A Simple and Efficient Algorithm for Authorship Verification

Mirco Kocher
University of Neuchatel, rue Emile Argand 11, 2000 Neuchatel, Switzerland. E-mail: Mirco.Kocher@unine.ch

Jacques Savoy
University of Neuchatel, rue Emile Argand 11, 2000 Neuchatel, Switzerland. E-mail: Jacques.Savoy@unine.ch

This paper describes and evaluates an unsupervised and effective authorship verification model called SPATIUM-L1. As features, we suggest using the 200 most frequent terms of the disputed text (isolated words and punctuation symbols). Applying a simple distance measure and a set of impostors, we can determine whether or not the disputed text was written by the proposed author. Moreover, based on a simple rule we can define when there is enough evidence to propose an answer or when the attribution scheme is unable to make a decision with a high degree of certainty. Evaluations based on 6 test collections (PAN CLEF 2014 evaluation campaign) indicate that SPATIUM-L1 usually appears in the top 3 best verification systems, and on an aggregate measure, presents the best performance. The suggested strategy can be adapted without any problem to different Indo-European languages (such as English, Dutch, Spanish, and Greek) or genres (essay, novel, review, and newspaper article).

Introduction

Automatic authorship attribution aims to determine, as accurately as possible, the true author of a whole document or a text excerpt (Stamatatos, 2009). To achieve this, a sample of texts written by each of the possible authors is needed. From this common starting point, different contexts can be encountered. In the closed-class attribution problem, the real author is one of several given possible candidates. Within the open-class problem, the real author might be one of the specified writers or another unknown one. In the verification question, the system must be able to determine whether or not a given author did in fact write a given text (e.g., a testimony, a letter, a threatening e-mail, etc.). Finally, authorship attribution can be limited to a profiling view (Pennebaker, 2011), where the system must mine demographic or psychological information about the author (e.g., gender, age, social status, personality traits, etc.).

In this paper we are using some well known historical questions such as "are the *Commentarii de Bello Gallico* (*The Gallic Wars*) really written by Julius Caesar?" or "Which parts of the *Book of the Mormon* are 'translated' by Joseph Smith?" (Jockers, Witten, & Criddle, 2010). With the Internet, the number of anonymous or pseudonymous texts is increasing. Therefore, proposing an effective algorithm for the verification problem represents an indisputable interest. Even though the answer to this verification process can be limited to a binary value (yes/no), a better output is to include a justification supporting the proposed answer. Moreover, an estimated degree of belief (or probability) that the given answer is correct will improve the confidence attached to the system response (Savoy, 2016).

This authorship verification question seems simpler than the classical authorship attribution problem, but it is not. For example, if we want to know if a newly discovered poem was really written by Shakespeare (Craig & Kinney, 2009; Thisted & Efron, 1987), the computer needs to compare a model based on Shakespeare's texts with all other possible representative non-Shakespeare models. This second part is hard to generate. Are we sure we have included all other writers having a style similar to Shakespeare? Moreover, we might take into account the fact that personal style might evolve during an author's life.

This paper is organized as follows. The next section describes the state of the art in authorship attribution and verification. We then go on to explain our proposed algorithm, called SPATIUM-L1. In the section that follows, we

Received May 1, 2015; revised August 18, 2015; accepted August 19, 2015

© 2015 ASIS&T • Published online 11 January 2016 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/asi.23648

present our test collections and the evaluation methods used in our experiments. Afterwards, we evaluate the proposed scheme and compare it to the best-performing schemes using six different test collections written in four distinct languages and genres. In the last section, an analysis of the results explains why the proposed algorithm works correctly or sometimes may fail to provide the correct answer. A conclusion summarizes the main findings of this study.

State of the Art

To solve the authorship attribution problem, a first set of approaches is based on unitary invariant values (Holmes, 1998). These invariant measures must reflect the particular style of a given author, but they should vary from one author to another. Following this perspective, we can find the use of lexical richness measures or word distribution factors, including average word length and mean sentence length, as well as Yule's *K* measure and statistics on type-token ratios (e.g., Herdan's *C*, Guiraud's *R*, or Honoré's *H*), and also the proportion of word types occurring once or twice (e.g., Sichel's *S*). None of these measures has proven very satisfactory, due in part to word distributions (including word bigrams or trigrams) dominated by a large number of very low probability elements (Large Number of Rare Events) (Baayen, 2008).

As a second family of approaches, we could apply multivariate analysis to capture each author's discriminative stylistic features. Some of the main approaches applicable here are principal component analysis (PCA) (Binonga & Smith, 1999; Craig & Kinney, 2009; Holmes & Crofts, 2010), cluster analysis (Labbé, 2007), and discriminant analysis (Jockers & Witten, 2010).

As a third set of approaches, various effective machine-learning classifiers have been proposed, such as *k*-nearest neighbors, naïve Bayes, decision tree, support vector machine, etc. (Stamatatos, 2009). Even if various classification strategies have been proposed, the general common procedure is the following (Juola, 2006). First, text samples are collected for each possible author. Based on these samples, a feature selection scheme might be applied to choose the most appropriate features able to discriminate between the possible authors. Then the classifier learns the discriminative stylistic aspects of each possible author based on those text samples. Finally, the disputed text is given to the learning system to determine the most probable author.

As a fourth type of approach, different distance-based measures have been suggested. Based on the differences in word distribution between authors, this strategy proposes to define a distance between the disputed text and either the author profile (concatenation of all texts written by the corresponding person) or the different texts for which the authorship is known. Well-known examples of this include the Burrows's Delta (2002) based on the top *k* most frequent word types (with $k = 40$ to 1,000), the Kullback-Leibler divergence (Zhao & Zobel, 2007) using a predefined set of 363 English word types, and the use of specific vocabulary (Savoy, 2012), or Labbé's method (2007) using the whole vocabulary.

