Authorship Attribution for Social Media Forensics

Anderson Rocha, *Senior Member, IEEE*, Walter J. Scheirer, *Senior Member, IEEE*, Christopher W. Forstall, Thiago Cavalcante, Antonio Theophilo, Bingyu Shen, Ariadne R. B. Carvalho, and Efstathios Stamatatos

***Abstract*— The veil of anonymity provided by smartphones with pre-paid SIM cards, public Wi-Fi hotspots, and distributed networks like Tor has drastically complicated the task of identifying users of social media during forensic investigations. In some cases, the text of a single posted message will be the only clue to an author's identity. How can we accurately predict who that author might be when the message may never exceed 140 characters on a service like Twitter? For the past 50 years, linguists, computer scientists, and scholars of the humanities have been jointly developing automated methods to identify authors based on the style of their writing. All authors possess peculiarities of habit that influence the form and content of their written works. These characteristics can often be quantified and measured using machine learning algorithms. In this paper, we provide a comprehensive review of the methods of authorship attribution that can be applied to the problem of social media forensics. Furthermore, we examine emerging supervised learning-based methods that are effective for small sample sizes, and provide step-by-step explanations for several scalable approaches as instructional case studies for newcomers to the field. We argue that there is a significant need in forensics for new authorship attribution algorithms that can exploit context, can process multimodal data, and are tolerant to incomplete knowledge of the space of all possible authors at training time.**

***Index Terms*— Authorship attribution, forensics, social media, machine learning, computational linguistics, stylometry.**

I. Introduction

IT IS well known that the real lives of Internet users sometimes turn out to be entirely different from who they appear to be online, but the nature and consequence of this phenomenon are changing. A recent exposé in the *New York Times Magazine* [41] documented the case of a Russian media agency that allegedly executed organized disinformation campaigns on social media using pseudonyms and virtual identities. It is assumed that some of these campaigns were state sponsored. With an office full of media professionals, the agency achieved success in promoting false news events and influencing public opinion on politics, and was even able to deceive the journalist covering the story for the *Times*.

On the Internet, this practice is known as "trolling" — a favorite pastime of bored adolescents, pundits, and unscrupulous social media coordinators. The organization and scale of these trolling campaigns, however, suggests that the practice has moved into a new phase, whereby corporations and governments seek to control the discourse surrounding popular events (both real and imagined) on social media. This poses a legal and security dilemma on multiple fronts. If the underlying identities of the Russian media agency's employees could be automatically determined, content originating from them could subsequently be flagged and monitored or blocked. However, the *Times* discovered that the agency always routed its Internet traffic through proxy servers, thus rendering useless the easy path to doing so via the originating IP addresses.

A. Motivation for Automated Authorship Attribution Methods for Social Media Forensics

The enabling factor in the above example is a reliance on anonymity to ensure the success of a social media campaign. How have recent changes in technology altered the landscape for anonymous social media use? The proliferation of unlocked smartphones and pre-paid SIMs from vending machines has made relatively anonymous mobile Internet use a reality [153]. Similarly, public wi-fi hot spots are a favorite point of access for illicit activity; it is simple to step into a coffee shop with a laptop and blend into the crowd [69]. Many users concerned about privacy are turning to the Tor service [48], an Onion Routing network [154] that encrypts user data and randomly sends it back and forth through various nodes for anonymization. Finally, like the Russian media agency, one can simply tunnel traffic through a series of proxy servers, many of which are freely available and open to the public.[1] None of these strategies is foolproof, but when two or more are combined by a very careful individual, they can frustrate an investigation to the point where network forensics cannot be used. In such cases, the text left on a social media platform may be our only clue to the author's identity.

Manuscript received January 18, 2016; revised May 23, 2016 and August 5, 2016; accepted August 21, 2016. Date of publication August 29, 2016; date of current version October 31, 2016. The work of A. Rocha, A. Theophilo, T. Cavalcante, and A. R. B. Carvalho was supported in part by the Brazilian Coordination for the Improvement of Higher Education and Personnel — CAPES (DeepEyes project), and in part by the São Paulo Research Foundation – FAPESP (through the DéjàVu Project under Grant 15/19222-9). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Mauro Barni. *(Anderson Rocha and Walter J. Scheirer contributed equally to this work.) (Corresponding author: Anderson Rocha.)*

A. Rocha, T. Cavalcante, and A. R. B. Carvalho are with the Institute of Computing, University of Campinas, Campinas 13083-852, Brazil (e-mail: anderson.rocha@ic.unicamp.br; thicosc@gmail.com; ariadne@ic.unicamp.br).

W. J. Scheirer, C. W. Forstall, and B. Shen are with the Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556 USA (e-mail: walter.scheirer@nd.edu; cforstall@gmail.com; bshen@nd.edu).

A. Theophilo is with the Institute of Computing, University of Campinas, Campinas 13083-852, Brazil, and also with the Center for Information Technology Renato Archer, Campinas 13069-901, Brazil (e-mail: antonio.theophilo@gmail.com).

E. Stamatatos is with the Department of Information and Communication Systems Engineering, University of the Aegean, 83200 Karlovasi, Greece (e-mail: stamatatos@aegean.gr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2016.2603960

[1] https://incloak.com/proxy-list/

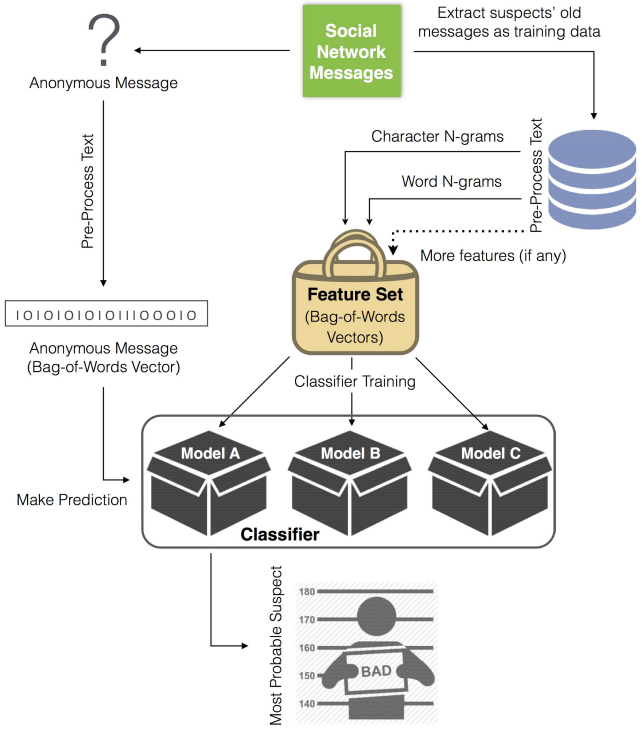


Fig. 1. Forensic authorship attribution is the process of inferring something about the characteristics of an author from the form and content of their writing present in a collection of evidence [89]. The emergence of social media as a primary mode of communication has challenged the traditional assumption that a forensic investigation will have access to long-form writing (*i.e.*, letters and emails). In this article, we frame the problem as a computational pipeline, in which features are extracted from very small samples of text, and scalable supervised learning is deployed to train author-specific models and make predictions about unknown samples.

The goal of *Authorship Attribution* is to identify authors of texts through features derived from the style of their writing; this is called *Stylometry*. In the context of a criminal investigation, this endeavor is termed *Forensic Authorship Attribution* (Fig. 1). In contrast to other authorship attribution tasks found in the broader world of signal processing like active authentication [61], [62], forensic authorship attribution does not assume a claimed identity before analysis to verify a match. Instead, it is assumed that the source of a text is either one author (or possibly several) out of a known set, or an author that is unknown to investigators.

Automated approaches to authorship attribution via the methods of statistical pattern recognition have been around for decades [28], [52], [140], with more recent work evaluating the utility of advanced machine learning techniques [50], [87]. Authorship attribution as an academic discipline has maintained a unique relationship to scholarship in the digital humanities, where the authentication of disputed literary works is of interest. For that reason, much of the existing work in this area remains unfamiliar to practitioners and researchers in forensics at large, in spite of excellent attempts at outreach (*e.g.*, see Juola [89], Koppel et al. [108], and Stamatatos [175]). However, the relationship between the work of humanists and forensic examiners cannot be denied: both groups seek to make accurate predictions about uncertainties related to textual data.

Notwithstanding, the underlying problem domains humanists and forensic examiners operate in can be rather different. The longer the text is, the easier it is to compute stylometric features, which become more reliable as more text is considered. For instance, a novel provides an investigator with a wealth of information from which to extract stylistic clues. But we face the opposite scenario when examining messages from social media, where texts are very short and therefore a smaller set of stylometric features is present in each one. In response to this, some researchers suggest joining the messages in a single document [139]. Even with preliminary results showing some improvement, this is not realistic since we may not have more than one message we wish to know the author of [166], and whenever dealing with anonymous messages we cannot guarantee that all of them belong to the same author. However, we know that there is often enough distinct information in even just a handful of sentences for a human reader to understand that they are from a common source. For instance, consider the following tweets from two prominent Twitter personalities. These from Author A:

> A.1: A beautiful reflection on mortality by a great man.
> A.2: Unintentional reductio ad absurdum: "Colleges Need Speech Codes Because Their Students Are Still Children"
> A.3: The great taboo in discussions of economic inequality: Permanent income has nonzero heritability.

And these from Author B:

> B.1: Incredible wknd w/ @CASAofOC. Thx to all of the folks that helped raise $2.8 million to help young people in need.
> B.2: Thx 4 having me. Great time w/ you all
> B.3: Great to meet you & thx for your support @ChrissieM10

When reading these tweets, there can be no doubt that Author A and Author B are different people. The first set of tweets comes from the experimental psychologist and popular science writer Steven Pinker.[2] A notable stylist, Pinker's tweets tend to be well composed and include diverse vocabulary ("mortality," "reductio ad absurdum," "taboo," "heritability") even under the 140 character constraint. The second set comes from Philadelphia Eagles quarterback Mark Sanchez,[3] who deploys a more colloquial style of writing, including frequent use of abbreviation ("Thx," "wknd"). Such traits can be incorporated into a model that is learned from whatever limited information is at hand.

