Detecting authorship deception: a supervised machine learning approach using author writeprints

Lisa Pearl and Mark Steyvers
University of California, Irvine

Abstract

We describe a new supervised machine learning approach for detecting *authorship deception*, a specific type of authorship attribution task particularly relevant for cybercrime forensic investigations, and demonstrate its validity on two case studies drawn from realistic online data sets. The core of our approach involves identifying uncharacteristic behavior for an author, based on a writeprint extracted from unstructured text samples of the author's writing. The writeprints used here involve stylometric features and content features derived from topic models, an unsupervised approach for identifying relevant keywords that relate to the content areas of a document. One innovation of our approach is to transform the writeprint feature values into a representation that individually balances characteristic and uncharacteristic traits of an author, and we subsequently apply a Sparse Multinomial Logistic Regression classifier to this novel representation. Our method yields high accuracy for authorship deception detection on the two case studies, confirming its utility.

Correspondence: Lisa Pearl, Department of Cognitive Sciences, University of California, Irvine, 2314 Social & Behavioral Sciences Gateway Building, Irvine, CA 92697-5100, USA.
Email: lpearl@uci.edu

1 Introduction

Detecting authorship deception can be thought of as a specific type of *authorship attribution* task (Baayen *et al.*, 1996; Diederich *et al.*, 2003; Holmes and Forsyth, 1995; Tweedie *et al.*, 1996 among others), where the basic decision is whether to attribute a given text sample to a given author. This can be particularly relevant for cybercrime forensic investigations where the authorship of a document is in question, and there is little additional evidence to go on beyond the message itself. This process is also sometimes called *authorship identification* (de Vel *et al.*, 2001; Iqbal *et al.*, 2008, 2010; Li *et al.*, 2006; Zheng *et al.*, 2003) or *authorship verification* (Koppel *et al.*, 2009), particularly if there is a suspected author who may be trying to anonymize his/her message or actively imitate another author's writing in order to conceal his/her true identity.

In this article, we describe a new supervised machine learning approach for detecting authorship deception in unstructured text, and show its utility on two case studies drawn from realistic online data sets. The core of our approach involves identifying uncharacteristic behavior for an author, given the document in question, and we do this by extracting a *writeprint* (Abbasi and Chen, 2008; Iqbal *et al.*, 2008, 2010; Li *et al.*, 2006) for the author from known samples of the author's writing. We follow a *computational stylistics* approach (Stamatatos *et al.*, 2000) and draw from both stylometric and content features to define an author's writeprint.

Stylometric features traditionally involve the internal statistics of a document, and are thought to reflect an unconscious 'aspect' (Holmes, 1998) of an author's style, appearing in the form of distinctive, quantifiable features that are salient, structural, and frequent (Bailey, 1979). Given this, an implicit

Literary and Linguistic Computing, Vol. 27, No. 2, 2012.

doi:10.1093/llc/fqs003 Advance Access published on 7 March 2012

assumption is that these features cannot be consciously manipulated (though see Brennan and Greenstadt (2009) for some evidence suggesting this is not true). Some stylometric features investigated previously include lexical features such as function word frequency and vocabulary richness, syntactic features such as passive structures and grammatical category sequences, and structural features such as the format of a signature or paragraph indentation style (Abbasi and Chen, 2008; Baayen *et al.*, 2002; Brennan and Greenstadt, 2009; Burrows, 1987, 1989; de Vel, 2000; Gamon, 2004; Holmes and Forsyth, 1995; Iqbal *et al.*, 2008, 2010; Juola, 2003, 2009; Li *et al.*, 2006; Mosteller and Wallace, 1964; Morton, 1978; Stamatatos *et al.*, 2000; Tweedie *et al.*, 1996).

Content features are based on the semantic content of a message, and are generally less used in authorship attribution tasks since they are seen as more variable, depending on the content the author wishes to express, and so under more conscious control of the author. This would then make them easier to manipulate. Previous studies in authorship attribution have employed content features based on high-frequency domain-specific keywords chosen a priori (Iqbal *et al.*, 2010) and semantic features derived from highly detailed syntactic annotation (Gamon, 2004), with the idea that these semantic features may be less subject to conscious manipulation.

We recognize that feature selection for writeprints is a serious issue (Liu and Motoda, 1998; Iqbal *et al.*, 2008, 2010) and endeavor to incorporate stylometric and content features that are relatively simple to calculate from unstructured text. Because of this, the stylometric features we have chosen are a subset of those explored previously and the content features are derived from *topic models* (Griffiths and Steyvers, 2004; Rosen-Zvi *et al.*, 2004; Steyvers *et al.*, 2004), an unsupervised approach for identifying relevant keywords that relate to the content areas of a document.

Our machine learning approach differs from previous statistical and machine learning techniques used in authorship attribution studies (e.g. simple frequency comparisons (Mosteller and Wallace, 1964), cross entropy (Burrows, 1987), principal components analysis (Burrows, 1989, 2003; Baayen *et al.*, 2002), genetic algorithms (Holmes and Forsyth, 1995; Li *et al.*, 2006), neural networks (Tweedie *et al.*, 1996), support vector machines (Diederich *et al.*, 2003; Gamon, 2004), and linear discriminant analysis (Baayen *et al.*, 2002), among many others—see work by Juola (2003, 2006, 2009) and Koppel *et al.* (2009) for a comparison of different methods). In particular, we represent the features used as input to a machine learning method not with the raw values calculated from the unstructured texts, but rather as a set of feature values that individually balance characteristic and uncharacteristic traits of an author. This has the advantage of distilling the unique components of a particular author's writeprint, such that standout traits are made more salient. We then apply a Sparse Multinomial Logistic Regression (SMLR) classifier (Krishnapuram *et al.*, 2005) to this feature representation, as the SMLR classifier also has the property of identifying a small number of informative writeprint features to base its decision on. This process thus allows the classifier to determine if a document is written by the author in question.