Various modifications of these attribution strategies can be applied in the more specific verification question. First, as for other authorship attribution problems, we need to extract style markers, and different feature sets can be used (e.g., *k* most frequent word types, functional words, frequencies of selected letters or *n*-grams of characters, part-of-speech [POS] *n*-grams, etc.) (Sebastiani, 2002; Juola, 2006; Stamatatos, 2009). The second step is to select a binary classifier able to discriminate between the proposed author (let's say, A) and all others (not-A). During the classification investigation, we can consider the disputed text (denoted Q) as a whole or we can extract from it a sequence of *c* chunks (e.g., each composed of 500 word tokens) and consider the result obtained by these *c* subparts of Q (Koppel, Schler, & Bonchek-Dokow, 2007).

A classical solution is to consider the proposed author A with a set of other possible writers called *impostors* (with a text sample for each of them). We then train a set of binary classifiers to learn models for A versus not-A, B versus not-B, etc. The *c* chunks of the doubtful text are then classified according to our learned models, and, if a preponderance of chunks is classified as A, then we conclude that A is the real author. Otherwise, we can infer that another unknown person wrote the text (Koppel & Winter, 2014). This strategy may fail if we do not consider all writers having a style similar to A. For example, we might have ignored author D depicting a style very similar to A. As soon as a classifier proposes A for a given chunk, we are never sure whether the author is really A or D. When applying such an attribution strategy, it is important to have imposters' texts written in the same period, genre, and on the same topics in order to keep constant other stylistic source variations than the author himself.

Another solution proposed by Koppel et al. (2007) is based on the unmasking technique. For each of the possible authors (let's say we have *m* candidates), we build a learning model with the *k* most frequent word types. We then determine the accuracy of the *m* models. From that point, we iterate a given number of times. After each iteration we remove a few strongly weighted positive and a few strongly weighted negative features. Finally, we plot the degradation of the performance achieved by the *m* models.

Using this approach, the performance graph will depict similar curves for all writers except the real author. To be more precise, when removing features strongly related to the true writer, the performance corresponding to him will clearly drop. Doing the same with another person, who is not the real author, the performance will only slightly decrease because the removed features do not present a strong association between the disputed text and this non-author. Of course, if no clear difference appears, with one author performing clearly worse than the rest, we may conclude that none of the proposed writers is the real one. However, the decision is somehow arbitrary; a decreased performance could be interpreted as marginal or substantial.

DOI: 10.1002/asi

The experiments supporting previous studies were usually limited to one language, one author, and one or a few texts. For real cases, this limitation makes sense; for example, we have only one newly discovered poem that might be attributed to Shakespeare (Thisted & Efron, 1987). To evaluate the effectiveness of a verification algorithm, the number of tests should, however, be larger. To create such benchmarks, and to promote studies in this domain, the PAN CLEF 2014 evaluation campaign was launched (Stamatatos et al., 2014). Thirteen research groups with different backgrounds from around the world participated in the PAN CLEF 2014 campaign. Each team has proposed a verification strategy that has been evaluated using the same method.

During the PAN CLEF 2014 campaign, various representations and classifiers were proposed. The best-performing system was based on the impostors' strategy in which each document is represented by numerous *n*-grams of letters and word types, as well as part-of-speech tags, with the number of features ranging from 3,300 to 73,000 (Khonji & Iraqi, 2014). A distance measure is applied to determine whether the query text is written or not by the proposed author. Moreover, to generate more possible impostors, texts have been downloaded from the web. Finally, the processing time of this solution was clearly more expensive (around 21 hours for around 800 verifications) than the others (around 2 hours).

The second-best performance was achieved using a decision tree model (CART algorithm) based on 17 distinct similarity measures, each of them based on numerous features (e.g., character 3-grams weighted by *tf idf*, correlation similarity, bigrams of word types) (Fréry, Largeron, & Juganaru-Mathieu, 2014). The third-best effectiveness was achieved by representing documents by three indexing schemes: all words, LSA (latent semantic indexing) (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990) using all words, and a combined surrogate based on prefixes, suffixes, *n*-grams (with $n = 1, 2, \ldots, 5$), punctuation symbols, stopwords, vowel combinations, and permutations (Castillo, Cervantes, Vilriño, Pinto, & León, 2014). The similarity between documents is defined as the maximum when considering four different similarity measures (cosine, Jaccard, Euclidian distance, Chebyshev). If the resulting similarity is higher than a given learned threshold, the system assumes that the same author has written the two texts.

As a general trend, we can see that text representation strategies are based on both *n*-grams of letters and other complementary schemes (e.g., POS tags, word types, LSA). The number of features therefore tends to be high, and larger than 1,000. The most effective solutions are based on machine-learning classifiers and the different research groups use distinct learning schemes. During the PAN CLEF 2014 evaluation campaign, the most effective approaches have chosen the impostors' strategy.

Simple Verification Algorithm

To solve the verification problem, we suggest an unsupervised approach based on a simple feature extraction and distance measure called SPATIUM-L1 (a Latin word meaning distance). The selected stylistic features correspond to the top *k* most frequent terms (isolated word types without stemming but with the punctuation symbols). Those terms are selected for the disputed text. For determining the value of *k*, previous studies have shown that a value between 200 and 300 tends to provide the best performance (Burrows, 2002; Savoy, 2015). This reduced number represents a huge difference compared to the 100,000 features used by Koppel and Winter (2014) or compared to the features set size employed in the best systems employed in PAN CLEF 2014. Moreover, the justification of the decision will be simpler to understand because it will be based on word types instead of letters, bigrams of letters, or combinations of several representation schemes or distance measures.

In the current study, a verification problem is defined as a query text, denoted Q, and a set of texts (between 1 and 5) written by the same proposed author. The concatenation of these texts forms the author profile A. To measure the distance between Q and A, SPATIUM-L1 uses the L1-norm as follows:

$$\Delta(Q, A) = \Delta_0 = \sum_{i=1}^{k} |P_Q[t_i] - P_A[t_i]| \quad (1)$$

where *k* indicates the number of term types (word types or punctuation symbols), and $P_Q[t_i]$ and $P_A[t_i]$ represent the estimated occurrence probability of the term t_i in the query text Q or in the author profile A, respectively. To estimate these probabilities, we divide the term occurrence frequency (denoted tf_i) by the length in tokens of the corresponding text (n), $\text{Prob}[t_i] = tf_i / n$.