That is not to say further complications do not exist when considering social media. Very different from essays, books, and even signals from the audio domain, postings to social media include irregular language usage common to the Internet such as overemphasized punctuation, emoticons,[4]

[2]https://twitter.com/sapinker
[3]https://twitter.com/mark_sanchez
[4]The ubiquitous faces made with ASCII symbols such as: ":-)", ":'(", ":-D" and ">:("

Internet expressions[5] and a conspicuous lack of spell-checking. On the surface, these characteristics appear to be a rich source of additional style markers. Notwithstanding, they also pose a challenge, because punctuation and grammatical usage, emoticons, and expressions are constantly evolving on the Internet, bringing any assumption of authorial consistency on the part of algorithm designers into question. To adapt authorship attribution to social media, we need to find stylometric features that capture the diversity of the language deployed therein. Such features will generally be sparse in nature, generating vectors of high dimensionality. Given that the messages are short, this task requires a large amount of training data to increase the classification accuracy, as well as fast and scalable classifiers to process the number of features that are generated.

It must be emphasized that even the most promising techniques in authorship attribution are not nearly as precise as DNA testing, and forensic stylometry will rarely, if ever, be used in a courtroom by itself.[6] Instead, we find that these techniques are used to give direction in criminal investigations, possibly by narrowing a search down to a few suspects or backing up evidence against a particular suspect. Our working hypothesis is that even with social media postings being much smaller than books or essays, it is still possible to perform authorship attribution that will be admissible in court by using a dynamic set of features, adaptable to each group of users in a particular social media ecosystem.

B. Contributions of This Review Article

In line with the above motivation, we present the following material in this article:

1) An overview of forensic authorship attribution, with a discussion of why it is distinct from more general authorship attribution applications found in audio, speech and language processing.
2) A comprehensive review of the methods for authorship attribution that are applicable to forensics in a social media context.
3) An analysis of feature sets that can extract a large amount of information from a small amount of text, which is crucial for achieving good performance in a forensic setting.
4) A detailed walk-through of supervised learning methods for authorship attribution for social media forensics based on several recent strategies from the literature.
5) A discussion of open problems in forensic authorship attribution.

To outline the rest of this article: a critical review of existing methods for authorship attribution and how they relate to forensic authorship attribution and social media is provided in the next section (Sec. II). Thereafter, we develop an instructional example for the reader that begins with a set of features used by most techniques in the literature (Sec. III), and goes on to evaluate these features using various classification approaches applied to a new data set of tweets collected for this article (Sec. IV). Finally, we conclude (Sec. V) by calling the attention of the information forensics community to the many research opportunities available and the scientific challenges that are waiting to be addressed in this exciting research area.

[5]Common examples are: lol = laugh out loud; brb = be right back; imho = in my humble opinion.

[6]Recent cases involving linguistic evidence have made their way into courtrooms around the globe [45], [66].

II. A Review of Methods in Authorship Attribution Relevant to Social Media Forensics

As a distinct discipline within stylometry at-large, forensic authorship attribution is different from more general authorship attribution applications found in audio, speech and language processing. Its defining characteristics are both domain and process dependent, yielding challenges that at present have not been fully addressed [89]. One such general attribution application is Authorship Verification [14], [113], which has received a considerable attention recently due to the PAN evaluations [91], [177], [178]. Authorship verification is a 1:1 classification task with a known positive class, and a negative class of "all texts by all other authors" that is vast and extremely heterogeneous. In contrast, forensic authorship attribution rarely, if ever, presents us with a known positive class from the outset. We more often are faced with a problem of 1:N identification: one unknown author and many potential known authors to match against. Further, unlike authorship in the digital humanities [81], [99], [120], the question of how to falsify a prediction is important for the admissibility of evidence. While it may be of no practical consequence to ever definitively prove or disprove that Shakespeare authored a disputed text in a lively scholarly debate, we do not have that luxury in forensics. In all cases of forensic authorship attribution, how we treat the task is instrumental to the design of appropriate algorithms.

The key considerations for forensic authorship attribution are:

- No control over the testing set that predictions will be made from, which could be limited to a single sample.
- No control over the quality of the training data used to create the attribution classifiers (in most circumstances). The training regime must be tolerant to some measure of noise, and a variable number of samples across known authors.
- The need for a well-defined process. This is necessary for accurate and efficient algorithms, as well as legal consideration.
- The determination of a well-defined error rate for an algorithm, before it is applied to a real-world problem. This is necessary to understand if the probability of making a mistake is too large for legal purposes.
- The potential for adversaries. It is possible that the author under investigation is deliberately evading automated attribution methods.

In light of the above, what are the components we need to build a computational pipeline that is suitable for forensic authorship attribution for social media analysis? The setup we consider in this article is the following:

TABLE I

FUNCTION WORDS ARE A VERY BASIC, BUT SOMETIMES USEFUL FEATURE FOR AUTHORSHIP ATTRIBUTION IN ALL CONTEXTS. THEY CAN BE PARTICULARLY EFFECTIVE FOR SOCIAL MEDIA ANALYSIS, BECAUSE THEY TEND TO BE THE WORDS THAT OCCUR MOST FREQUENTLY. THUS THE PROBABILITY OF OCCURRENCE IN EVEN SMALL SAMPLES LIKE TWEETS IS HIGH

Part of Speech	Example English Function Words
Articles	the, a, an, some
Pronouns	I, we, he, she, they
Prepositions	in, under, towards, before
Conjunctions	and, or, nor
Auxiliary Verbs	be, can, could, ought
Interjections	hi, bye, well, oh
Particles	to, off, on, up
Expletives	there, it

- A training corpus drawn from postings to social media.
- A pre-processing regime that filters (*e.g.*, removes empty samples or samples not in a target language), and extracts features from text.
- A feature selection and combination process that yields a bag-of-words representation suitable for direct vector-to-vector comparison or machine learning (either supervised or unsupervised).
- A classifier training regime that yields author-specific models.
- A decision making process that makes a prediction for an anonymous tweet based on the classifier at hand, after the source data has been pre-processed and transformed into a bag-of-words representation.

Fig. 1 depicts how these components fit together. In the following pages of this section, we will explore each component in depth by reviewing the most relevant literature for features, classifiers, and strategies for attribution of small samples, as well as those drawn directly from social media. Further, we will introduce the idea of counter-attribution, leading to our recommendation that algorithms should always be designed with an adversarial scenario in mind. The papers presented below were chosen based on their relevance to the problem of social media forensics. Many other works exist in the broader field of authorship attribution, and we refer the interested reader to the existing surveys [89], [108], [175] that are more general in scope.

A. General Stylometric Features for Forensic Authorship Attribution

At the most basic level, the words of a text are useful features in and of themselves for authorship attribution. However, all words cannot simply be treated as features: the authorial signal will be buried in the noise of extraneous information selected by chance during composition. Thus, it is common to discard the *function words*, those words that occur most frequently but carry little if any semantic meaning (Table I) to isolate a more stable signal. However, function words can be useful for attribution in some cases. For instance, function words can be coupled with the most frequent punctuation, or other stable features, becoming more flexible and discriminative for use in a bag-of-words representation [93] that disregards grammar, but preserves the underlying statistics of language. Indeed, despite being one of the earliest features suggested for manual authorship attribution [128], statistics related to function words still appear as input to some algorithmic approaches [111].

Other instantiations of word-based features also lend themselves to attribution tasks. Term Frequency-Inverse Document Frequency (TF-IDF) is a weighted measure of word importance that is commonly used in information retrieval tasks [93]. It is calculated by multiplying the term frequency by the inverse document frequency for a specific word, where term frequency refers to the number of times a particular word appears in a document divided by the total number of words in the document, and inverse document frequency a logarithmically scaled quotient reflecting the total number of documents in a corpus divided by the number of documents containing the word. When applied to authorship attribution, TF-IDF emphasizes the importance of key words commonly deployed by a specific author, while deemphasizing those that are function words. More basic measures of word rarity, such as the raw statistics of usage from a reference corpus, are also useful features. Further, it is possible to compute similar statistics based on the rarity of various word stems or lemmas (automatically determined via stemming [158] or lemmatization [64]).

Beyond word rarity measures, more elemental probabilistic features of language prove to be very effective for attribution tasks. And they do not need to be constrained to the word-level of analysis. The n-gram is a feature that represents the probability of an element e occurring given some history h, or $P(e|h)$. We will look at this feature in detail in Sec. III. The advantage of using n-grams is that they capture lexical preferences without the need of any *a priori* knowledge of a grammar for a language, which is necessary to identify features like function words. Word-level n-grams have been shown to be effective for authorship attribution [79]. The motivation for such a feature is that some authors might have a preference for some expressions composed of two or more words in sequence, the probability of which is captured by n-grams of these words. Further, Forstall *et al.* [56] and Forstall and Scheirer [57] have argued that character-level n-grams serve as useful proxies for phonemes, and express the *sound* of words — another facet of language that can be quantified as a feature for authorship attribution.

With respect to the interaction between features and attribution scenarios, cross-topic and cross-genre attribution represent realistic, but difficult attribution circumstances that must be considered for many forms of writing. Stamatatos has studied the robustness of n-grams for authorship attribution [176], and has shown that the vast majority of existing research in this area only examines a case in which the training and testing corpora are similar in terms of genre, topic, and distribution of texts. There are doubts over the possibility that all low-level representational approaches are equally effective across diverse sources of data. This is of particular interest in the context of social media, due to the highly varying nature of the messages. Sapkota et al. [160] show that not all n-grams are created equal, and group them into

three categories: morphosyntax, thematic content and style. This categorization improves our understanding of authorship attribution for a social network combining numerous demographics and special interests. Further, cross-topic analysis is growing in importance [159]. Given the rapid turnover in topics discussed on social media, the topics used for one user in training will likely not be the same during testing. Features that are able to address this are paramount [177].

The availability of natural language processing (NLP) toolkits for many languages enables the use of more complicated stylometric features based on syntactic or semantic analysis of texts [9], [64], [73], [74], [168], [179], [188]. Such features are more abstract and, in theory, more resilient to thematic changes. However, syntactic and semantic features heavily rely on the underlying NLP toolkits generating them, which can vary in effectiveness according to the domain and genre of documents being processed. In general, these measures are noisy and less effective than low-level features such as word- and character-level n-grams. However, they are useful complements that can enhance the accuracy of attribution models [9], [188].