As a demonstration of our approach, we present the results from two practical cases of authorship deception detection. In the first case study, we verify the author of a blog entry from a set of approximately 2,200 blog authors with ten to twenty posts each, derived from the Spinn3r Personal Story Subset (Gordon, 2008). This corresponds to the detection of authors who are attempting to post anonymously or attempting to post under someone else's name, and our approach achieves 89% accuracy using a combination of stylometric and content features. In the second case study, we detect *imitation attacks* (from the Attack Corpus of Brennan and Greenstadt (2009)) on a particular author, where the imitators consciously attempted to alter their normal writing style to match the author's. This corresponds to the detection of online document forgery, and our approach achieves 100% accuracy using stylometric features alone.

In the remainder of this article, we will describe the features of our writeprint implementation, and how they are derived from unstructured text. We will then discuss the machine learning approach

Table 1 Stylometric features available for writeprint characterization

Feature type	Description	#	Implementation	Example calculation
Characters	Letters a, b, c, . . ., z, all digits, all punctuation marks	28	#/(total # character tokens)	(# digits)/(total # letters, digits, and punctuation tokens)
Punctuation marks	?! .; :,	6	#/(total # punctuation tokens)	(#!)/(total # punctuation tokens)
Fine-grained grammatical categories	Part-of-speech tags	37	#/(total # word tokens)	(# *VB*)/(total # word tokens)
Coarse-grained grammatical categories	Nouns, adjectives, verbs, adverbs, function words[a]	5	#/(total # word tokens)	(# nouns)/(total # word tokens)
1st person pronouns	I, me, my, mine, we, us, our, ours	1	#/(total # word tokens)	(# 1st person pronouns)/(total # word tokens)
Lexical diversity[b]	Word type to word token ratio	1	(# word types)/(# word tokens)	Same as implementation
Average sentence length	Average sentence length, based on word tokens in sentence	1	(# words in document)/(# sentences)	Same as implementation
Average word length	Average number of characters in a word document, based on alphabetic word tokens	1	(# letters in document)/(# words in document)	Same as implementation
Total words	Total words	1	total # of words in document	Same as implementation

Note that for all proportion calculations (the first five feature types), a smoothing constant (1) was added to the raw counts.
[a]Nouns consist of tags NN, NNS, NNP, NNPS, PRP, and WP. Adjectives consist of tags JJ, JJR, JJS, PDT, PRP$, and WP$. Verbs consist of tags MD, VB, VBD, VBG, VBN, VBP, and VBZ. Adverbs consist of tags RB, RBR, RBS, and WRB. Function words consist of tags CC, DT, EX, IN, TO, and WDT.
[b]Values range between 0 and 1, with values near 1 indicating more diverse usage (each word type is used only once or twice).

in more detail, and verify its effectiveness on the two realistic case studies mentioned. For each case study, we will describe the data set, how training and test sets were created, and detailed results of the machine learning approach. We will conclude with general discussion of the results, implications for writeprint characterization, and areas of future research.

2 Writeprint Characterization

Our writeprint characterization can include both stylometric and content features. We have nine principle stylometric feature types, corresponding to eighty-one individual features, as shown in Table 1: character distribution, punctuation mark distribution, fine-grained grammatical category distribution, coarse-grained grammatical category distribution, first person pronoun frequency, lexical diversity, average sentence length, average word length, and total words. We note that these are a subset of available stylometric features used in previous studies, and correspond to fairly shallow linguistic information that is easy to extract from unstructured text. In particular, all these features can be extracted directly using a text manipulation programming language such as PERL and a part-of-speech tagger such as the freely available Stanford Log-linear Part-of-Speech Tagger[1] (Toutanova *et al.*, 2003, http://nlp.stanford.edu/software/tagger.shtml).

Our content features consist of topics, which are probability distributions over keywords that relate to a cohesive concept (see Fig. 1 for some sample topics). These topics, and the keywords that comprise them, are identified in an unsupervised fashion using topic models (Griffiths and Steyvers, 2004) from a collection of documents. Without any additional information beyond the documents themselves, topic models can use the words contained in the documents to identify both the topics

Concept ≈ gambling games Topic probability in collection = 0.01541	
Keyword	**Probability of keyword, given topic**
game	0.03981
team	0.01637
games	0.01577
play	0.01560
win	0.01014
poker	0.00961
casino	0.00942

Concept ≈ food Topic probability in collection = 0.02728	
Keyword	**Probability of keyword, given topic**
food	0.01274
eat	0.00830
chicken	0.00751
cream	0.00724
cheese	0.00654
cake	0.00651
chocolate	0.00644

Concept ≈ racing Topic probability in collection = 0.02070	
Keyword	**Probability of keyword, given topic**
run	0.01827
time	0.01704
running	0.01144
bike	0.00784
race	0.00782
started	0.00780
miles	0.00741

Concept ≈ commerce Topic probability in collection = 0.01101	
Keyword	**Probability of keyword, given topic**
10	0.01700
store	0.01508
card	0.01409
item	0.01362
items	0.01344
price	0.01323
money	0.01117

Fig. 1 Sample topics automatically extracted from a collection of blog entries using a topic model. The top seven keywords most associated with each topic are listed from highest to lowest probability. Given these keywords, an interpretation of the concept the topic represents is provided.

expressed in a given document and which topic each word, sentence, or subsection of the document most likely belongs to. We report results from content features based on fifty extracted topics.[2,3] We can also calculate for each author the topics that author is most likely to write about. Figure 2 shows the distribution over topics for some sample authors. The most likely topics in this distribution give a high-level summary of the typical content that is associated with the author, such as *gambling games* and *electronic communication* for the author *poker_star*. We will use the probability distribution over topics for a given author as the set of content features, with the probability of a given topic for that author being the value for that topic's content feature for the author.