To verify whether the resulting Δ_0 value is small or rather large, we need to select a set of impostors. To achieve this, three profiles from other problems in the test set were chosen randomly. This value of three is arbitrary and will be denoted by the variable *m*. After computing the distance between Q and each of these *m* profiles, we retain only the smallest distance.

Instead of limiting the number of possible impostors to *m*, we iterate this last stage *r* times, and we suggest fixing the value $r = 5$. After this last step, we have *r* values denoted $\Delta_{m1}, \ldots, \Delta_{mr}$, each of them corresponding to the minimum value of a set of *m* impostors. Instead of working with *r* values, we compute the mean, denoted Δ_m, of the sample $\Delta_{m1}, \ldots, \Delta_{mr}$.

Finally, the decision rule is based on the value of the ratio Δ_0 / Δ_m as follows:

$$\begin{cases} \text{if } \Delta_0/\Delta_m < 0.975 & \textit{same author} \\ \text{if } \Delta_0/\Delta_m > 1.025 & \textit{different authors} \\ \quad \text{otherwise} & \textit{don't know} \end{cases} \quad (2)$$

Thus, when the Δ_0 value is similar to Δ_m (in the range ±2.5%), the system specifies that the solution of this problem cannot be determined with good certainty and provides the answer *don't know*. On the other hand, when Δ_0 is small compared to Δ_m, the evidence is in favor of assuming that the author of profile A is the real author. Finally, when Δ_m is small compared to Δ_0, we conclude that Q and A are written by different authors. The limit of two times 2.5% was chosen arbitrarily but corresponds to a well-known limit value in statistical tests.

Instead of considering complex text representations, we opt for simpler ones based on the most frequent word types. This strategy has the drawback of ignoring some stylistic features such as POS distribution, complex sentence construction measures, or other type-token ratios. On the other hand, simpler text representation approaches have the advantage of simplicity, have proven to be efficient (Burrows, 2002; Hoover, 2004; Savoy, 2015), and can be understood by the final user. After an attribution has been proposed by the system, the final user may require a justification (e.g., in a court decision). To achieve this, working with frequent words the generation of such an explanation is simpler than having to extract information in a huge space of features (e.g., more than 2,000) or in complex text representation models.

As an attribution method, we propose a simple distance measure (Equation [1]) instead of a complex learning scheme usually based on a "black box" strategy (e.g., neural network, support vector machine [SVM], combination of multiple attribution models). Even if the current computer technology allows us to deploy such complex approaches, the resulting effectiveness depends on large and representative training data sets. Moreover, simpler attribution schemes may provide a high or very high level of effectiveness (Holte, 1993). For example, Hand (2006) shows that for 10 well-known data sets, the difference in performance between the best method and a simple linear approach varies from 15% to 0% (in three cases, the simple linear model produces the best possible answer).

Test Collections and Evaluation Method

During PAN CLEF 2014, six test collections were built, each containing between 100 to 200 problems. In this context, a problem is defined as: *given a set of documents (between one and five) written by the same author, is the new document also written by that author?* In each collection, all the texts matched the same language, genre, and time period. Thus, important factors related to the style are kept constant, and the main remaining stylistic variations can be related to the author. The topics of the text are recognized as having a clear impact on the vocabulary but this factor varies from one document to the other. In fact, it is usually impossible to keep this parameter constant in a test collection.

This test collection includes texts written in four different languages: English, Dutch, Spanish, and Greek. More precisely, we can find two benchmarks for the English and Dutch languages, and only one is written in Spanish or Greek. These last two corpora contain newspaper opinion articles extracted from the newspapers *El Pais* and *To Bhma*. The Dutch collections were written by students, either as an essay or a review. Authors of the English essay corpus were Finnish students having English as their second language. The second English corpus is composed of short novels (horror fiction). In total, we count four different genres in these six benchmarks.

An overview of these test collections is depicted in Table 1 in which the column "Training" indicates the number of problems in the training set. We will ignore the training set in order to be able to compare our results with those of the PAN CLEF 2014 campaign. For the test set, the number of problems is given under the label "# Problems." The mean number of documents for each problem in the test set is indicated in the column "Mean document," and the mean number of word tokens per document under the label "Mean words." For example, with the English novel corpus the style of the proposed author can be analyzed as having, on average, one document containing 6,104 word tokens.

When inspecting the Dutch collections, the number of words available is rather small (mean 116 word tokens for each review, and $2 \times 398 = 796$ mean per essay). When studying the relation between the size of text samples and the accuracy of authorship attribution methods, Eder (2015) found that a minimum length of 5,000 word tokens is required to provide stable results. To obtain reliable attributions, Labbé (2007) suggests working with disputed texts having at least 10,000 word tokens. Therefore, we can expect the mean performance for this language to be lower than that for the other languages. For the Spanish corpus, Table 1 indicates that we have, on average, five documents to learn the stylistic features of the proposed author. A

TABLE 1. PAN CLEF 2014 corpora statistics.

		Training	Test		
Language	Genre	# Problems	# Problems	Mean documents per problem	Mean words per problem
English	essay (EE)	200	200	2.6	833
English	novel (EN)	100	200	1.0	6,104
Dutch	essay (DE)	96	96	2.0	398
Dutch	review (DR)	100	100	1.0	116
Spanish	article (SA)	100	100	5.0	1,537
Greek	article (EA)	100	100	2.7	1,121

TABLE 2. Evaluation over all six test collections (micro-averaging).

Rank	Run	c@1	Merit-0	Merit-1	Merit-2	Success
1	*Meta-classifier*	**0.713**	**568**	340	112	**0.714**
2	Spatium-L1	0.687	535*	**344**	**153**	0.709
3	Fréry et al. (2014)	0.684	540	298	56	0.685
4	Khonji and Iraqi (2014)	0.683	543	291	39	0.683
5	Castillo et al. (2014)	0.676	529*	301	73	0.682
6	Baseline (yes)	0.5	398*	0	–398	0.500

relatively higher performance can be assumed with this benchmark. A similar conclusion can be expected with the English novels collection consisting of longer documents (mean, 6,104 word tokens).