Part-of-Speech (POS) Tagging, the automated grammatical annotation of text samples, is a potentially rich source of additional feature information. For instance, processing the line "A short sentence" with a POS tagger will yield (if accurate): [('A', 'Indefinite Article'), ('short', 'Adjective'), ('sentence', 'Noun'), ('.', 'Punctuation')]. The motivation for the use of a POS tagger in authorship attribution is that grammatical usage can serve as an important indicator of an author's style — even in short messages. To this end, it is possible to use a POS tagger specifically trained for messages from social media like tweets [65], extract POS n-grams from additional tweets therein, and train a classifier using these features. The features may be effective in isolation, or they may be used to augment other lexical feature sets. Such an approach can also incorporate Twitter-specific content because hashtags, links, retweets, etc. are assigned to specific POS tags. While POS tagging is a standard tool in NLP, we suggest for the first time its use coupled with supervised machine learning in authorship attribution for social media (see Secs. III and IV).

An alternative to feature sets based on frequentist statistics and supervised machine learning is to view the problem of authorship attribution under the lens of complexity analysis of authorship signal. Malyutov [124] rightly argues that many machine learning approaches assume feature independence, which is false when considering aspects of natural language like function words. Using the theory of Kolmogorov conditional complexity, one can quantify the entire signal of a text, thus capturing any latent dependency, and assess similarity between different signals in an information theoretical sense. This can be accomplished via the use of compression algorithms [36], [37], [124], [145], which produce individual models for the texts they compress. Compression models for individual known authors are applied to a text of unknown provenance, and the one that results in the highest compression rate observed determines the author that produced the text. According to Oliveira, Jr., *et al*. [145], complexity analysis has several advantages: (1) a straightforward implementation and application; (2) parameter-free operation; and (3) an overall judgement on the document as a whole. We will examine the utility of compression-based methods for social media forensics in Secs. III and IV.

B. General Classifiers for Forensic Authorship Attribution

Once a feature set has been chosen, the next step is to select a classification method. Unsupervised clustering is appealing, in that there is no need to assemble large pre-labeled training sets before making use of an attribution algorithm. In an early contribution to the field, Burrows [28]–[31] proposed the use of multivariate analysis in stylometry. In essence, the algorithm generates vectors of frequencies of function words and applies Principal Component Analysis (PCA) over them. Authors are then classified via data clustering. Multivariate analysis achieved some measure of success, and it quickly became well-established for authorship attribution (see: [12], [13], [19], [20], [75], [76], [80], [117], [132], [137], [186]). However, the presence of some labeled data often improves results dramatically. Accordingly, supervised approaches now dominate the field.

Simple supervised classification methods are often dismissed in favor of the more elaborate, highly parametrized, algorithms that are prevalent in the current literature. Occam's razor, of course, should not be ignored in cases where a desired error rate can be achieved without a large amount of computational time or extensive parameter tuning. Recall that a classifier is any mapping from unlabeled data to discrete classes [71]. Thus, a basic embodiment is statistical hypothesis testing. For attribution problems, this means formulating a null hypothesis that suggests that two works under consideration are from different authors, and testing it via a hypothesis test (*e.g.*, Student's t-test) [22], [39], [170], [171]. Of the various hypothesis tests that have been proposed, χ^2 is particularly attractive in that it can yield a ranked list of candidate authors in a 1:N attribution scenario [68], [134]–[136]. Another straightforward classification method is to compute distance to known vectors, and assign authorship based on the shortest distance to a known vector that passes a particular matching threshold. Cosine distance is one possibility for this [104].

The aforementioned techniques are formally supervised classifiers, but they do not learn from data. With the availability of good quality labeled data, supervised classifiers can be trained to make use of common patterns across many samples, instead of making direct comparisons between individual samples. Along these lines, K-nearest Neighbors (K-NN) assigns class membership based on the distance of an unknown point to K points that are closest to it [71]. For attribution, if the majority of those K nearest points are from the same author, we can conclude that the unknown sample should also be associated with the author of those points [78], [97], [105], [195]. Another supervised classification method is Naïve Bayes, which is a straightforward application of Bayes' theorem, whereby the probability of a given label y, and the probabilities of particular feature vectors x_i given a label y are learned via Maximum A Posteriori (MAP) estimation. When applying Naïve Bayes to authorship attribution, the resulting

probability value for class assignment can be consulted to determine if a match to a known author (reflected by the label y) has been made [42], [78], [141], [150], [161], [195].

More sophisticated probabilistic formulations go beyond the assumption of Naïve Bayes that all feature vectors are independent in order to capture meaningful dependencies across data samples. Markov Models for authorship attribution calculate probabilities of letter or word transitions, which are style-specific markers, placing them into author-specific transition matrices [100], [101], [114]. Classification proceeds by calculating the probability that an unknown text was generated by a particular transition matrix. Similarly, Kullback-Leibler Divergence is used to measure the relative entropy between the probability mass functions of features extracted from texts, with authorship assigned to a pairing with the lowest relative entropy [196], [197]. Other information theory variants incorporating cross-entropy have been shown to be viable as well [90], [183].

Highly parameterized models are a better option to learn complex relationships in high dimensional feature spaces. As artificial neural networks took hold in natural language processing in the 1990s, they found their way into authorship attribution in part for this reason. Multi-layer neural networks are well suited to learning models for non-linear problems, and can process an enormous diversity of feature sets. Matthews and Merriam [129] proposed extracting specific word ratios and function words as features, and then applying back propagation to learn the weights for a small network topology of five inputs, three hidden units, and two outputs (a comparatively tiny architecture, considering today's deep learning architectures [17]). This work was followed by many others in the same vein, some of which trained more elaborate networks as computational resources expanded (see: [78], [102], [103], [105], [118], [120], [127], [138], [184], [187], [190]). While it is commonly alleged that it is not possible to determine the basis by which neural networks make classification decisions, the recent resurgence of this area has yielded work that shows that this is possible for some architectures in computer vision [193]. A similar analysis for NLP would be an important next step.

Like other areas of machine learning such as core NLP and computer vision, the development of features and classification methods for authorship attribution has tracked the evolution of processing power in computer hardware, moving from univariate to multivariate analysis and expanding from a handful of simple function words to thousands of character- and word-level n-grams. In authorship attribution, expanding the feature space improves classification, but also requires additional computational resources for better classifiers to adequately process the number of generated features. Fortunately, algorithms more efficient than neural networks exist for such scenarios.

Nearest Shrunken Centroid [71] is a viable classification method for authorship attribution from high-dimensional feature vectors [88]. During training, the algorithm calculates a standardized centroid (the average of all of the training points) for each authorship class, which is subsequently made more compact via a parameterized threshold. A text whose authorship is questioned is then compared to each centroid, with the resulting smallest distance indicating the authorship prediction. More recent work on open set authorship attribution [162], when it is assumed that not all possible authors are known at training time, has applied an additional threshold over the distance scores to allow for the possibility of rejection (*i.e.*, no centroid for a known author is close enough to the text under consideration). Traditional multi-class classifiers will always assign class membership, regardless of whether or not the sample under consideration is known. Open set recognition is a key open issue in authorship attribution, and we will return to it in the following sections with specific examples.

Following trends in supervised machine learning, Support Vector Machines (SVM) emerged as a method in authorship attribution to address the same problem of high-dimensional feature vectors for which Nearest Shrunken Centroid was proposed. In practice, the SVM's concept of maximum margin leads to better generalization, and thus better accuracy for binary and multi-class classification problems. SVM was first applied to authorship attribution through the works of de Vel et al. [46] for e-mails and Diederich et al. [47] for German newspapers. Many subsequent works highlighted the overall success of SVM in classifying authors (see: [1], [9], [53], [57], [63], [64], [73], [106], [110]–[112], [148], [159], [161], [168], [174]), making it a dominant classification strategy in the field. The accuracy of approaches that use both n-grams and SVM are further discussed in a recent report from Stamatatos [176], who has investigated the question of whether the n-grams remain effective for cross-topic authorship attribution.

SVMs are powerful, but they can easily overfit the training data when kernelized, and are more suited to binary problems. Decision Trees, which use a graphical model over multiple inputs to assign a class label to incoming data, are an alternative. There is some utility to using decision trees on their own [1], [8], but a meta-learning technique considering ensembles of them is more powerful. Random Forest treats decision trees as weak learners, and randomly subsamples sets of features from a specific training dataset to improve accuracy and mitigate overfitting. Recent research efforts have made extensive use of random forests for authorship attribution and verification [15], [33], [123], [147], [151]. Random forest classifiers are also attractive because they provide a clear indication of the feature weighting via variable importance measures. We will return to the discussion of Random Forests and SVMs in the following sections of this article.

Tables VI, VII and VIII in the appendix summarize the classification approaches discussed in this section, along with their respective feature sets. Which of these algorithms is the best? Jockers and Witten [87] provide a comprehensive examination of top performing classification methods for authorship attribution, including: Delta (a simple z-score-based method prevalent in literary studies [82]), Nearest Shrunken Centroid, SVM, K-NN, and Regularized Discriminant Analysis [70] (other forms of discriminant analysis were successfully applied to authorship attribution in the past [40], [179], [180] as well). On a benchmark data set of English language prose, nearest shrunken centroid and regularized discriminant analysis were shown to be the best choices. Other studies have revisited the

utility of simpler approaches such as multivariate analysis via PCA clustering and Delta [99], where both are shown to be sufficient to address the problem of collaborative authorship analysis (*i.e.*, where two or more authors collaborate to produce a single text). The choice comes down to the specific problem domain — one must select the right tool for the right task. Thus, we turn our attention next to methods that are suitable for social media forensics.

C. Authorship Attribution for Small Samples of Text

The approaches discussed thus far have mostly been applied to problems in which a large amount of text is available (*e.g.*, novels, essays, newspaper articles, etc.). What do we do when we do not have that luxury? A tweet, for instance, is a mere 140-characters long, and does not yield a large amount of information at the word-level, or from its syntactic structure. Even before the advent of social media, researchers had been investigating this problem in the context of forensic stylometry in e-mails, where short form writing is the norm. Some of the strategies we discussed above, such as similarity measures [27], [109] and SVM [113], [152] apply directly, but better performance can be achieved with features and classification approaches custom-tailored for attribution problems with small samples of texts.

Anderson et al. [6] and de Vel et al. [46] turned to a variety of character-level statistics such as capitalization, white space, and punctuation counts to compensate for the small amount of information inherent in such content. More specific to Internet messaging, they also evaluated structural attributes of the messages including the presence of a greeting, farewell, and signature in the text. Combined with SVM for classification, these features were shown to be reasonably effective (classification accuracies typically between 80-100%) for attribution problems consisting of a small number of authors (three). Early to note idiosyncratic usage, Koppel and Schler [106] looked at specific error patterns including sentence fragments, mismatched tense, and letter inversion as style markers for e-mail. When combined with decision trees, such features alone can achieve nearly 70% classification accuracy for a corpus of 11 authors.