3 Application of Machine Learning Techniques

The basic representation of the problem the authorship deception classifier is designed to solve involves a comparison between a document with unknown authorship (the target document) to a document or set of documents from a known author (the reference document(s)). The classifier must decide if the target document is by the same author as the reference document(s). In order to develop the classifier, the following sets of documents are created for each author:

(i) A1 = single randomly chosen target document from the author

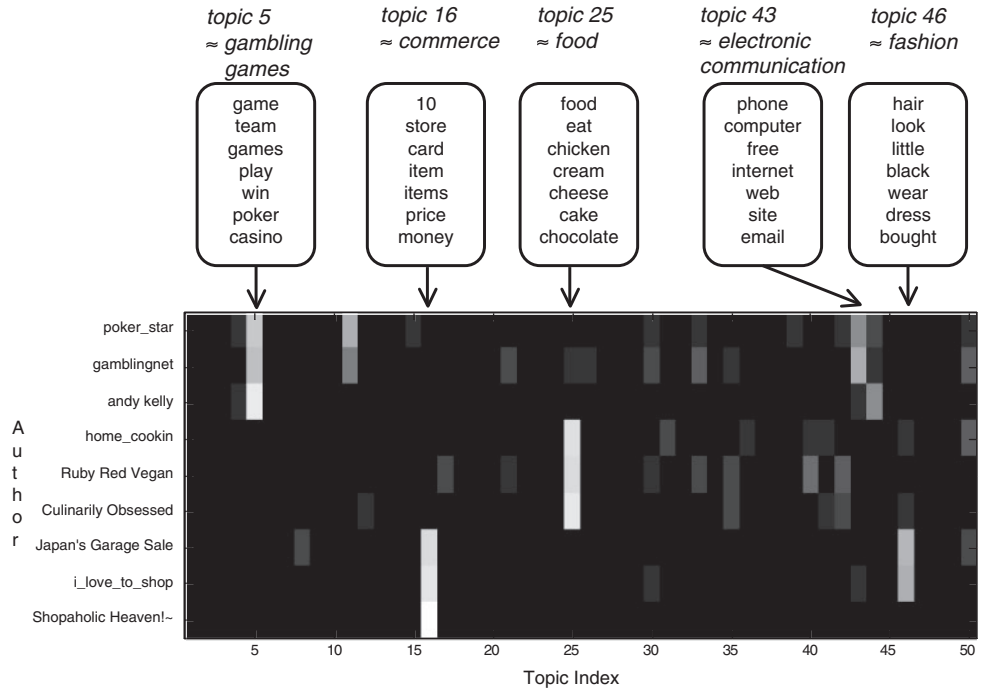

Fig. 2 Sample topic distributions over authors, derived from a collection of blog entries. Lighter shading indicates higher probability, whereas darker shading indicates lower probability. Note that the blog author's user handle sometimes corresponds directly to the topic, such as *poker_star* and *gamblingnet* for the gambling games topic, *home_cookin, Culinarily Obsessed*, and *Ruby Red Vegan* for the food topic, and *Japan's Garage Sale, i_love_to_shop*, and *Shopaholic Heaven!*~ for the commerce topic. Also note that interpretable correlations among topics emerge. For example, authors who write about commerce also tend to write about fashion, and authors who write about gambling games also tend to write about electronic communication.

(ii) A2 = remaining reference document(s) from the author
(iii) X1 = single randomly chosen target document from a different author

The set A2 serves as the reference set for our classifier. These are the known documents that belong to the author. The documents A1 and X1 then serve as target documents that either belong to the original author (A1) or a different author (X1). The training cases for the classifier are not based on single documents but on sets of documents, such that the classifier can learn from the relationships between these sets of documents. In one set, we pair A1 and A2, and based on the input features for this set of documents, the classifier should learn to say that the target document is the *same* author as the original author. In another set, we pair X1 and A2, and the classifier should learn to say that the target document belongs to a *different* author from the original author. We also create test cases for the classifier in order to test the generalization performance. For test cases, we again create pairs (A1, A2) and (X1, A2) but now select authors for the reference documents that do not appear in the training set. By doing this, we can test the classifier's ability to detect authorship deception for new authors.

Instead of applying the classifier to the raw feature values for A1, A2, and X1, we take the additional step of transforming the raw feature values into a more informative representation. Specifically, the classifier examines the target document's value for a given feature, and compares the likelihood of

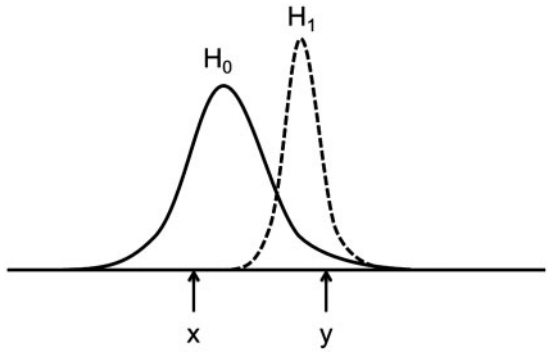

Fig. 3 An example of the log transform of feature values for a particular feature using the distribution over all authors (H_0) and the distribution for a particular author (H_1). The values x and y represent specific values the feature could have.

that value being produced from the general population of authors against the likelihood of that value being produced by the author in the reference documents. This comparison allows the classifier to determine if the target document's feature value is unusual for the reference documents' author.

More specifically, suppose we have feature f for the target document (either A1 or X1). We first log transform all values for f from the entire document collection (including all authors), which creates a distribution of values that are roughly normally distributed, provided the sample size is large enough (see Fig. 3 for an example). We then estimate the best-fitting normal distribution for this observed distribution, and call this the H_0 distribution. This represents the distribution of feature values we expect from the general population of authors. We then apply the same process to the author reference documents alone (A2). In particular, we log transform all values for f from the author reference documents, and then estimate the best-fitting normal distribution for this observed distribution. We call this the H_1 distribution, and this represents the feature distribution for the reference author. The potential difference between the H_0 distribution and H_1 distribution is illustrated by the example in Fig. 3. For that feature, the author's distribution is significantly different from the distribution in the general population, and so this feature could be helpful in identifying that author.