When considering the six benchmarks as a whole, we have 796 problems to solve. When inspecting the distribution of the correct answers, we can find the same number (398) as positive or negative answers. In each of the individual test collections, we can also find a balanced number of positive and negative answers.

During PAN CLEF 2014, a system must return a value between 0.0 and 1.0 for each problem. A value larger than 0.5 indicates that the query text was written by the proposed author and a value lower than 0.5 the opposite. Returning the value 0.5 indicates that the system is unable to make a decision based on the given information. Of course, a value closer to 1.0 (or to 0.0) can be viewed as stronger evidence in favor of (or against) the authorship.

As a performance measure, two evaluation measures were used during the PAN CLEF campaign. The first performance measure is the area under the curve (AUC) of the receiver operating characteristic (ROC) curve (Witten, Frank, & Hall, 2011). This curve is generated according to the percentage of false positives (or false alarms) in the x-axis and the percentage of true positives in the y-axis over the entire test set. The maximum value of 1.0 indicates a perfect performance. Both the ROC and the AUC measures are, however, rather complex and difficult to interpret by the final user.

As another measure, the PAN CLEF campaign adopts the c@1 measure (Peñas & Rodrigo, 2011). This evaluation measure takes into account both the number of correct answers and the number of problems left unsolved in the whole test set. The exact formulation is given in Equation (3), with a minimum value of 0 and an optimum value of 1.

$$c@1 = {}^{1}\!/_{np} \cdot \left(nc + {}^{nc}\!/_{np} \cdot nu\right) = {}^{nc}\!/_{np} \cdot \left(1 + {}^{nu}\!/_{np}\right) \quad (3)$$

in which *np* is the number of problems, *nc* the number of correct answers, and *nu* the number of problems left without an answer. This measure differentiates between an incorrect answer and the absence of an answer (indicating that the provided evidence is not enough to make a definitive decision) (Stamatatos et al., 2014). For example, with $np = 100$ and $nc = 80$ ($nu = 0$), the accuracy rate is $nc/np = 0.8$, and c@1 gives the same value. But when 10 of the "incorrect" decisions are left without an answer ($nu = 10$), the c@1 measure does not view them as wrong, and the c@1 = 0.88.

As additional performance measures that can take account of the answer *don't know*, we can attribute 1 point when the decision is correct, 0 when it is incorrect, and 0.5 when the system decision is *don't know*. To determine the quality of an attribution scheme, we can sum these values (or compute a relative value) to define a merit score. Of course, we can also specify that an incorrect decision must be penalized more strongly and attribute a value of −1 or −2 for such wrong attributions. We will report this performance measure in our evaluations.

Finally, to statistically determine whether or not a given verification strategy would be better than another, we applied the sign test (Conover, 1980). This test is rather conservative and requires strong evidence to detect a statistically significant performance difference. More precisely, when comparing two attribution schemes, the sign test considers only the direction of the difference, denoted by a + or – sign. When the two schemes return the same decision, this observation is ignored. When the decision differs, we assign the sign + if the first scheme returns a better answer than the second one. In the reverse case, this observation receives the negative sign. As the null hypothesis H_0, we assume that both verification schemes produce similar performances. Such a null hypothesis would be accepted if two verification schemes returned statistically similar decisions, otherwise it must be rejected. Thus, when H_0 is true, the number of + must be similar to the number of −. On the other hand, when the number of the two signs diverges, there is a small probability that H_0 is true. In the experiments presented in this paper we limit this probability to 5%. In other words, statistically significant differences are detected by a two-sided sign test (significance level 5%).

Evaluation

Based on the described evaluation method, we achieved the overall results depicted in Table 2 corresponding to the 796 problems present in the six test collections. These means are computed using the micro-averaging principle in which each decision has the same importance. In this table we have reported one performance measure applied during the PAN CLEF campaign, namely, the c@1. These values will be used to rank the different attribution strategies.

As additional information, Table 2 shows three additional measures. Under the label "Merit-0," we assume that a good answer counts as 1 point, the decision *don't know* 0.5, while an incorrect answer returns 0. As a more complete answer, the attribution system may provide a degree of belief that the proposed attribution is correct (Savoy, 2016). Of course the ultimate goal is to reach a zero-mistake rate. When an error-free system is unlikely, we should penalize the wrong decisions. We clearly prefer a system able to know when "it doesn't know" and provide an answer when the evidence is strong enough to make a decision. Providing wrong answers clearly hurts the credibility of an automatic system. Faced with stupid or incorrect answers, the end user will lose his confidence in the system. Such an attribution scheme cannot be used, for example, to support court decisions.

To reflect this perspective, we attribute −1 point for an incorrect decision under the label "Merit-1," and −2 points under the column "Merit-2." As we can see, Spatium-L1 proposes the highest performance with these measures. Finally, the last column "Success" indicates the proportion of correct decisions when ignoring the answers *don't know*.

In Table 2, we have added the system Meta-classifier corresponding to the combination of all 13 systems submitted at the PAN CLEF 2014 evaluation campaign (but without the Spatium-L1 system). The underlying decision is based on an aggregation of the answers obtained by the 13 systems. We have also added a baseline corresponding to a system that always produces the answer *yes* (trivial acceptor). For each evaluation measure, the best performance is indicted in bold.

The last line of Table 2 corresponds to the trivial acceptor, and this baseline achieves a value of 0.5 under the performance c@1. The score under the "Merit-0" column is 398 and reflects the fact that this baseline answered correctly 398 problems over 796. With the "Merit-1" measure, the performance drops to zero because the number of correct and incorrect decisions is the same. Using the "Merit-2" measure, the performance is negative (−398) since the weight of an incorrect decision is −2. Ignoring the decisions *don't know*, the proportion of correct answers is 0.5, as indicated in the last column.