In turn, Sanderson and Guenter explored using "Author Unmasking" [111] to address this problem [158]. The idea behind author unmasking is that the differences between two texts from the same author will be reflected in a relatively small number of features. This set can be extracted by use of an author unmasking curve, which corresponds to the accuracy of an SVM classifier when essential features are repeatedly removed. However, it was shown that there is significant aliasing between different-author and same-author performance curves when considering samples of 5,000 words or less. The sequence-based approaches of Character-level Markov Chains and Character-level Sequence Kernels [189] are suggested as alternatives. Partial parsing for short texts, which is a compromise between complete parsing and chunking (*i.e.*, phrase labeling), was described by Hirst and Feiguina [74]. When features from partial parsing are combined with SVM, high accuracy can be achieved (over 90%) for samples as small as 200 words. The downside of these particular works for forensic examiners primarily concerned with amateur writing on the Internet is that the corresponding evaluations only simulated the form of short writing found there.

Small samples of text outside of the realm of natural language are also an interesting test case for attribution algorithm development. Source code, by its very nature, consists of small routines like functions or methods, which can reflect certain stylistic traits of the programmer. Source-code attribution is used for cybercrime forensics [172] and plagiarism detection [58], but some methods [59], [60], [115] generalize beyond code to all languages (formal and natural), making them attractive for social media analysis. What features are effective for identifying the authors of code? Some are very task dependent, such as the line, comment, operator, and operand counts suggested by Hayes [72]. Others are based on n-gram features, which consist of sequences of tokens [32]. Frantzeskou et al. [59], [60] extended this idea with two key differences: the use of raw frequencies of byte-level n-grams and a simple overlap measure for classification. We will look at this Source Code Authorship Profiling (SCAP) methodology in more detail in Secs. III and IV.

Further along the lines of programming language authorship attribution, the state of the art for de-anonymizing programmers shows significant promise for the analysis of short samples of text. The best reported methodology makes use of features derived from abstract syntax trees. Caliskan-Islam et al. [34] applied this methodology to code stylometry, where random forest classifiers were trained from a very large (*e.g.*, "120,000 features for 250 authors with 9 solution files each") set of lexical, layout and syntactic features from the abstract syntax trees of target source-code solutions. Information gain was applied to select only the more informative features, making the approach more accurate and the computation more tractable. When validated on code solutions that were, on average, 70 lines of code long, accuracies for distinguishing between sets of 1,600 and 250 programmers reached 94% and 98%. This methodology is not constrained to just source code — Caliskan-Islam et al. also demonstrated its applicability to executable binaries [35].

The authorial style of malicious source code often percolates through other media where short-form writing is prevalent. Afroz et al. [3] conducted a large-scale study of posts on underground forums related to password cracking, spam, credit card fraud, software exploits, and malicious search engine optimization. Given the tight coupling between the culture of the computer underground and the code it develops, common lexical features are often present. Afroz et al. suggest several that are amenable to SVM classification: character-level unigrams and tri-grams, word-level bi-grams, numbers used in place of letters, capitalization, parts of speech, and the presence of foreign words. These features are directly applicable to social media at large, and we will consider several of them in our practical walk-through of authorship attribution below.

It is also possible to gather more information out of limited sample sizes by looking for primitive sound features.

Forstall and Scheirer introduced the concept of the *functional n-gram* [57], which applied at the character-level, is an n-gram-based feature that describes the most frequent sound-oriented information in a text. Similar to function words, functional n-grams are those n-grams that are elements of most of the lexicon, necessitating their use. While originally developed for the analysis of poetry and other fragmentary literary writing, this feature is genre-independent, and can be used for social media forensics to generate over a hundred independent feature dimensions from a tweet. The utility of these features was also demonstrated for a similar task of *influence detection* [56], whereby one author's influence on another is gauged by comparing stylistic commonalities between their respective texts.

Table IX provides a summary of the work in authorship attribution for short samples of text described above, along with the database used in the analysis. While many approaches have been suggested, very few works have examined the fundamental problem of authorship attribution for small samples in a theoretical sense. How small can we go before the signal becomes lost in the noise? Eder examined this problem in a method-independent manner [49], and determined that for modern languages, the minimal stable sample size is 5,000 words. Curiously, this is in direct opposition to the prior work we have just discussed. The finding suggests that some instances of authorship attribution are domain dependent — Eder examined modern language novels, a prose form that is rather distinct from the type of writing typically found on the Internet. Particularly with respect to authors attempting to have impact in 140 characters, a personal style is likely to be honed and more forcibly deployed on Twitter.

D. Authorship Attribution Specifically for Social Media

A growing body of work has attempted to mine and analyze actual online postings. Abbasi and Chen produced a body of work that included an analysis of eBay comments and online forum posts [2]. A Karhunen-Löeve transform-based technique dubbed "WritePrints" was applied to this data, showing accuracy as high as 94% when differentiating 100 distinct authors. With the appearance of Twitter in 2006, the problem grew significantly more difficult. To date, only a handful of researchers have tackled the authorship attribution problem for tweets collected in the wild using the techniques described above. Methods relying on SVM for classification [18], [67], [139], [166], [169] outperform other approaches to authorship attribution on tweets [166], namely Naïve-Bayes [5], [23], Source-Code Authorship Profiling [116], and other simple similarity measures [85], [96], [144].

Almost all of these approaches used the same set of features, character- and word-level n-grams. These features are typically parameterized via empirical performance: $n = 4$ for character-level n-grams [113], [116], and $n = (2, \ldots, 5)$ for word-level n-grams in a traditional bag-of-words model (which we will describe in detail in Sec. III). A limiting factor has been the difficulty of learning useful decision boundaries from short messages with the typical implementations of common supervised machine learning methods. Other strategies also exist outside of the definition of specific feature sets for small texts of any type. Arakawa et al. [7] explored a Twitter-specific approach that assessed the type and number of retweets.

A further complication is the need for automatic language understanding for Eastern and Near Eastern language posts, where non-Latin character sets are used, and individual characters may express entire words. While language-dependent approaches like partial parsing fail without substantial retuning in such cases, language-independent character-level n-gram- [97], [149] and TF-IFD-based [4] approaches work just fine for non-Latin characters with no adjustment. For Arabic, elongation (characters are extended for stylistic purposes), inflection (many stem words are derived from a common root), and diacritics (marks above or below characters that are used to indicate special phonetic values) all present additional information in the feature space. Abbassi and Chen designed a specialized feature set that captured this information, and used it to determine the authorship of posts to a forum associated with the Palestinian Al-Aqsa Martyrs group[7] [1].

One might also ask if a writing style evolves over time in a way that is unique to a specific social media platform like Twitter. Azarbonyad et al. [11] have studied this question, and have isolated distinct vocabulary changes for the same authors of tweets over a period of months. The cause might be as simple as a change in the circumstances of an author's life, or as nuanced as the absorption of stylistic traits after reading the tweets of others. It is possible to design a time-aware attribution algorithm that constructs a language model for distinct periods from an author's collected messages, which can be achieved by calculating decay factors that are applied as weights to the periods. Using character-level n-grams as a feature basis, Azarbonyad et al. showed that a time-aware SCAP approach is far more effective than a baseline without any temporal weighting.

Going beyond lexical- and sound-oriented features, semantic analysis can also be applied to attribution tasks. Seroussi et al. [167] propose combining Latent Dirichlet Allocation (LDA) and the Author-Topic Model approach to form a Disjoint Author-Document Topic Model (DADT). In DADT, author topics are disjoint from document topics, different priors are placed on the word distributions for author and document topics, and a ratio between document words and author words is learned. The feasibility of this approach has been demonstrated on emails and blog posts. However, it is not always possible to perform meaningful semantic analysis on sample sizes as small as tweets with any of today's topic modeling algorithms. In our own work, we have found that the LDA implementations contained within Mallet [130], Gensim [155], and R [83] all fail by producing radically different match scores for the exact same input across multiple trials. According to the literature [24], this stems from random bootstrapping with inadequate sampling. LDA is wonderful for modeling the types of large-scale corpora found in electronic collections such as JSTOR or Wikipedia,

[7] A modified version of the feature set was also used by Abbassi and Chen to determine the authorship of posts to a US forum belonging to a group associated with the Ku Klux Klan.

but available implementations are not designed with the same set of assumptions under which a forensic investigator reading posts on social media is operating. Surprisingly, a general numerical solver that is stable over many small samples has not been forthcoming.

More specialized semantic analysis approaches have been proposed to address this shortcoming. By aggregating tweets [77] into per-user profiles for training and testing, conventional topic modeling algorithms can be applied with little trouble. However, this strategy is not feasible if we are considering just a single testing tweet in an actual investigation. Building from work in generative models for single small texts with a single topic [119], [185], Zhao et al. [194] achieved stable results for semantic analysis applied to individual tweets. This was done by modeling the tweet generation process as sampling from a distribution for a chosen topic (as opposed to a more complicated mixture for longer form text), based on the assumption that an author has a single chosen topic in mind, which is reflected in the tweet. Another potentially viable approach is Brown clustering, which partitions the words in a text into a base set of clusters, and creates a hierarchy among them by optimizing the likelihood of a hidden Markov model with a one-class-per-lexical-type constraint. By leveraging a training corpus of aggregated tweets, Owoputi showed that Brown clustering is feasible to apply on a per-tweet basis during testing [146]. In all of these cases, it remains to be seen if semantic analysis can be specific enough to stand by itself as an authorship attribution method. More realistically, it is likely to be useful as an additional feature dimension within a larger feature set. Table X provides a summary of the work in Eastern and Near Eastern languages, authorship attribution for social media, and semantic analysis for short texts.

E. The Threat of Counter-Attribution

Naturally, in authorship attribution, there exists some element of the "offense and defense" dynamic present in the broader world of computer security. Counter-attribution techniques, where there is an intentional act of changing one's writing style, have emerged to thwart authorship attribution systems. An admirable goal of counter-attribution is to facilitate anonymous free speech on the Internet [142]. Of course, counter-attribution can be misused by malicious actors attempting to evade identification by various authorities. Nonetheless, several strategies exist that could be used for legitimate or illegitimate purposes.

