Proceeding of PACLING'03 (pp. 255–264). Halifax, Canada: PACLING.

Koppel, M., & Schler, J. (2004). Authorship verification as a one-class classification problem. In Proceedings of 21st International Conference on Machine Learning (ICML 2004) (pp. 489–495). Banff, Canada: ICML.

Koppel, M., Schler, J., & Argamon, S. (2008). Computational methods in authorship attribution. Journal of the American Society for Information Science and Technology, 60(1), 9–26.

Koppel, M., Schler, J., & Argamon, S. (2011). Authorship attribution in the wild. Language Resources and Evaluation, 45(1), 83–94.

Luyckx, K., & Daelemans, W. (2008). Authorship attribution and verification with many authors and limited data. In Proceedings of the 22nd International Conference on Computational Linguistics (COLING 2008) (pp. 513–520). Manchester, UK: CONLING.

Luyckx, K., & Daelemans, W. (2010). The effect of author set size and data size in authorship attribution. Literary and Linguistic Computing, 1–21.

Madigan, D., Genkin, A., Lewis, D.D., Argamon, S., Fradkin, D., & Ye, L. (2005). Author identification on the large scale. In Proceedings of the Meeting of the Classification Society of North America, 2005.

Malyutov, M. (2006). Information transfer and combinatories. Lecture Notes in Computer Science 4123, 3.

Meyer zu Eissen, S., Stein, B., & Kulig, M. (2007). Plagiarism detection without reference collections. In R. Decker & H.J. Lenz (Eds.), Advances in data analysis (pp. 359–366). Berlin: Springer.

Narayanan, A., Paskov, H., Gong, N., Bethencourt, J., Stefanov, E., Shin, R. (2012). On the feasibility of Internet-scale author identification. IEEE S&P, 300–314. doi: 10.1109/SP.2012.46

Novak, J., Raghavan, P., & Tomkins, A. (2004). Anti-aliasing on the web. In Proceedings of the 13th International Conference on World Wide Web (ACM 2004). New York: ACM.

Plakias, S., & Stamatatos, E. (2008). Author identification using a tensor space representation. In Proceedings of the 18th European Conference on Artificial Intelligence (ECAI) (pp. 833–834). Greece: ECAI.

Reynolds, D.A. (1995). Speaker identification and verification using Gaussian mixture speaker models. Speech Communication 17(1–2): 91–108.

Sanderson, C., & Guenter, S. (2006). Short text authorship attribution via sequence kernels, Markov chains and author unmasking: An investigation. In Proceedings of Int'l Conference on Empirical Methods in Natural Language Processing (EMNLP 2006) (pp. 482–491). Sydney, Australia: EMNLP.

Stamatatos, E. (2009). A survey of modern authorship attribution methods. Journal of the American Society for Information Science and Technology, 60(3), 538–556.

Uzuner, U., & Katz, B. (2005). A comparative study of language models for book and author recognition. Lecture Notes in Computer Science, 3651, 969–980.

Winter, Y. (2012). Determining whether two anonymous short documents are by the same author. Unpublished M.Sc. dissertation, Dept. of Computer Science, Bar-Ilan University.

Zelikovitz, S., Cohen, W., & Hirsh, H. (2007). Extending WHIRL with background knowledge for improved text classification. Information Retrieval, 10(1), 35–67.

Zhao, Y., & Zobel, J. (2005). Effective authorship attribution using function word. In Proceedings of the 2nd Asian Information Retrieval Symposium (AIRS 2005) (pp. 174–190). New York.

Zheng, R., Li, J., Chen, H., & Huang, Z. (2006). A framework for authorship identification of online messages: Writing-style features and classification techniques. Journal of the American Society for Information Science and Technology, 57(3), 378–393.

DOI: 10.1002/asi