For each feature value f_v in the target document (either A1 or X1), we then calculate the log odds ratio, which compares the probability of f_v coming from the general population's distribution H_0 to the probability of f_v coming from the reference author's distribution H_1:

$$\text{Log odds ratio comparison: } \log\left(\frac{p(f_v|H_0)}{p(f_v|H_1)}\right) \quad (1)$$

A negative value indicates the probability of f_v coming from the author's distribution H_1 is larger than the probability of f_v coming from the general population's distribution H_0, which suggests this feature value is typical for that author and so likely to be from that author. For example, feature value $f_v = y$ in Fig. 3 illustrates this outcome. Conversely, a positive value indicates the probability of f_v coming from the author's distribution H_1 is smaller than the probability of f_v coming from the general population's distribution H_0, which suggests this feature value is atypical for that author and so likely to be from a different author. In Fig. 3, feature value $f_v = x$ illustrates this outcome. It is important to note that the distributions H_0 and H_1 are calculated separately for each feature in the writeprint. Therefore, the model analyzes the diagnosticity of each feature separately. Whether the particular raw feature values are associated with large or small values is irrelevant—what matters is how much evidence each feature value gives about the target author as compared to any randomly chosen author from the population of all authors.

We then apply a SMLR classifier (Krishnapuram *et al.*, 2005) to the log odds ratios for each feature, training it on the data points from the training set and evaluating its performance on the data points from the test set. This kind of regression analysis allows us to identify classifier features that are particularly useful for detecting authorship deception in the training set. In particular, not all features may be useful and this analysis allows us to downweight the features that are less discriminative.

During training, the classifier is provided with the correct classification for each data point, either *same* (1) or *different* (0) (e.g. A1 = 1, X1 = 0).

During testing, the classifier predicts the probability that the target document is from the *same* author as the reference document (e.g. target = example of A1 document with some probability). The classifier can rank test data points by their probability score, which is the probability the classifier believes the target document is not from the author in question (e.g. target documents with low *same* probability are likely to be examples of X1 documents, and so from a different author).

4 Two Practical Demonstrations

We now demonstrate our authorship deception detection classifier on two real world data sets.

4.1 Identifying blog authors

In the first case study, we examine the authorship of blog entries, based on a subset of the Spinn3r Personal Story Subset (Gordon, 2008) from the Spinn3r Blog Dataset (Burton *et al.*, 2009), consisting of approximately 28,500 blog posts from 2,194 unique authors and containing approximately 5.3 million words. This subset included authors who had between ten and twenty blog posts, which ensured a reasonable amount of data from each author. The average length of a blog post from this subset sample was 404 words (excluding punctuation). Authorship deception in this case can be thought of as an author attempting to post anonymously and thus conceal his/her identity, or attempting to post as a different author (perhaps by using that author's username and account). Here, the classifier attempts to decide if a given blog entry belongs to the author in question.

Training and test data sets were constructed, with 75% of the total data used for training and 25% used for testing. This led to 1,646 training authors with 16,460 training cases, and 548 test authors with 5,480 test cases. Note that there was no overlap in authors between training and test, so anything the classifier learned from the training set would be about characterizing author writeprints in general rather than characterizing specific author writeprints.

To construct the writeprint, the classifier used the eighty-one stylometric and fifty content features described in Section 2: character distribution, punctuation mark distribution, fine-grained grammatical category distribution, coarse-grained grammatical category distribution, first person pronoun frequency, lexical diversity, average sentence length, average word length, total words, and fifty topics extracted from the entire blog corpus subset.

As mentioned previously, the SMLR classifier is able to weight features, based on the training set, such that some features are deemed more diagnostic while many are deemed less diagnostic. The thirty most diagnostic writeprint features are shown in Table 2, including both stylometric and content features. Most of these are stylometric, ranging in granularity from individual characters up through document-level characteristics like lexical diversity, though many are at the individual character (e.g. !, *r*, and :) and fine-grained grammatical category (e.g. proper nouns, possessive pronouns, and non-3rd person singular present tense verbs) level. One content feature does get ranked as highly diagnostic, which is the feature based on a topic that likely corresponds to an informal writing style, given the keywords associated with it (e.g. *oh*, *lol*, *yeah)*.

We subsequently applied the SMLR classifier, with its internal ranking of diagnostic features, to the test cases. In order to evaluate the classifier's performance, we calculated true positive rate (TPR) versus false positive rate (FPR), which are standard metrics in signal detection theory. The TPR describes how often the classifier says the target document is from a different author when it really is from a different author (i.e. detecting deception when it is present), whereas the FPR describes how often the classifier says the target document is from a different author when it is really from the same author as the referent documents (i.e. detecting deception when there is none). This can be represented as shown in (2). The goal of the classifier would be to maximize the TPR while minimizing the FPR.

(2) TPR versus FPR calculations

TP (true positive case) = classified as *different* when really *different*

Table 2 The thirty most diagnostic writeprint features identified by the SMLR classifier, after learning from the training set