Kacmarcik and Gamon [94] describe shallow anonymization, whereby 14 changes per 1,000 words disrupts an SVM classifier trained with function words, and deep anonymization, whereby increasing numbers of feature modifications defeat an approach relying on the rate of degradation of the accuracy of learned models [107]. Juola and Vescovi [92] studied the impact of the Brennan-Greenstadt corpus [26], which was cleverly crafted to deliberately mask style, on the Java Graphical Authorship Attribution Program.[8] The Brennan-Greenstadt corpus makes use of the following strategies for counter-attribution: obfuscation (*i.e.*, identity hiding), imitation (*i.e.*, the deliberate misappropriation of another author's style), and translation (*i.e.*, the use of machine translation to alter an author's signal). The findings of Juola and Vescovi indicate that all common feature sets are impacted by counter-attribution, but some less than others (*e.g.,* character-level n-grams).

Fundamentally, if an author's signal is scrubbed from a text, we would have to turn to other evidence associated with the case to make a confident attribution decision. However, it remains unknown whether or not a perfect counter-attribution solution can be developed, given the vast linguistic feature-space of any underlying text. The Anonymouth framework of McDonald et al. [131] is a step in this direction, but for prose. It is an open question as to whether or not an anonymization framework like this can be applied to small samples such as tweets.

III. Walk-Through of Authorship Attribution Techniques for Social Media Forensics

In the previous section, we surveyed what is available in the literature for authorship attribution, and looked at specific approaches that are suitable for social media forensics. In this section, we will walk through the process of identifying the author of a given set of tweets from Twitter. This material is meant to illustrate the process for newcomers to the field, and highlight the difficulty of the problem when examining very small messages.

The basic strategy we will look at relies on a set of features capturing patterns extracted from the original texts in a bag-of-words model dynamically created for a set of users.[9] When creating a bag-of-words model, we can consider one model for a set of authors or one model per author. A dynamic model for each author could allow for a more fine-grained stylometric evaluation, while a general bag of the most frequent patterns comprising many authors at the same time may overlook some discriminative features of particular authors, as those features may not be strong enough to appear globally. However, as the number of authors increases, creating a dynamic model for each investigated author is much more time consuming. In both cases, the bag-of-words model works as a projection space in which we aim to highlight similar patterns of a particular author or set of authors, while decreasing other patterns not in accordance with the overall style for those authors of interest. For computational efficiency, we will consider the case of a model per set of authors.

Various classification methods are applicable to this problem. We address high-dimensional feature representations using variants of one of the most used classifiers in the literature: the Support Vector Machine. Specifically, we show that it is worth exploring implementations that better handle large-scale classification [192] with respect to accuracy, as well as speed of processing. For open set attribution problems, classifiers specifically designed to mitigate the risk of the unknown are of interest [163]. To provide a comparison to SVM-based classifiers, we also look at Random Forests, as

[8]https://github.com/evllabs/JGAAP

[9]Much of what we discuss below can be implemented via the Stylo package [51] in R.

recent work in machine learning singles them out as very efficient and effective for various classification tasks [55]. Alternatives to all of these more general supervised machine learning-based approaches are provided in the form of Source Code Authorship Profiling (SCAP) [175] and compression-based attribution [183].

The source code for all implemented algorithms in this article will be publicly released following publication.

A. General Framework for Authorship Attribution

Fig. 1 depicts a general framework for the authorship attribution system we will consider here. For social media analysis, we recommend an approach that scales well with a large number of suspects. A typical scenario could be a tweet that describes illicit activity, sent anonymously from within a small company (from a handful up to five hundred employees). If there is no physical evidence linking one of those employees to the message, all employees become suspects, and the use of a machine learning approach becomes paramount.

During training, messages are harvested from the social media accounts of known suspects (Sec. III-B1). After enough examples have been collected, the raw data is pre-processed (Sec. III-B2) to remove very sparse features, very short messages, and non-English messages, which enforces consistency in the subsequent feature extraction. In this article, we evaluate character-level n-grams, word-level n-grams, part-of-speech n-grams, and diverse lexical and syntactic statistics as features (Secs. III-C1, III-C2, III-C3, and III-C4). All features are collected into feature sets (Sec. III-C) that are based on the common bag-of-words model in natural language processing. These feature sets are then used to train a binary or multi-class classifier, depending on the number of users considered. For classification, we examine the utility of the Power Mean SVM (Secs. III-D1), W-SVM (III-D2), Random Forests (Sec. III-D3), SCAP (Sec. III-D4), and compression-based attribution (Sec. III-D5).

Testing begins with a message of unknown authorship, which proceeds through the exact same feature extraction process as the training messages. The resulting feature vector is submitted to the pre-trained classifier, which produces a prediction of its authorship. This result points out the most probable suspect from a set of possible ones.

B. Source Data From Twitter

Two preliminary steps are required before feature extraction and classification: data extraction and text pre-processing.

1) Data Extraction: Since the ultimate forensic goal is authorship attribution, all the retweets, which are messages retransmitted from other users, should be removed. This is done by removing all tweets marked by the Twitter API with a specific retweet flag, as well as tweets containing the meta tag RT [116], [166]. Our focus is on English-language tweets, thus non-English tweets can be removed using the python library guess-language [173], which itself uses the spell-checking library pyenchant [95] to build an accurate prediction of the language of a text consisting of three or more words. For this reason, we recommend removing all messages that contain only one or two words. This is not done to the detriment of accuracy — these short messages do not provide meaningful information about the author and end up introducing noise into the classification task [166].

2) Text Pre-Processing: From our experience, there is no need for strong pre-processing in authorship attribution. This is an artifact of the choice of features we commonly use. While a writer's preferences for letter capitalization and word suffixes, along with their grammatical mistakes, may frustrate many natural language processing systems, they are also an integral part of their writing style. In this context, it does not make sense to stem words or correct the grammar of messages under consideration. While such actions would greatly reduce the number of features to be analyzed, they would also remove many idiosyncrasies, which are unique to a user [116], [166] such as Internet expressions, repeated grammatical mistakes, abbreviations and even preferences for certain *Emojis*.[10]

Therefore, instead of more elaborate tokenization [143], [146], our pre-processing will focus on normalizing very sparse characteristics such as numbers, dates, times and URLs. These are relevant characteristics in some sense, but it is unlikely that a user will be sharing, for example, the same date many times. Hence the pre-processor takes away the original text and replaces it with a standard tag that represents the replaced content.

Moreover, it has been noted that hashtags[11] and user references[12] make authorship attribution easier [116]. A user might repeatedly use the same hashtag or frequently address messages at a single person. However, it also makes supervised learning methods unreliable, because a user might make references to the same person across her messages, creating a strong bias towards that particular feature in training; and any message with a reference to that same person would subsequently be misclassified as being from that user.

In the following example, we show three tweets before and after the aforementioned pre-processing procedure. The "after" examples contain each type of tag.

Before pre-processing:
Tweet 1: "Do not forget my bday is on 03/27 #iwantgifts"
Tweet 2: "@maria I will be sleeping @00:00AM"
Tweet 3: "Check out this amazing website: http://www.ieee.org"

After pre-processing:
Tweet 1: "Do not forget my bday is on DAT TAG"
Tweet 2: "REF I will be sleeping @TIM"
Tweet 3: "Check out this amazing website: URL"

C. Bag-of-Words Model

The bag-of-words is a classic model in natural language processing. It is an orderless document representation of

[10]Emojis are small images used to express ideas or sentiment in text messages and now social media. With roots in Japan, the word emoji literally means "picture" (e) + "character" (moji).

[11]Hashtags are keywords used in tweets and other social media to make searching for messages with a common topic easier; usually they are preceded by a '#' character and found amid the text.

[12]The users of Twitter and other social media platforms can send messages to other users using an '@' character followed by their username.

TABLE II
DICTIONARY FOR A BAG-OF-WORDS MODEL DERIVED FROM TWO LINES OF TEXT. SUCH A DICTIONARY DEFINES THE STRUCTURE FOR FEATURE VECTORS THAT CAN BE USED IN CONJUNCTION WITH SUPERVISED MACHINE LEARNING

#	1	2	3	4	5	6
Word	to	be	or	not	that	is
#	7	8	9	10	11	12
Word	the	question	die	sleep	perchance	dream

feature frequencies from a dictionary [157]. Although the term "word" is found in the model's name, the dictionary may consist of groups of word- and character-level n-grams, or any other features that can be extracted from text. For natural language processing tasks such as information retrieval and sentiment analysis, it has been suggested that function words should be removed [126]. However, for a task such as authorship attribution, function words can sometimes provide more information about the author than more meaningful words [57].

That observation is largely derived from Zipf's law [200], which states that the frequencies of the words are inversely proportional to their rank in an overall frequency table. Even though the most used words vary for each author, they are always among the function words, thus they become reliable features across even very large populations of authors. We could extract many of the nuances of style for a particular author by looking at the function words alone [133]. But in authorship attribution for very short messages, it is paramount to gather as many features as possible. Due to the compressed nature of such messages, most of the words contained in a single text are likely to appear only once.

Let us work through an example of the bag-of-words model that strives to maximize the amount of information that will be available for feature extraction, starting with the following two lines of Shakespeare, which will serve as our short "messages":

Text 1: "To be, or not to be, that is the question."
Text 2: "To die, to sleep. To sleep, perchance to dream;"

In the first step, we create a dictionary that maps each feature onto a unique index. In this example, which is shown in Table II, we simply use the words from each line as the features. The descriptive indices are then used to define the structure of the feature vectors.

With the dictionary, we can create individual feature vectors based on statistics calculated from the texts. This could be as simple as using the raw frequency of each word (*i.e.*, the counts of the words in an individual example text), but other statistics might be considered as well, such as term frequency–inverse document frequency (tf-idf) scores [57]. In this example, we will use binary activation vectors that indicate the occurrence of the feature (1) or its absence (0). Binary activation is effective for small messages because in most cases, words will not be repeated, and the few that are should not bias the final outcome. Using that strategy, the final feature vectors for this example are:

Feature vector 1: [1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0]
Feature vector 2: [1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1]

The bag-of-words model is used throughout the rest of this article, and its dictionaries are constructed using character-level n-grams, word-level n-grams, part-of-speech n-grams, and diverse lexical and syntactic statistics as presented below.