When comparing the different strategies using the c@1 values, Table 2 indicates that the performance differences are usually small, except with the trivial acceptor. The Meta-classifier tends, however, to present a slightly better performance (0.713). It is, however, difficult to clearly understand the differences in the system behaviors with this measure. Inspecting the three merit measures, we can see that the Spatium-L1 system provides good overall performance. These high values can be explained by the fact that this verification scheme tends to opt more often for a *don't know* answer when the decision is uncertain. Having enough evidence (see Equation [2]), Spatium-L1 is then able to propose either a positive or a negative answer.

Using the best performance as a baseline (the first row in Table 2), we compared its effectiveness with other verification models. Statistically significant differences detected by the sign test (two-sided, significance level 5%) are indicated by an asterisk (*) after the corresponding "Merit-0" value. The Meta-classifier tends to propose a statistically better performance than the other attribution schemes, except with Frery's or Khonji and Iraqi's classifier, where the performance difference cannot be viewed as significant.

To have an overview of the individual test collections, we report in the Appendix the performance across the six benchmarks and for the three best verification schemes.

Finally, to gain a better understanding of the choice of the two different parameters within the Statium-L1 classifier, we performed various experiments. We can modify the number of rounds *r* (fixed at 5) and the number *k* of the most frequent word types (fixed at 200). Varying the value of *r* from 1 to 7, and the value of *k* from 40 to 400, the highest c@1 value obtained was 0.691, with a Merit-0 score of 541. From a statistical point of view, this difference is not significant compared to the performance reported in Table 2.

The last possible parameter is the value of 2.5% used in Equation (2) to define when the Statium-L1 classifier is able to make a decision with some certainty. Increasing this percentage to 4% or decreasing it to 1.5% does not significantly modify the overall performance. For some languages and genres, such a modification could improve the effectiveness, while for others the same change will hurt the performance. Figure 1 illustrates the performance change in the six corpora when varying the threshold around the proposed 2.5% value. Reducing this threshold to 1% or below tends to force the system to always make a decision without enough evidence. The overall performance (depicted in Figure 1 with the line labeled "Mean") therefore decreases. On the other hand, selecting a value larger than 5% encourages the classifier not to make a decision. Answering more often *don't know* will reduce the performance over a correct decision and the overall performance tends to be clearly reduced.

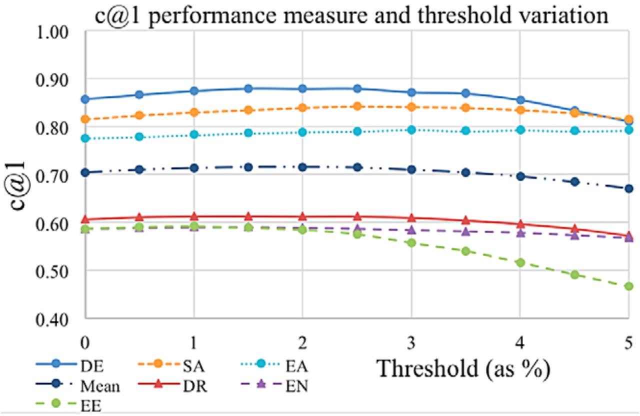

FIG. 1. Relation between the performance and the threshold variation (proposed value 2.5%). [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

TABLE 3. Evaluation over the two English collections (micro-averaging, 400 problems).

Run	c@1	Merit-0	Merit-1	Merit-2	Success
Spatium-L1	0.58	233.5	107.5	−18.5	0.61
Spatium-L1 (Zhao & Zobel, 2007)	0.50	206	46	−114	0.52
Spatium-L1 (Hughes et al., 2012)	0.53	214	57	−100	0.54
Baseline (yes)	0.5	200	0	−200	0.5

As the number of features, we suggest taking the 200 most frequent word types and punctuation marks in the disputed text. Instead of having a list varying from one text to another, we can opt for a fixed prior list of word types. In authorship attribution studies, Zhao and Zobel (2007) propose that such a list contains 363 English word types (composed mainly of function words). Likewise, Hughes, Foti, Krakauer, and Rockmore (2012) suggest a list of 309 English word types. To verify whether those lists may support a better overall performance, Table 3 reports the different performance measures with these two lists compared to the proposed scheme. As we can see, the performance differences are small and statistically not significant. Having a prior list of discriminative word types could simplify the attribution scheme. We need, however, to define such a list for each language used.

Deeper Analysis

In text categorization studies, we are convinced that a deeper analysis of the evaluation results is important to obtain a better understanding of the advantages and drawbacks of a suggested scheme. By just focusing on overall performance measures, we only observe a general behavior or trend without being able to develop a better explanation of the proposed assignment. To achieve this deeper understanding, we will analyze some problems extracted from the English essays (EE) corpus. Usually, the relative frequency (or probability) differences with very frequent word types such as *when*, *is*, *in*, *that*, *to*, or *it* can explain the decision. In the following discussion, and to simplify the presentation, we only mention the probability of one (the best) of the randomly chosen candidates (instead of considering the $m = 3$ candidates or impostors), and we will evaluate the decision after one iteration (instead of $r = 5$).

As a first correct (true negative) example, we selected Problem #EE002. In this case, the pronoun/determiner *that* has a probability of 0.009 in the query text compared to 0.019 in the proposed author profile and 0.009 in the best candidate. For the auxiliary verb *is*, the probabilities are 0.014 (query), 0.038 (profile), and 0.015 (candidate). The conjunction *and* appears with a relative frequency of 0.021 in the query text, compared to 0.039 (profile), and 0.023 (impostor). As we can see, these three terms tend to indicate that the profile of the proposed author is not the real one, while the best impostor appears more credible. However, not all of the 200 terms follow the same pattern. For example, the auxiliary verb *have* is the most decisive term in favor of the profile, with an estimated probability of 0.016 in the query text, compared to 0.010 (profile), and 0.003 (candidate). Moreover, some of the selected terms are related to the topic discussed in the essay, and thus they don't occur in the profile nor in the impostors. For example, we can encounter the words *listening* and *accent*, both appearing with a probability of 0.004 in the query text but not in the others. The L1-distance between the query text and the best impostor is 0.560 while this distance is 0.663 with the profile of the proposed author. The correct decision taken by Spatium-L1 was to answer *different authors* due to the large distance difference.