Feature	Example
! proportion, given all punctuation	Hey**!**
Proper noun proportion, given all fine-grained grammatical categories	**Jack**, **Lily**
Punctuation proportion, given all characters	Hey**,** Jack**!** What's up**?**
Possessive pronouns proportion, given all fine-grained grammatical categories	Is that **your** drink? Yeah, that's **mine**.
Foreign words proportion, given all fine-grained grammatical categories	**Hola, amigo**! What's up?
r proportion, given all characters	Is that you**r** d**r**ink?
Non-3rd-person singular present tense verb forms, given all fine-grained grammatical categories	You **go**. We **stay**. That's how they **roll**.
: proportion, given all punctuation	Dear Jack**:** This is fine.
3rd-person singular present tense verbs forms, given all fine-grained grammatical categories	He **goes**. She **stays**. That's how it **works**.
Plural noun forms, given all fine-grained grammatical categories	Did you see the **penguins**?
, proportion, given all punctuation	Hey Jack**,** did you see that?
Coordinating conjunction proportion, given all fine-grained grammatical categories.	We go **and** you stay.
? proportion, given all punctuation	Hey Jack – what's up**?**
c proportion, given all characters	**C**an you please open the **c**ar door?
Average word length	Average (Hey Jack) = 3.5
Personal pronoun proportion, given all fine-grained grammatical categories	**I** can't go there yet – **you**'ll have to.
; proportion, given all punctuation	. . . there**;** conversely, . . .
h proportion, given all characters	I can't go **th**ere yet – you'll **h**ave to.
1st-person pronoun proportion, given all words	(**I** need to) = 1/3
Lexical diversity	LexDiv('What did he say? What?') = 4/5
Adverbs proportion, given all coarse-grained grammatical categories	We did that pretty **easily**.
s proportion, given all characters	We **s**aid he had to **s**tay.
Past participle proportion, given all fine-grained grammatical categories	We should have **gone**.
p proportion, given all characters	We did that **p**retty easily.
Topic 21	*oh, lol, yeah, people, pretty, fucking, shit, stuff, guy, gonna, love, fun, . . .*
Average sentence length	Average(Hey! Come here!) = 1.5
Past tense proportion, given all fine-grained grammatical categories	We **came**, we **saw**, we **conquered**.
. proportion, given all punctuation	Come here**.** We want to see**.**
Present participle proportion, given all fine-grained grammatical categories	We're **going** now.
wh-adverb proportion, given all fine-grained grammatical categories	**How** can we do this? **Where** can we go?

Features are ranked from most diagnostic to least diagnostic, with examples of the salient part of the feature provided.

FP (false positive case) = classified as *different* when really *same*
TN (true negative case) = classified as *same* when really *same*
FN (false negative case) = classified as *same* when really *different*

TPR versus FPR
$TPR = TP/(TP + FN)$
$FPR = FP/(FP + TN)$

Given how these metrics are defined, there are tradeoffs between TPR and FPR. For example, it is easy to get a 100% TPR by classifying every document as from a *different* author—however, this means that the FPR will also be quite high, which is bad. The receiver operating characteristic (ROC) curve in Fig. 4 shows the tradeoff between TPR and FPR for the classifier on the test data from the blog data set. The area under the ROC curve (abbreviated as AUC in Fig. 4) represents the average probability of the classifier making the correct classification across all true positive/false positive thresholds.

Table 3 shows more detailed results for the classifier on this data set, including how well the

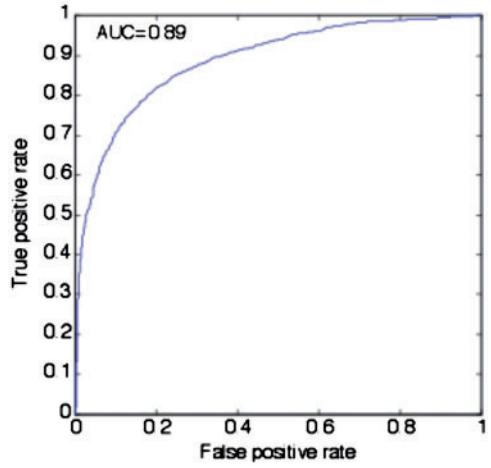

Fig. 4 ROC curve for authorship deception classifier performance on the blog data set, using a writeprint consisting of stylometric and content features. AUC is shown.

Table 3 Classifier performance on the blog data set, given writeprints comprised of different transformed features

Classifier performance	**AUC**	**TPR**	**FPR**
SMLR writeprint features (transformed values)			
81 stylometric features	0.88	0.80	0.20
50 content features	0.83	0.76	0.25
81 stylometric + 50 content features	0.89	0.81	0.19
Comparison machine learning methods, using 81 stylometric features (raw values)		**TPR**	**FPR**
(KNN)—50 neighbors		0.69	0.28
Decision Tree		0.58	0.42
Mean Margins Tree		0.65	0.36
Naïve Bayes		0.50	0.50
Naïve Instance—50 neighbors		0.50	0.50

AUC is provided, as well as the best balance of TPR and FPR for the SMLR classifier. In addition, the results of several other machine learning methods that use the same stylometric features (but raw feature values rather than transformed values) as well as the same training and test sets are provided for comparison.

classifier does using writeprints constructed from only stylometric and only content features. In addition, Table 3 provides the results of several existing machine learning methods on this same data set as a baseline for performance, but using only the raw values for the writeprint features, rather than the transformed feature values. These additional methods were run using the freely available Waffles toolkit.[4] In particular, for each comparison machine learning method, the training set consisted of pairs of documents, one target document (A1) and one referent document (X1 or A2), along with a classification of 'same' or 'different' for each document pair. The test set was similarly constructed.

From these results, we can see quite good performance from our classifier. Using writeprints made of both stylometric and content features that were transformed, our classifier will identify a deceptive author (i.e. a different author from the reference author) 89% of the time on average, with a TPR of 81% and a FPR of 19%. This is a notable improvement over the best-performing comparison machine learning method (the KNN algorithm with 50 neighbors) operating over raw feature values, which had a TPR of 69% and a FPR of 28%. Interestingly, it appears that the contributions of stylometric and content features in the writeprint are not additive. Stylometric features on their own achieve correct classification 88% of the time while content features on their own achieve correct classification 83% of the time. This suggests that stylometric features on their own can be very distinctive of an author, irrespective of the content of an author's message. However, we also find that sophisticated content features can be fairly distinctive on their own, even if the best performance is found by combining stylometric and content features. We note that the high performance of a classifier using stylometric features only in the writeprint is useful since the topic models that are used to generate the content features require a fairly large number of writing samples (on the order of thousands). This is relevant for situations where we do not have a large quantity of data to work with, as in the next case study.