1) Character-Level n-*Grams:* Character-level n-grams are often used for authorship attribution in texts from social media because they can capture unusual features, such as emoticons and special use of punctuation. They also help mitigate the effect of small typos that authors do not repeat very often, which are not style markers. For example, for the word "misspeling", the generated character-level 4-grams would still have "miss," "issp," "sspe," "spel" and "ling" in common with the 4-grams generated for the correct word "misspelling."

Following the literature [116], [158], [166], we focus on character-level 4-grams herein, with whitespace and meta tags included in the n-grams. With respect to Twitter, whitespace is appended at the beginning and at the end of each tweet. Also, we discard any character-level 4-gram which does not appear at least twice for the same author [166] in the training set, thus eliminating *hapax legomena* (features that occur exactly once in a particular setting). This decision improves efficiency by removing noisy features that are unlikely to appear again in the future for the same user.

The features are case-sensitive, since the author's preference for capitalization of letters is also one of the traits that can be used for attribution. Many users of social media have a preference for capitalizing some words to emphasize them or writing in the exaggerated camelCase form.[13]

The following example shows a punctuation rich phrase and the complete list of 4-grams extracted from it (for readability, we replaced spaces with '_'):

Text: "2B!!!! or n2B!!!!!!! ;)"

4-grams: 1. "_2B!" 2. "2B!!" 3. "B!!!" 4. "!!!!" 5. "!!!_" 6. "!!_o" 7. "!_or" 8. "_or_" 9. "or_n" 10. "r_n2" 11. "_n2B" 12. "n2B!" 13. "2B!!" 14. "B!!!" 15. "!!!!" 16. "!!!!" 17. "!!!!" 18. "!!!!" 19. "!!!_" 20. "!!_;" 21. "!_;)" 22. "_;)_"

This approach is able to isolate the repeating patterns of exclamation marks and the emoticon.[14] If we included character-level n-grams for other values of n, we would generate redundant data either by repeating something captured by $n = 4$ when choosing $n < 4$, or by capturing a crude proxy for word-level unigrams (another feature we will deploy) when $n > 4$. Having the same feature duplicated in the training set creates a bias towards it, which is undesirable.

2) Word-Level n-Grams: Word-level n-grams let us capture more semantically meaningful information from a text in the form of short phrases [98]. When considering messages from social media, the constraints placed on message length force users to be more judicious in their writing. Thus, it is reasonable to assume that authors will economize, and only repeat very short phrases [166]. Punctuation sequences can

[13] camelCase is a style that alternates between lower and upper case letters *e.g.*, "ExAmPlE."

[14] Emoticons should not be replaced by meta tags because they are an integral part of internet culture. Users of social media may very well have a particular combination of emoticons they use on a regular basis [169].

also be included, considering that they may be a significant component of a phrase.

A good rule of thumb is to use word-level n-grams where $n \in \{1, \ldots, 5\}$, but special consideration should be given to the unigrams (1-grams). Character-level 4-grams will generally include many of the unigrams, and their use substantially increases the length of the feature vectors. However, contrary to their typical characterization in the literature, we will show that they can improve classification performance when used under the right circumstances. Similar to our treatment of leading and trailing whitespace for the character-level n-grams, meta tags are used at the beginning and end of each tweet to distinguish words frequently used to start and to end messages. All word-level n-gram features are also case-sensitive. Similar to the procedure for character-level n-grams, we also eliminate *hapax legomena* for word-level n-grams.

The following example shows a simple phrase and the complete lists of unigrams, bigrams, and trigrams extracted from it[15]:

> **Text:** "To be, or not to be, that is the question."
>
> **Unigrams:** ("To", "be", "or", "not", "to", "be", "that", "is", "the", "question")
>
> **Bigrams:** ("BEGIN To", "To be", "be or", "or not", "not to", "to be", "be that", "that is", "is the", "the question", "question END")
>
> **Trigrams:** ("BEGIN To be", "To be or", "be or not", "or not to", "not to be", "to be that", "be that is", "that is the", "is the question", "the question END")

For n-grams in general, feature vector length varies as a function of the choice of n, the number of known authors, and the number of texts considered during training. For word-level n-grams, the vector length can grow rapidly. For instance, when $n = 4$ the feature dimensionality varies from 20,000-dimensional vectors (50 users and 50 training tweets per user) to around 500,000-dimensional vectors (500 users and 500 tweets per user). Although some researchers have argued for $n < 4$ as a default setting (*e.g.*, Forstall and Scheirer [57] used $n = 2$), for messages from social media, we need a larger n to capture the idiosyncratic language usage of the Internet, which includes emoticons, onomatopoeia (a word that resembles its associated sound, *e.g.* cuckoo), abbreviations, and other unconventional usage.

Moreover, some work in the literature has consciously avoided the use of word-level unigrams for message characterization and feature generation. This is because unigrams are supposedly captured by character-level n-grams [166], and subject to the explosion in dimensionality of the feature representation (in the worst case, a probability value would be assigned to each word in the document under analysis), which hampers the classification process. However, in Sec. IV, we show that unigrams, when combined with character-level n-grams, improve the classification accuracy of authorship attribution for social media, and researchers must take them into account when exploring solutions in this field.

[15]In the example, we used "BEGIN" and "END" to mark the start and end of the text, although when implementing this in code we used non-printable characters to avoid mismatching with the respective English words.

3) Part-of-Speech (POS) n-Grams: The simplest stylistic features related to syntactic structure of texts are part-of-speech (POS) n-grams [122]. POS tagging is a process that can be performed easily and with relatively high accuracy. Given a POS tagger that has been trained with texts possessing similar properties to the ones under investigation, it can achieve near-optimal accuracy at the token level [182]. Noise in stylistic measures can significantly affect the performance of the attribution model; POS tags are one way to mitigate this concern because they reduce the feature-space to a limited number of very general elements.

As described above, the texts found in social media are usually characterized by peculiar use of language. Each social network essentially defines its own textual genre based on service-specific characteristics (*e.g.*, Twitter only permits short messages). In this study, we use a POS tagger specifically designed for tweets in English [65]. A POS tagset of 25 tags, covering twitter-specific features like hashtags, at-mentions, retweets, URLs and emoticons was used [7]. Previous studies using this POS tagger report a tagging accuracy of about 90% at the token-level for English tweets [65].

The following example shows the output of the POS tagger for tweet B.2 from Sec. I:

> **Tweet:** "Thx 4 having me. Great time w/ you all"
>
> **POS tags:** ("N", "P", "V", "O", "W", "A", "N", "P", "O", "D")

where tags "N", "P", "V", "O", "W", "A", and "D" correspond to common noun, preposition, verb, pronoun, punctuation mark, adjective, and determiner, respectively. Appendix B shows the complete set of POS tags considered here.

4) A More Diverse Feature Set for Open Set Attribution: Open set recognition problems are among the hardest in machine learning — and authorship attribution in this setting is no exception [181]. In open set attribution, we have the expectation that many samples from unknown authors will be submitted to an attribution model, all of which should not be assigned association to any known author. Given the scale of data found on social networks, this means that a model could assess millions of negative samples, while rarely (if ever) encountering samples from a known author. To introduce the concepts of open set attribution, we consider the most basic matching scenario in this article: *authorship verification* via pair matching. Given two tweets, a prediction is made as to whether or not they are from the same author. Even for this very simple scenario, a feature set that is limited to just n-grams, which is effective for closed set attribution (*e.g.*, all authors seen at testing time are known at training time), does not yield enough information diversity to make accurate predictions. Thus we must turn to *feature-level fusion* over more diverse feature types to capture additional facets of style at the character and word levels, which will allow us to match many common elements across samples.

Feature-level fusion is the process of combining different individual features that have been extracted from the input text samples into one feature vector before training or classification

TABLE III

DIVERSE FEATURE TYPES GENERATED FOR THE OPEN SET ATTRIBUTION APPROACH, WHICH NEEDS TO RELY ON MORE STATISTICAL INFORMATION THAN JUST n-GRAMS BECAUSE OF THE DIFFICULTY OF THE UNDERLYING PROBLEM. 71 FEATURES DERIVED FROM THESE BASIC CATEGORIES ARE USED FOR THE AUTHORSHIP VERIFICATION APPROACH DESCRIBED IN THIS ARTICLE

Feature Type
Number of exact-word matches
Number of stem-based matches
Number of unique forms of matching stems
Number of unique forms of matching words
Mean frequency of word matches
Minimum frequency of word matches
Sum of inverse frequencies of word matches
Mean frequency of all words
Minimum frequency of all words
Sum of inverse frequencies of all words
Mean tf-idf score of matching words
Sum of tf-idf scores of matching words
Max of tf-idf scores of matching words
Mean tf-idf score of all words
Sum of tf-idf scores of all words
Max of tf-idf scores of all words
Distance between two furthest matching words
Combined distance between two furthest matching words
Distance between the lowest frequency words
Distance between the two highest tf-idf frequency words
Fraction of matching character-level unigrams out of total
Fraction of matching character-level bigrams out of total
Fraction of matching character-level trigrams out of total
Similarity of bigram frequencies between two strings
Similarity of trigram frequencies between two strings

takes place. This blending is a concatenation meant to lift the accuracy of machine learning by enhancing classification with longer, and ideally more distinct, feature vectors. For the assessment of whether or not two samples of text are by the same author, the process begins by calculating 71 distinct features derived from the basic set listed in Table III. This is followed by a concatenation operation to build a single feature vector. Such a feature vector can then be used as a training sample, or as a testing sample that we would like to make a prediction about. These features broadly fall into the following three categories: (1) frequency of word matches, (2) frequency of all words, and (3) character-level n-grams. With pairs of texts available, features can be generated by considering each sample individually, or together. Further, some features utilize additional information from the aggregate tweets for an author (*e.g.*, used as the "document" in tf-idf calculations), or entire corpus (*e.g.*, a frame of reference to determine whether or not a word is rare in general). The reference code that will be included in the final version of this paper details the exact parameters for each feature.

D. Classification Strategies

We now turn our attention to the five classification strategies considered in this study: the Power Mean SVM, W-SVM, Random Forests, Source-code Author Profile, and Compression-based Attribution.

1) Power Mean SVM (PMSVM): While most of the supervised machine learning-based approaches described in Sec. II make use of a traditional formulation of the SVM classifier, one downside is that it does not handle large training data sets and high-dimensional feature vectors very well. This is one of the reasons why researchers have avoided unigrams thus far for feature representation, despite their discriminative power.