With Problem #EE224 Spatium-L1 also makes the correct decision (*same author*, true positive). When inspecting the determiner *a*, we have very similar relative frequencies in both the proposed author profile (0.021) and in the query text (0.020), but not in the best candidate (0.014). With the preposition *in*, we found a similar pattern (0.016 in query, 0.018 in the profile, and 0.026 in the impostor). The term *to* tends to confirm this finding with very similar relative frequencies in both the profile and in the query text (0.035) justifying the decision *same author*. For some terms, the probability differences are not always as close. In most cases, however, the probability estimate differences between the query text and the candidate are even higher. As an example, we can inspect the preposition *of* having a probability of 0.007 in the query text, 0.016 in the profile, and 0.023 in the candidate. The conjunction *and* follows the same pattern. In this case, the author uses less frequently the word types *and* and *of* in the query text compared to his profile. Some stylistic variations are always possible, as shown in this example. Finally, the L1-distance of the query to the proposed author profile is 0.601, and the one with the best impostor is about 10% larger (0.663). Most of the probabilities estimates are similar, justifying the decision *same author* (with a moderate degree of belief).

As an example of incorrect decisions returned by Spatium-L1, we can analyze Problem #EE064 (false negative). In this case, the probability for the article *the* is 0.048 in the query text, 0.062 in the author profile, and 0.047 in the best candidate. The negation *not* reinforces this pattern. The probability estimates are 0.012 in the query text, 0.005 in the profile, and 0.012 in the candidate. The punctuation symbol , (comma) is also clearly against the profile with the probabilities 0.059 (query), 0.074 (profile), and 0.055 (candidate). On the other hand, the punctuation mark (period)

DOI: 10.1002/asi

supports the opposite decision; its probability estimates are 0.036 (query), 0.032 (profile), and 0.051 (candidate). The words *Brutus* and *Cassius* are topical terms appearing frequently in the query text (probability estimates 0.017 and 0.015) but they are absent from the other texts. The L1-distance between the query and the best candidate is 0.485, while the distance to the profile is 0.524. The 8% difference leads to the incorrect decision *different authors* (with, however, a weak support).

With Problem #EE527 Spatium-L1 achieves an incorrect decision (false positive), partly because the probability estimate for the term *to* is 0.038 in the query text, 0.035 in the author profile, and 0.022 in the best impostor. With the determiner *the*, the same pattern occurs (0.027 in query, 0.034 in the profile, and 0.051 in the candidate). The pronoun *it* reinforces this finding, with similar frequencies in the query text (0.015), and in the profile (0.018), compared to the best impostor (0.009). The words *unfamiliar* and *subtitles* are topical terms occurring only in the query (0.002) but never in the other texts. The L1-distance between the query and the candidate is 0.479, while the difference with the profile is 0.410, leading to the incorrect decision *same author*. The difference of 17% can be interpreted as a moderate degree of belief supporting this assignment.

Conclusion

This paper proposes a simple, unsupervised technique to solve the authorship verification problem. Unlike many other attribution techniques, the proposed classifier does not require a learning stage to define appropriate values assigned to different parameters. As features to discriminate between the proposed author and different impostors, we propose using the top 200 most frequent terms types (word types and punctuation symbols). This choice was found effective for other related tasks such as authorship attribution (Burrows, 2002). Moreover, compared to various feature selection strategies used in text categorization (Sebastiani, 2002), the most frequent terms tend to select the most discriminative features when applied to stylistic studies (Savoy, 2015). In order to make the attribution decision, we propose using a simple distance measure called Spatium-L1 based on the L1 norm.

The proposed unsupervised approach tends to perform very well in four different languages (English, Dutch, Spanish, and Greek) as well as with four genres (essay, novel, review, and newspaper article). Compared to the PAN CLEF 2014 results, the proposed attribution scheme achieved a performance usually among the three best systems within the six different test collections. When computing an overall mean over the six test collections, Spatium-L1 shows the best performance level. Thanks to this simple implementation, the proposed scheme can be easily used as a strong baseline to evaluate other verification strategies. Such a classifier strategy can be described as having a high bias but a low variance (Hastie, Tibshirani, & Friedman, 2009). Even if the proposed system cannot capture all possible stylistic features (bias), changing the available data does not modify significantly the overall performance (variance).

Moreover, Spatium-L1 returns a numerical value (between 0 and 1) that can be used to determine a degree of certainty (Savoy, 2016). More important, the proposed attribution can be clearly explained because it is based on a reduced set of features, on the one hand, and, on the other, those features are word types or punctuation symbols. Thus, the interpretation for the final user is clearer than when working with a huge number of features, when dealing with *n*-grams of letters, or when combining several similarity measures. The Spatium-L1 decision can be explained by large differences in relative frequencies (or probabilities) of frequent words, usually corresponding to functional terms.

To improve the current classifier, we will investigate the effect of other distance measures as well as other feature selection strategies. In this latter case, we want to maintain a reduced number of term types. In a better feature selection scheme, we can take account of the underlying text genre, as, for example, the most frequent use of personal pronouns in narrative texts. As another possible improvement, we can ignore specific topical terms or character names appearing frequently in an author profile, terms that can be selected in the feature set without being useful in discriminating between authors.

Finally, being able to accurately estimate the degree of belief or certainty of a proposed decision is an important aspect, however often neglected in authorship attribution studies. Producing many wrong decisions, especially without warning, will seriously damage the credibility of an attribution scheme. Therefore, each automatic decision should be given with some degree of support reflecting the quality and quantity of evidence in favor of the proposed decision.