4.2 Identifying imitation attacks

The second case study we examined used data gathered from writers who specifically tried to conduct imitation attacks on an existing author (Brennan and Greenstadt, 2009). Authorship deception in this case can be thought of as an author attempting

to impersonate another author and write a message as that author, concealing his/her own true identity. We used a subset of the Attack Corpus (Brennan and Greenstadt, 2009) that contained a writing sample from one author with a fairly distinctive style (writer Cormac McCarthy) and twelve imitation attacks by writers who saw this sample and attempted to mimic the original author's style. The attacks ranged in length from 478 to 521 words, with an average length of 497 words. The McCarthy sample they used as a basis for their imitation was 2,541 words long. We obtained 24 additional samples from Cormac McCarthy to supplement the set of author reference documents, for a total of 80,262 words (with an average of 3,210 words per sample).

Because this data set was too small to extract topics as content features, we used a writeprint that consisted only of the 81 stylometric features used in the first case study. Training and test sets were constructed, with 70% of the writing samples used for training and 30% of the writing samples used for testing. There was no overlap in the imitation samples between training and test. This led to a training set consisting of 80 different author cases (X1 versus A2) and 50 same author cases (A1 versus A2), and a test set consisting of 40 different author cases (X1 versus A2) and 70 same author cases (A1 versus A2).

The results of our classifier were excellent, with the classifier achieving perfect detection of imitation attacks, as shown in Table 4: 100% TPR and 0% FPR. This is a substantial improvement over the results found with state-of-the-art methods tested by Brennan and Greenstadt (2009), such as the Signature stylometric system, neural networks, and a synonym-based classifier, which had average accuracy scores below 5%, and a highest accuracy score around 10%. In addition, Table 4 provides the results of several existing machine learning methods on this same data set as a baseline for performance, but using only the raw values for the writeprint features, rather than the transformed feature values. These additional methods were run using the freely available Waffles toolkit. In particular, for each comparison machine learning method, the training set consisted of pairs of documents, one target document (A1) and one referent document (X1 or A2), along with a classification of 'same' or 'different' for each document pair. The test set was similarly constructed.

Table 4 Classifier performance on the Attack Corpus data set, given writeprints comprised of transformed stylometric features.

Classifier performance	**TPR**	**FPR**
SMLR writeprint features (transformed values)		
81 stylometric features	1.00	0.00
Comparison machine learning methods, using 81 stylometric features (raw values)	**TPR**	**FPR**
KNN—50 neighbors	1.00	0.00
Decision Tree	1.00	0.00
Mean Margins Tree	1.00	0.00
Naïve Bayes	1.00	0.00
Naïve Instance—50 neighbors	1.00	0.00

The best balance of TPR and FPR is provided for the SMLR classifier. In addition, the results of several other machine learning methods that use the same stylometric features (but raw feature values rather than transformed values) as well as the same training and test sets are provided for comparison.

Surprisingly, we found that *all* of the machine learning methods we applied to this corpus using the 81 stylometric features gave this same excellent performance—whether the feature values were transformed (as in the case of the SMLR classifier) or not (as in the case of the rest of the machine learning methods). This may be due to the diagnostic nature of some of the stylometric features—it turned out that many of the stylometric features were able to individually distinguish true author samples from imitations, such as lexical diversity, part-of-speech usage, and average sentence length. Figure 5 shows the transformed feature values for lexical diversity, average sentence length, proportion of adjectives, and proportions of verbs when comparing writing samples from the original author, McCarthy, with imitation writing samples and additional normal writing samples available for imitators from the Attack Corpus (there were 63,000 words in the normal writing samples, with an average length of 500 words per sample). Notably, despite the imitators' conscious attempts to mimic McCarthy's style, their lexical diversity and adjective

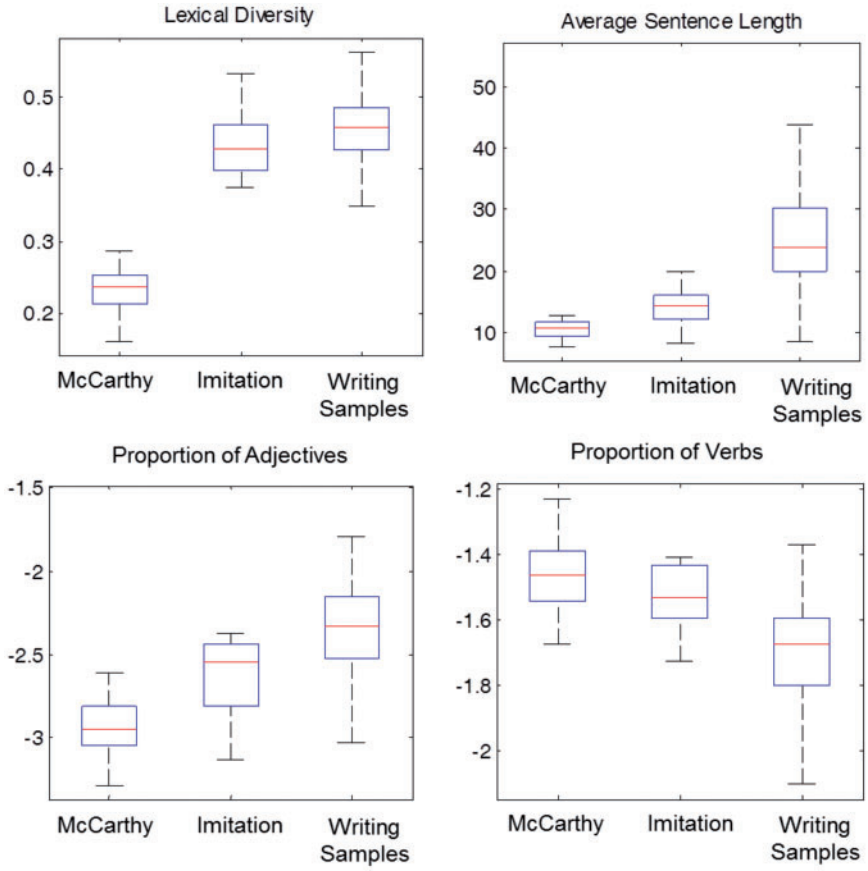

Fig. 5 An illustration of several stylometric feature distributions for McCarthy's writing samples, the imitation attack samples, and the imitators' normal writing samples. The line in the center of each box indicates the median value, while the boxes indicate the 25th and 75th percentiles, and the dashed lines indicate the full range of values (excluding outliers).

usage was much higher and therefore more similar to their normal writing style. Similarly, Fig. 5 shows that even though verb usage and average sentence length did change in the imitation condition (and so were more subject to conscious manipulation), the imitators did not change their style sufficiently to successfully mimic McCarthy's style.