An alternative is the Power Mean SVM (PMSVM) formulation [192], which was originally proposed for large-scale image classification. The power mean kernel generalizes many kernels in the additive kernel family. These kernels directly apply to applications such as image and text classification, where the data is well represented by histograms or bag-of-word models. Also, this kernel family is not very sensitive to parametrization, avoiding overfitting to the training data. It has been shown that additive kernels are more accurate in problems with millions of examples or dimensions [192].

The power mean kernel aggregates the advantages of linear SVM and non-linear additive kernel SVM. It performs faster than other additive kernels because, rather than approximating the kernel function and the feature mapping, it approximates the gradient function using polynomial regression. This approach outperforms fast linear SVM solvers (*e.g.*, LIBLINEAR SVM [54] and Coordinate Descent SVM [84]) by about 5× and the state-of-the-art additive kernel SVM training methods by about 2× (*e.g.*, HIK SVM [191]) [192]. Therefore, this kernel converges using only a small fraction of the iterations needed for the typical linear solver when faced with a large number of features and training samples.

An SVM kernel κ is additive if it can be written as a sum of a scalar function for each feature dimension d, *i.e.*, for two vectors $\vec{x}$ and $\vec{y}$,

$$\kappa(\vec{x}, \vec{y}) = \sum_{i=1}^{d} \kappa(x_i, y_i), \tag{1}$$

and a power mean function M_p is defined by a real number $p \in \mathbb{R}$ and a set of positive numbers $x_1, \ldots, x_n \in \mathbb{R}$:

$$M_p(x_1, \ldots, x_n) = \left(\frac{\sum_{i=1}^{n} x_i^p}{n} \right)^{\frac{1}{p}}. \tag{2}$$

Many of the additive kernels are special cases of the power mean function, including the χ^2, Histogram Intersection and Hellinger's kernels [192]. The power mean kernel for two vectors $\vec{x}, \vec{y} \in \mathbb{R}_+$ is a generalization of those three kernels:

$$M_p(\vec{x}, \vec{y}) = \sum_{i=1}^{d} M_p(x_i, y_i). \tag{3}$$

Wu proved in [192] that the kernel is well defined for any value of p. Usually, this formulation would lead to higher training times, but the PMSVM algorithm uses the coordinate descent method with a gradient approximation to solve the dual SVM problem. As a result, training is also faster and the approximation avoids overfitting to the training data [192].

2) W-SVM for Open Set Attribution: One of the best performing algorithms for open set supervised machine learning is currently the W-SVM [163], a Weibull-based formulation that combines a 1-Class SVM with a binary SVM, both with non-linear kernels. Why does such an algorithm help us for this

problem? First, a binary model gives us an explicit class for rejection in the authorship verification scenario we introduced in Sec. III-C. Second, when Weibull modeling is coupled with a 1-Class SVM with a radial basis function kernel, it can be proved that the probability of class membership decreases in value as points move from known training data toward open space (See the proofs in Sec. 3 of [163]). Third, the Weibull distribution provides better probabilistic modeling at the decision boundary for a binary SVM. The key to the algorithm's effectiveness in both cases is the use of the statistical extreme value theory (EVT) [43], from which the Weibull distribution is derived.

A problem such as open set authorship attribution is difficult because there are often small interclass distances in the feature space – in some cases, an author's style is very similar to that of other authors, as a function of education, influence or culture. The W-SVM ensures that the probability models do not treat data at the decision boundaries as low probability members of a class, where separation between different authors in a raw distance sense may be close. The W-SVM training algorithm consists of four distinct steps split into two different classification regimes: 1-Class and Binary. These steps can be summarized as follows:

- 1-Class RBF SVM Training. The objective of the 1-Class SVM [165] is to find the best margin with respect to the origin. The resulting binary classification function f^o after training takes the value $+1$ in a region capturing most of the training data points, and -1 elsewhere. For authorship verification, this model is trained only with pairs of samples by the same author.
- 1-Class RBF SVM EVT Calibration. The probability of class inclusion for a 1-Class SVM can be modeled by fitting a Weibull distribution to scores generated by classifying the training data $\{x_1, x_2, \ldots, x_m\}$ using the corresponding trained model f^o. This provides a set of scores from which the extrema (a sampling of the tail not exceeding 50% of the overall number of scores) are used for modeling. If the 1-Class SVM predicts $P(y|f^o(x)) > \delta_\tau$ via the Weibull Cumulative Distribution Function (CDF), even with a very low threshold δ_τ, that a given input x is a member of class y, then we will consider the binary classifier's estimates. A rejection at this step means there is likely no relationship between the two samples of text being compared.
- Binary RBF SVM Training. The 1-Class SVM serves as a good conditioner for decision making, but it is well known that the 1-Class formulation tends to overfit the training data from the positive class [125], [164], [199]. Some knowledge of known negative pairs during training improves discrimination by enforcing separation between known classes. This means that when a model f is trained with positive pairs from the same author, and negative pairs from different authors, it can generalize to unseen samples much more effectively than the 1-Class SVM.
- Binary RBF SVM EVT Calibration. Different from the 1-Class case, EVT distributions are fit separately to the positive and the negative scores from f. To produce a probability score for an SVM decision $f(x)$, two CDFs are used. Given a test sample x, two independent estimates for $P(y|f(x))$ are possible: P_η, based on the Weibull CDF derived from the matching pair scores, and P_ψ based on the reverse Weibull CDF derived from the non-matching pair scores, which is equivalent to rejecting the Weibull fitting on the non-matching class scores.

3) Random Forests (RFs): Random Forests is a method which comprises a collection of classification or regression trees, each constructed from a random resampling of the original training set. Following the notation provided in [25], let a training set be $\mathcal{L} = \{(\boldsymbol{x_i}, y_i),\ i = 1, 2, ..., N\}$, where N is the number of samples, $\boldsymbol{x_i}$ is the vector of attributes and $y_i \in \{1, 2, ..., C\}$ is the n-th example in the training set. Random Forests rely on the concepts of bootstrap aggregation and bagging applied to tree learners.

With a training set $\mathcal{L}$, the bootstrapping consists of repeatedly selecting random samples with replacement from the training set and fitting different trees to such samples. This process is repeated B times. In each iteration b out of B, we sample with replacement N examples from $\mathcal{L}$, creating $\mathcal{L}_b$, and train a classification tree[16] f_b on $\mathcal{L}_b$. After training, we can predict the outcome of unseen examples $\mathbf{x}_t$ by majority voting considering the individual classification trees on $\mathbf{x}_t$. In addition to the bootstrapping process described above, RFs also use a modified tree learning algorithm, which selects a random subset of the features for each candidate split (tree) in the learning process, a process oftentimes referred to as "feature bagging." For more information about Random Forests and their properties, we refer the reader to [25].

Besides classification, we can also consider the use of Random Forests for measuring the importance of the features at hand. The idea is to determine how the different features contribute to the final attribution process. To measure the importance of each feature f_i in a vector $\vec{v} \in R^n$, the values of the features in the training data are permuted and the error is computed on this perturbed data set for each tree in the forest. The importance score for f_i is computed by averaging the difference in the classification error before and after the permutation over all trees in the forest. The score is then normalized by the standard deviation of these differences.

For a more meaningful understanding of feature importance, we can group sets of features of the same type and cast a vote for them each time a feature of that group appears in the 100 most important features. This gives us an idea of the importance of a given group, instead of just individual features. For instance, suppose a feature vector is composed of four different sets of features, *e.g.*, 1:4-grams at the word-level. Assuming we focus on the 100 most important features for each set, each time a feature of a set appears in the top 100, it casts a vote (weighted by the rank position) to that set. If only features from the unigram set, for instance, receive votes, the final set importance will indicate 100% importance for the unigram set and zero importance for the other sets.

4) Source-Code Author Profile (SCAP): The methods described thus far are *instance-based*, meaning that each tweet is treated as a separate instance of an author's style.

[16]This process can also be applied to create regression trees.

In contrast, *profile-based* methods first concatenate all available training tweets per author and then extract a single representation from them attempting to collectively describe the author's profile. In the evaluation phase, each tweet of unknown authorship is separately processed and attributed to the author with which it is most similar [175]. Since this training phase is very simple, an inherent advantage of profile-based approaches is that they can easily be scaled to additional candidate authors. A typical example of the profile-based approach is SCAP, an authorship attribution method originally proposed for source code [59], [60] that has also been applied to Twitter with promising results [116].

SCAP builds a profile for each author that is based on the k most frequent character-level n-grams in the texts of that author. Each evaluation text is also represented using the list of its k most frequent character-level n-grams, with attribution decisions made based on which author it shares the most n-grams with. In other words, the similarity of a text to each candidate author is measured by the intersection of their profiles. This method has two parameters that have to be fine-tuned according to the specific domain, genre, and language of the documents. In this article, we followed the suggestion of Layton et al. [116] that maximal values of k (*i.e.*, all available character-level n-grams are included in the profiles) are appropriate for this domain. In addition, they examined several n values for tweets in English and the best results were provided for $n = 4$. Thus, we also use character-level 4-grams for the experiments described in Sec. IV.

5) Compression-Based Attribution: A popular and effective method for authorship attribution is based on compression tools [16], [37], [100], [183]. The main idea is that a text of unknown authorship is more likely to be effectively compressed with other texts of its true author rather than with texts of other authors. Such an approach can easily be implemented using off-the-shelf text compression algorithms like rar, bzip2, and gzip [100], [145]. Compression-based methods do not extract a concrete representation of texts with clearly defined features. They are usually based on character sequences repeatedly used within texts, and can identify common patterns between the unknown texts and the candidate authors. The most successful compression-based methods follow a profile-based paradigm.

For this article, we implemented and used a compression-based method originally proposed by Teahan and Harper [183]. This method uses Prediction by Partial Matching (PPM), one of the most effective lossless text compression schemes, to compress the concatenation of all available training texts per author. Then, for a given document of unknown authorship D, it calculates the *document cross-entropy* that corresponds to the average number of bits per symbol to encode the document using the author's model,

$$H(L, p_A, D) = -\frac{1}{n} \log_2 p_A(D), \tag{4}$$

where p_A is the PPM model extracted from documents of author A in language L, and D is a sequence of n symbols in that language. The candidate author that minimizes document cross-entropy is the most likely author of D. An important parameter that needs to be set in this method is the order of the PPM model that corresponds to the fixed order context of a Markov approximation. Following indications from previous work on authorship attribution based on this method [183], we used a fixed order context of five. This model is henceforth called PPM-5.