Acknowledgments

We would like to thank the reviewers for their helpful suggestions and remarks. This research was supported, in part, by the NSF under Grant #200021_149665/1.

References

Baayen, H.R. (2008). Analysis linguistic data: A practical introduction to statistics using R. Cambridge, UK: Cambridge University Press.

Binonga, J.N.G., & Smith, M.W. (1999). The application of principal component analysis to stylometry. Literary and Linguistic Computing, 14(4), 445–465.

Burrows, J.F. (2002). Delta: A measure of stylistic difference and a guide to likely authorship. Literary and Linguistic Computing, 17(3), 267–287.

Castillo, E., Cervantes, O., Vilriño, D., Pinto, D., & León, S. (2014). Unsupervised method for the authorship identification task. In L. Cappellato, N. Ferro, M. Halvey, & W. Kraaij (Eds.), Proceedings CLEF-2014, Working Notes (pp. 1035–1041). Aachen, Germany: CEUR.

Conover, W.J. (1980). Practical nonparametric statistics. New York: John Wiley & Sons.

Craig, H., & Kinney, A.F. (2009). Shakespeare, computers, and the mystery of authorship. Cambridge, UK: Cambridge University Press.

Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., & Harshman, R. (1990). Indexing by latent semantic indexing. Journal of American Society for Information Science & Technology, 41(6), 391–407.

Eder, M. (2015). Does size matter? Authorship attribution, small samples, big problem. Digital Scholarship in the Humanities, 30(2), 167–182.

Fréry, J., Largeron, C., & Juganaru-Mathieu, M. (2014). UJM at CLEF in author identification. In L. Cappellato, N. Ferro, M. Halvey, & W. Kraaij (Eds.), Proceedings CLEF-2014, Working Notes (pp. 1042–1048). Aachen, Germany: CEUR.

Hand, D.J. (2006). Classifier technology and the illusion of progress. Statistical Science, 21(1), 1–14.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning. data mining, inference, and prediction. New York: Springer.

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

Holmes, D.I., & Crofts, D.W. (2010). The diary of a public man: A case study in traditional and non-traditional authorship attribution. Literary and Linguistic Computing, 25(2), 179–197.

Holte, R.C. (1993). Very simple classification rules perform well on most commonly used datasets. Machine Learning, 11(1), 63–90.

Hoover, D.L. (2004). Testing Burrows's Delta. Literary and Linguistic Computing, 19(4), 453–475.

Hughes, J.M., Foti, N.J., Krakauer, D.C., & Rockmore, D.N. (2012). Quantitative patterns of stylistic influence in the evolution of literature. Proceedings of the National Academy of Science 109(20), 7682–7686.

Jockers, M.L., & Witten, D.M. (2010). A comparative study of machine learning methods for authorship attribution. Literary and Linguistic Computing, 25(2), 215–223.

Jockers, M.L., Witten, D.M., & Criddle, C.S. (2010). Reassessing authorship of the *Book of Mormon* using delta and nearest shrunken centroid classification. Literary and Linguistic Computing, 23(4), 465–491.

Juola, P. (2006). Authorship attribution. Foundations and Trends in Information Retrieval, 1(3), 1–104.

Khonji, M., & Iraqi, Y. (2014). A Slightly-modified GI-based Author-verifier with Lots of Features (ASGALF). In L. Cappellato, N. Ferro, M. Halvey, & W. Kraaij (Eds.), Proceedings CLEF-2014, Working Notes (pp. 977–983). Aachen, Germany: CEUR.

Koppel, M., & Winter, Y. (2014). Determining if two documents are by the same author. Journal of American Society for Information Science & Technology, 65(1), 178–187.

Koppel, M., Schler, J., & Bonchek-Dokow, E. (2007). Measuring differentiability: Unmasking pseudonymous authors. Journal of Machine Learning Research, 8(6), 1261–1276.

Labbé, D. (2007). Experiments on authorship attribution by intertextual distance in English. Journal of Quantitative Linguistics, 14(1), 33–80.

Pennebaker, J.W. (2011). The secret life of pronouns. What our words say about us. New York: Bloomsbury Press.

Peñas, A., & Rodrigo, A. (2011). A single measure to assess nonresponse. In *Proceedings 49th ACL*, 1415–1424.

Savoy, J. (2012). Authorship attribution based on specific vocabulary. ACM—Transactions on Information Systems, 30(2), 170–199.

Savoy, J. (2015). Comparative evaluation of term selection functions for authorship attribution. Digital Scholarship in the Humanities, 30(2), 246–261.

Savoy, J. (2016). Estimating the probability of an authorship attribution. Journal of American Society for Information Science & Technology, DOI: 10.1002/asi.23455 in print.

Sebastiani, F. (2002). Machine learning in automatic text categorization. ACM Computing Survey, 34(1), 1–27.

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

Stamatatos, E., Daelemans, W., Verhoeven, B., Sanchez-Perez, M.A., Stein, B., Juola, P., . . . Barrón-Cadeño, A. (2014). Overview of the author identification task at PAN 2014. In L. Cappellato, N. Ferro, M. Halvey, & W. Kraaij (Eds.), Proceedings CLEF-2014, Working Notes (pp. 877–897). Aachen, Germany: CEUR.

Thisted, R., & Efron, B. (1987). Did Shakespeare write a newly-discovered poem? Biometrika, 74(3), 445–456.

Witten, I.H., Frank, E., & Hall, M.A. (2011). Data mining. Practical machine learning tools and techniques (3rd ed.). Burlington, MA: Morgan Kaufmann.

Zhao, Y., & Zobel, J. (2007). Searching with style: Authorship attribution in classic literature. In Proceedings of the Thirtieth Australasian Computer Science Conference (ACSC2007) (pp. 59–68). Ballarat: CRPIT.

Appendix

To have an overview of the individual test collections, we report in this Appendix the performance across the six benchmarks for the three best verification schemes. For example, Table A.1 reports the performance obtained with the English essays corpus (200 problems), while Table A.3 for the Dutch essays collection (96 problems), and Table A.5 for the Spanish newspaper articles corpus (100 problems). In these tables we used the Meta-classifier performance as a baseline. Statistically significant differences are indicated by an asterisk (*) after the corresponding "Merit-0" score.