Both these case studies on realistic authorship data demonstrate the effectiveness of our approach for detecting authorship deception. In each case, our classifier has yielded excellent results, using write-prints based either on both stylometric and content features, or stylometric features alone. Notably, while the particular representation of features that we propose is not always necessary to achieve good performance when paired with state-of-the-art machine learning methods (as in the case of the imitation attacks), it *can* significantly improve performance in some more challenging cases (as in the blog entries). In particular, we feel the blog data set was likely more challenging due to the diversity of authors used as target authors—compared with the target author in the imitation attack data set (Cormac McCarthy), who had a markedly distinctive writing style, the target authors for the blog data set were less likely to all have such distinctive styles.

This can make them more difficult to identify as distinct from each other, and as such, may represent a more realistic authorship deception scenario.

5 Discussion and Conclusion

In this article, we have described a new supervised machine learning approach to detecting authorship deception, involving a novel representation of writeprint feature values that is paired with a classifier which bases its decisions on highly informative features that it identifies from the training set. Importantly, this approach can be combined with a writeprint characterized by any kind and number of features, as demonstrated in the two case studies, where the writeprints were defined differently in each. Moreover, the feature representation may be combined with a number of different machine learning methods to achieve very good performance.

The characterization of an author's writeprint is one of the keys to successful authorship deception detection. Here, we have identified a more sophisticated kind of content feature for writeprints, based on topic models, that may be successfully integrated into a writeprint characterization. Interestingly, we found that these topic-based features are highly successful at characterizing an author on their own, though they also improve the classifier's performance when integrated into a writeprint containing stylometric features.

Incorporating more sophisticated features that combine stylometry and content is an important area for writeprint research. For example, some features worth considering, particularly in the realm of online communication where messages may be created in a more fluid manner similar to conversational patterns, are distinctive capitalization patterns and emoticon patterns (e.g. *i'm* versus *I'm*; :) versus :-D) and distinctive synonym usage (e.g. *Daddy* versus *Dad*; *heya* versus *hi*; *fabulous* versus *great*; *heck* versus *hell*). Some of these features may be more or less difficult to consciously manipulate, which allows us to gauge their utility in author writeprints.

To conclude, we believe that this study highlights how stylometrics, computational linguistics, and machine learning can be combined to yield informative writeprints for authorship deception detection, and that this provides a useful basis for future cybercrime forensic investigations.

References

Abbasi, A. and Chen, H. (2008). Writeprints: a stylometric approach to identity-level identification and similarity detection in cyberspace. *ACM Transactions on Information Systems*, **26**(2): 1–29.

Bailey, R. W. (1979). Authorship attribution in a forensic setting. In Ager, D. E., Knowles, F. E., and Smith, J. (eds), *Advances in Computer-aided Literary and Linguistic Research*. Birmingham: AMLC, pp. 1–15.

Baayen, R., Van Halteren, H., and Tweedie, F. (1996). Outside the cave of shadows: using syntactic annotation to enhance authorship attribution. *Literary and Linguistic Computing*, **2**: 110–20.

Baayen, R., van Halteren, H., Neijt, A., and Tweedie, F. (2002). An experiment in authorship attribution. In *Proceedings of JADT 2002,* St. Malo. Université de Rennes, pp. 29–37.

Brennan, M. and Greenstadt, R. (2009). Practical attacks against authorship recognition techniques. In *Hacking At Random 2009 Conference*. The Netherlands: Vierhouten. http://www.cs.drexel.edu/~mb553/stuff/brennan_iaai09.pdf (accessed 9 February 2012).

Burrows, J. F. (1987). Word patterns and story shapes: the statistical analysis of narrative style. *Literary and Linguistic Computing*, **2**: 61–70.

Burrows, J. F. (1989). 'an ocean where each kind . . .': statistical analysis and some major determinants of literary style. *Computers and the Humanities*, **23**(4–5): 309–21.

Burrows, J. F. (2003). Questions of authorships: attribution and beyond. *Computers and the Humanities*, **37**(1): 5–32.

Burton, K., Java, A., and Soboroff, I. (2009). The ICWSM 2009 Spinn3r Dataset. In *Proceedings of the Third Annual Conference on Weblogs and Social Media (ICWSM 2009)*. San Jose, CA. http://www.icwsm.org/data/ (accessed 10 February 2012).

de Vel, O. (2000). Mining e-mail authorship. In *Proceedings of the Workshop on Text Mining in ACM International Conference on Knowledge Discovery and Data Mining (KDD)*. http://www.cs.cmu.

edu/~dunja/KDDpapers/DeVel_TM.pdf (accessed 10 February 2012).

de Vel, O., Anderson, A., Corney, M., and Mohay, G. (2001). Mining E-mail content for author identification forensics. *SIGMOD Record*, **30**(4): 55–64.

Diederich, J., Kindermann, J., Leopold, E., and Paass, G. (2003). Authorship attribution with support vector machines. *Applied Intelligence*, **19**: 109–23.

Gamon, M. (2004). Linguistic correlates of style: authorship classification with deep linguistic analysis features. In *Proceedings of the 20th International Conference on Computational Linguistics*. Geneva: Switzerland. http://research.microsoft.com/apps/pubs/default.aspx?id=68964 (accessed 10 February 2012).

Gordon, A. (2008). Story management technologies for organizational learning. *Special Track on Intelligent Assistance for Self-Directed and Organizational Learning*. Austria: International Conference on Knowledge Management Graz. http://ict.usc.edu/files/publications/IKNOW08.PDF (accessed 10 February 2012).

Griffiths, T. and Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences*, **101**: 5228–35.