The Dutch essay (Table A.3), the Spanish (Table A.5), and the Greek article (Table A.6) are the corpora that return the best overall performances (c@1 or Success). Unlike our expectation, the Dutch essay collection, with its short author profile (mean 2 × 398 = 796 word tokens), was not a challenging corpus. The two English collections were more difficult for all attribution schemes. It is difficult to clearly detect general trends. For a given language, the ranking of the systems differs from one genre to the next. The ranking across the genres seems a little bit more stable. For the two article collections (Tables A.5 and A.6), for example, we can find the Statium-l1 or Khonji & Iraqi systems as the best-performing classifiers, followed by Castillo's and Frery's systems. The performance differences, however, are not statistically significant.

TABLE A.1. Evaluation with the English essay (EE) collection (200 problems).

Rank	Run	c@1	Merit-0	Merit-1	Merit-2	Success
1	Frery et al. (2014)	**0.710**	**139.5**	**86.5**	**33.5**	**0.71**
2	*Meta-classifier*	0.680	136	72	8	0.68
3	Khonji and Iraqi (2014)	0.583	116.5*	33.5	−49.5	0.58
4	Castillo et al. (2014)	0.580	116*	32	−52	0.58
5	Spatium-L1	0.577	117.5*	60.5	3.5	0.62
6	Baseline (yes)	0.5	100*	0	−100	0.5

TABLE A.2. Evaluation with the English novel (EN) collection (200 problems).

Rank	Run	c@1	Merit-0	Merit-1	Merit-2	#Success
1	*Meta-classifier*	**0.906**	**87**	**78**	69	0.91
2	Frery et al. (2014)	**0.906**	**87**	**78**	69	0.91
3	Spatium-L1	0.899	80.5	75.5	**70.5**	**0.93**
4	Khonji and Iraqi (2014)	0.844	81	66	51	0.84
5	Castillo et al. (2014)	0.861	82	69	56	0.86
6	Baseline (yes)	0.5	48	0	–48	0.5

TABLE A.3. Evaluation with the Dutch essay (DE) corpus (96 problems).

Rank	Run	c@1	Merit-0	Merit-1	Merit-2	Success
1	*Meta-classifier*	**0.645**	**129**	**58**	**−13**	**0.65**
2	Castillo et al. (2014)	0.615	123	46	−31	0.62
3	Khonji and Iraqi (2014)	0.610	122	44	−34	0.61
4	Frery et al. (2014)	0.588	117.5	35.5	−46.5	0.59
5	Spatium-L1	0.581	116	47	−22	0.59
6	Baseline (yes)	0.5	100	0	−100	0.5

TABLE A.4. Evaluation with the Dutch review (DR) corpus (100 problems).

Rank	Run	c@1	Merit-0	Merit-1	Merit-2	Success
1	Khonji and Iraqi (2014)	**0.650**	**65**	30	−5	**0.65**
2	Spatium-L1	0.621	61.5	**30.5**	**−0.5**	0.64
3	*Meta-classifier*	0.580	58	16	−26	0.58
4	Frery et al. (2014)	0.578	57.5	17.5	−22.5	0.58
5	Baseline (yes)	0.5	50	0	−50	0.5
6	Castillo et al. (2014)	0.370	59	56	53	0.87

TABLE A.5. Evaluation with the Spanish article (SA) collection (100 problems).

Rank	Run	c@1	Merit-0	Merit-1	Merit-2	Success
1	Spatium-L1	**0.866**	**83**	**73**	**63**	**0.88**
2	*Meta-classifier*	0.790	82	64	46	0.82
3	Khonji and Iraqi (2014)	0.778	77.5	55.5	33.5	0.78
4	Castillo et al. (2014)	0.760	76	52	28	0.76
5	Frery et al. (2014)	0.750	75	50	25	0.75
6	Baseline (yes)	0.5	50	0	−50	0.5

TABLE A.6. Evaluation with the Greek article (EA) collection (100 problems).

Rank	Run	c@1	Merit-0	Merit-1	Merit-2	Success
1	Khonji and Iraqi (2014)	**0.810**	**81**	**62**	**43**	**0.81**
2	Spatium-L1	0.785	76.5	57.5	38.5	0.79
3	*Meta-classifier*	0.760	76	52	28	0.76
4	Castillo et al. (2014)	0.730	73	46	19	0.73
5	Frery et al. (2014)	0.642	63.5	30.5	−2.5	0.65
6	Baseline (yes)	0.5	50	0	−50	0.5

Note. The performances of the Spatium-L1 system depicted in the previous tables depend on a random factor, namely, the choice of the impostors. To verify the impact of this selection in the reported performance measures, we show in Table A.7 the c@1 measures based on 500 different choices. In this table, and per test collection, we have indicated the mean, the standard deviation, and the estimated confidence interval covering 95% of the cases. As we can see, the possible variation around the mean performance is relatively small. The reported measures on previous tables are usually closely related to the mean.

TABLE A.7. Variation around the c@1 performance for the Spatium-L1 system.

	c@1		
Test collection	Mean	Standard deviation	Interval (95%)
English Essay (EE)	0.5763	0.0163	[0.5444–0.6083]
English Novel (EN)	0.5889	0.0158	[0.5580–0.6197]
Dutch Essay (DE)	0.8778	0.0143	[0.8498–0.9057]
Dutch Review (DR)	0.6128	0.0198	[0.5739–0.6517]
Spanish Article (SA)	0.8441	0.0196	[0.8057–0.8825]
Greek Article (EA)	0.7917	0.0207	[0.7510–0.8323]

With the English essay corpus, the hardest for our system, Spatium-L1 encounters more difficulties. With the Spanish collection (Table A.5), Spatium-L1 shows high performance levels. In this case, we have longer texts both in the query (mean 1,537 word tokens) and in the proposed author profile (on average 7,685 word tokens).