Holmes, D. and Forsyth, R. S. (1995). The *Federalist* revisited: new directions in authorship attribution. *Literary and Linguistic Computing*, **10**: 111–27.

Holmes, D. (1998). The evolution of stylometry in humanities scholarship. *Literary and Linguistic Computing*, **13**: 111–17.

Iqbal, F., Hadjidj, R., Fung, B., and Debbabi, M. (2008). A novel approach of mining write-prints for authorship attribution in e-mail forensics. *Digital Investigation*, **5**: 42–51.

Iqbal, F., Binsalleeh, H., Fung, B., and Debbabi, M. (2010). Mining writeprints from anonymous e-mails for forensic investigation. *Digital Investigation*, **7**: 56–64.

Juola, P. (2003). The time course of language change. *Computers and the Humanities*, **37**(1): 77–96.

Juola, P. (2006). Questioned electronic documents: empirical studies in authorship attribution. In Olivier, M. S. and Shenoi, S. (eds), *Research Advances in Digital Forensics II*. Heidelberg: Springer. http://www.mathcs.duq.edu/~juola/papers.d/forensics.pdf (accessed 10 February 2012).

Juola, P. (2009). 20,000 ways not to do authorship attribution – and a few that work. In *Proceedings of the International Association of Forensic Linguist. Chicago Colloquium on Digital Humanities and Computer Science*. http://www.mathcs.duq.edu/~juola/papers.d/20Kways.txt (accessed 10 February 2012).

Koppel, M., Schler, J., and Argamon, S. (2009). Computational methods in authorship attribution. *Journal of American Society Information Science Technology*, **60**(1): 9–26.

Krishnapuram, B., Figueiredo, M., Carin, L., and Hartemink, A. (2005). Sparse multinomial logistic regression: fast algorithms and generalization bounds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**: 957–68.

Li, J, Zheng, R, and Chen, H. (2006). From fingerprint to writeprint. *Communications of the ACM*, **49**(4): 76–82.

Liu, H. and Motoda, H. (1998). *Feature Selection for Knowledge Discovery and Data Mining*. Norwell, MA: Kluwer Academic Publishers.

Morton, A. Q. (1978). *Literary Detection*. New York: Scribners.

Mosteller, F. and Wallace, D. L. (1964). *Applied Bayesian and Classical Inference: The Case of the Federalist Papers*. Reading, MA: Addison-Wesley.

Rosen-Zvi, M., Griffiths, T., Steyvers, M., and Smyth, P. (2004). The author-topic model for authors and documents. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*. Arlington, VA: AUAI Press, pp. 487–94.

Stamatatos, E, Kokkinakis, G, and Fakotakis, N. (2000). Automatic text categorization in terms of genre and author. *Computational Linguistics*, **26**(4): 471–95.

Steyvers, M., Smyth, P., Rosen-Zvi, M., and Griffiths, T. (2004). Probabilistic author-topic models for information discovery. In Kim, W., Kohavi, R., Gehrke, J., and DuMouchel, W. (eds), *The Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Seattle, Washington: ACM, pp. 306–15.

Toutanova, K., Klein, D., Manning, C., and Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of HLT-NAACL 2003*, Edmonton, Canada: HLT-NAACL, pp. 252–9.

Tweedie, F., Singh, S., and Holmes, D. I. (1996). An introduction to neural networks in stylometry. *Research in Humanities Computing*, **5**: 249–63.

Zheng, R., Qin, Y., Huang, Z., and Chen, H. (2003). Authorship analysis in cybercrime investigation. In *Proceedings of the 1st International Symposium on Intelligence and Security Informatics (ISI)*. Tucson,

Arizona. http://www.mendeley.com/research/authorship-analysis-in-cybercrime-investigation/ (accessed 10 February 2012).

Notes

1 The list of tags from the Stanford Part-of-Speech tagger used is as follows, with an example of each tag in parentheses: CC = coordinating conjunction (***and***), CD = cardinal number (***one*** *penguin*), DT = determiner (***the***), EOS = end of sentence marker (there's a penguin here**!**), EX = existential there (***there****'s a penguin here*), FW = foreign word (***hola***), IN = preposition or subordinating conjunction (***after***), JJ = adjective (***good***), JJR = comparative adjective (***better***), JJS = superlative adjective (***best***), LS = list item marker (***one, two, three,*** . . .), MD = modal (***could***), NN = singular or mass noun (***penguin, ice***), NNS = plural noun (***penguins***), NNP = proper noun (***Jack***), NNPS = plural proper noun (*There are two* ***Jacks****?*), PDT = predeterminer (***all*** *the penguins*), POS = possessive ending (*penguin****'s***), PRP = personal pronoun (***me***), PRP$ = possessive pronoun (***my***), RB = adverb (***easily***), RBR = comparative adverb (***later***), RBS = superlative adverb (***most*** *easily*), RP = particle (*look it* ***up***), SYM = symbol (this = that), **TO** = infinitival to (*I want* ***to*** *go*), UH = interjection (***oh***), VB = base form of verb (*we should* ***go***), VBD = past tense verb (*we* ***went***), VBG = gerund or present participle (*we are* ***going***), VBN = past participle (*we should have* ***gone***), VBP = non-3rd person singular present tense verb (*you* ***go***), VBZ = 3rd singular present tense verb (*he* ***goes***), WDT = wh-determiner (***which*** *one*), WP = wh-pronoun (***who***), WP$ = possessive wh-pronoun (***whose***), WRB = wh-adverb (***how***).

2 The topic model is implemented using a hierarchical Bayesian model, and Gibbs sampling is used to converge on the topics (see Griffiths and Steyvers (2004) for details). We use model parameters $\alpha = 0.1$ and $\beta = 0.001$, with 250 iterations, four chains, and a single sample saved at the end of each chain.

3 Note that we also looked at using 100 and 200 extracted topics, but found that results with fifty were better, perhaps due to the quantity of data available. In addition, the topics were more easily interpretable when we used fifty topics, as opposed to a higher number of topics.

4 http://waffles.sourceforge.net/