It should be noted that the implementations of PPM-5 and SCAP methods are slightly different with respect to the text pre-processing stage as described in Sec. III-B2. Instead of replacing numbers, dates, and timestamps with separate symbols, each digit of a token that does not include letter characters is replaced by the same symbol. Let that digit symbol be "D." For example, the tokens "11:00", "2014", "12/12/2014" would be replaced by "DD:DD", "DDDD", and "DD/DD/DDDD" while the tokens "1st" and "8tracks" will remain the same since they contain some letter characters.

IV. Experimental Results

Schwartz et al. [166] introduced a viable validation regime for authorship attribution methods targeted at social media. Thus, we use it here to examine the approaches we have described above. This validation looks at two important aspects of the problem: the impact of varying training set sizes, and the impact of varying numbers of authors. This type of scalability assessment is a key element of authorship attribution evaluation [121]. Further, we will look at a separate validation regime [164] for open set authorship verification, which lets us control the amount of unseen data at evaluation time.

In this section, we describe the experiments that we performed to examine the approaches introduced in Sec. III for large-scale attribution tasks, which are common in this type of forensics work:

1) A comparison of the performance of various feature types using a fixed pool of 50 Twitter users using PMSVM and Random Forests classifiers;
2) A comparison of the performance of various classifiers by varying the number of Twitter users and feature types;
3) An assessment of algorithm efficiency and search-space reduction;
4) An analysis on feature importance given a fusion method using different groups of features;
5) A comparison of different methodologies for open set attribution.

A. Data Set and Pre-Processing

To our knowledge, no publicly available data set exists for authorship attribution applied to social media forensics. Moreover, the restrictive terms of use put in place by the major social networks prohibit the dissemination of such data sets. Thus, data from the existing research described in Sec. II are largely inaccessible to us. In response to this, we created our own large-scale data set that was designed with algorithm scalability evaluations in mind.

The set was constructed by searching Twitter for the English language function words present in [116, Appendix A],

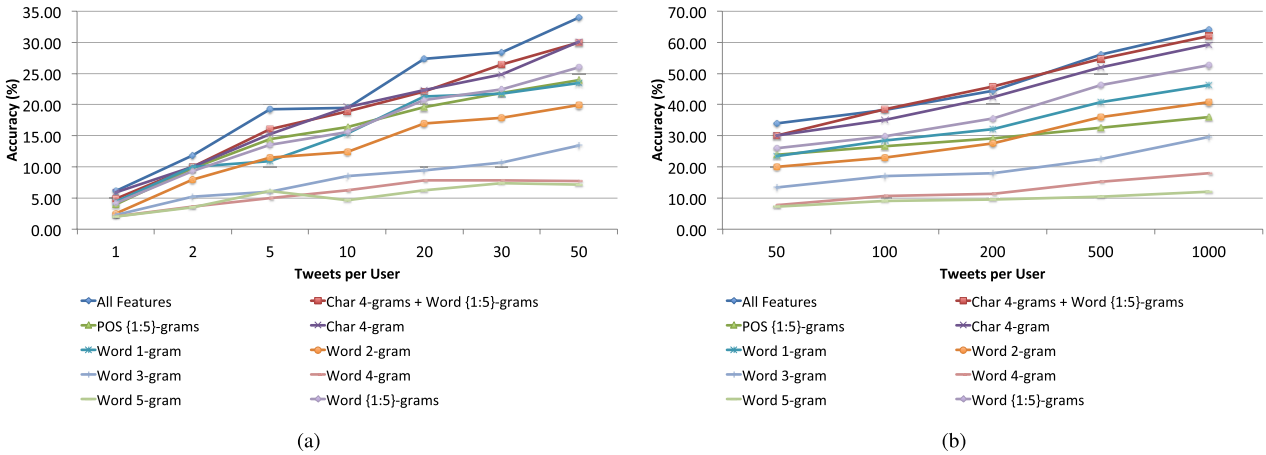


Fig. 2. Relevance of each feature used by itself, or combined with other features for setups including 50 different Twitter users. Here we used the PMSVM classifier (Sec. III-D.1) in all cases. The character-level 4-gram is the best individual feature set, and word-level 5-gram is the sparsest feature set. Unigrams (`Word 1-gram`) also yields good performance, although neglected in the literature thus far (Sec. IV-E shows this feature importance in a fusion scenario). The more advanced tagging method, POS n-grams, does not yield good results in isolation but when combined with character- and word-level n-grams, it contributes to the best overall feature set. Finally, achieving good attribution results is still an open problem when training data is very scarce (small sample size problem). (a) Setup with 50 users training with 1..50 messages per user. (b) Setup with 50 users training with 50..1,000 messages per user.

yielding results from English speaking public users.[17] These results were used to build a list of public users from which we could extract tweets by using the Twitter API. We collected ten million tweets from 10,000 authors[18] over the course of six months in 2014. Each tweet is at most 140-character long and includes hashtags, user references and links.

Although we could not use data from other researchers due to the restrictions placed on us by Twitter's terms of service, the data set was created with the same methods used by other authors [116], [166]. While we cannot release the actual messages, we will release all of the features derived from them after this paper is published in an effort to provide the community with a standardized resource for evaluation.

Pre-processing of each tweet includes removing all non-English tweets, tweets with less than four words, and tweets marked as retweets or any tweet containing the meta tag RT. As we discussed previously, for most of the methods we replace numbers, URLs, dates and timestamps by the meta tags NUM, URL, DAT, and TIM, respectively. Moreover, the hashtags and user references were replaced, since they enrich the feature set for authorship attribution in such a way that makes the task artificially easier yet ultimately unreliable [116] For PPM-5 and SCAP, each digit of a token that does not include letters is instead replaced by the same symbol.

The data set was partitioned into training and test sets via k-fold cross validation. For Secs. IV-B-IV-E, each experiment was repeated 10 times and the authors considered in each fold are chosen at random. Average classification accuracy is reported as a summary statistic over the 100 (10×10) different results. Similarly, the open set experiments in Sec. IV-F make use cross-validation, but with five folds.

[17]Public user data is not subject to any form of protection via copyright law. However, the Twitter data extraction policy still applies, therefore this data can be used but not shared.

[18]The Twitter API only allows the extraction of the most recent 3,200 tweets from a user.

B. Comparison of Different Feature Types

In order to assess the usefulness of feature types, we first performed independent tests with two classifiers and different sets of features: word-level n-grams (for $n \in \{1, \ldots, 5\}$), character-level 4-grams, and POS n-grams (for $n \in \{1, \ldots, 5\}$). Figs. 2(a:b) show results for different methods when considering 50 authors and a varied number of training messages per author.

The classification using only a few micro messages is still an open problem with the performance steadily improving as more messages are present in the training pool. This is an excellent research opportunity for the community: how to deal with the small sample size problem in micro-message authorship attribution.

The attribution performance is improved as more than 50 messages are available for training. In this case, the PMSVM, even using the sparsest feature set — which, for this experiment, is word 5-grams — is still better than random chance for 50 users (2% accuracy). In addition, the most relevant independent feature set is the character-level 4-grams. This is expected, because that feature set captures relevant style markers like repetitions in punctuation and emoticons.

The figure also shows that the unigrams are relevant features. When combined with other word-level n-grams (see `Word {1:5}-grams`), they help produce an accuracy that comes close to matching character-level 4-grams. This reflects user preferences for some words that they are more comfortable with, which are not captured by character-level n-grams due to their size or likelihood of appearance with other word prefixes and suffixes.

Another interesting result is the effect of using POS n-grams. While this feature by itself (see `POS {1:5}-grams`) is not as strong as unigrams or character-level 4-grams, it complements the other features when used in combinations (see `All Features`). This combined representation reflects many author preferences, from words

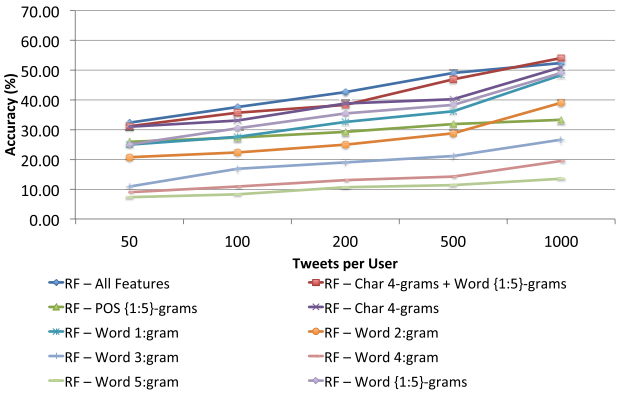


Fig. 3. Relevance of each feature used by itself, or combined with other features for tests including 50 different Twitter users and varying number of tweets per user. Here we used the Random Forest classifier (Sec. III-D.3) in all cases. The obtained results, in general, are below the ones obtained with PMSVM (see Fig. 2).

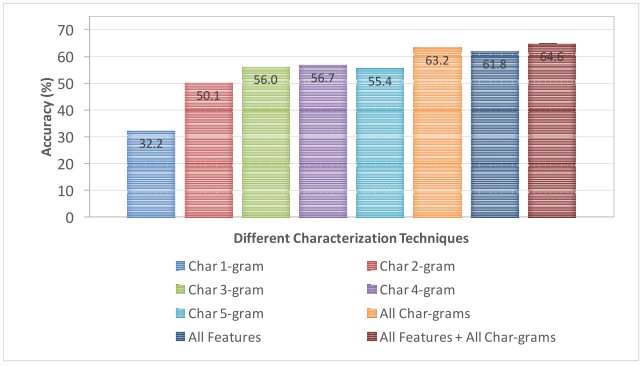


Fig. 4. Effects of choosing different character n-grams and their combination.

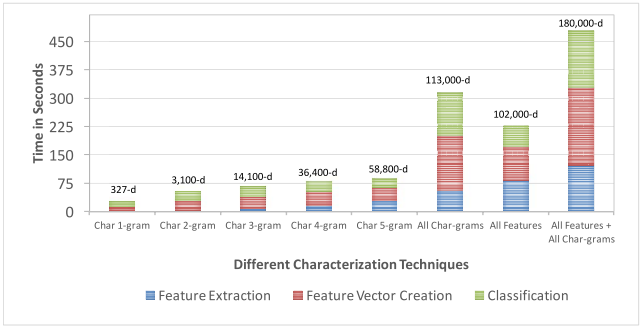


Fig. 5. Training time breakdown into three major tasks: feature extraction, vector creation and classification learning. On top of each stacked bar, we show the final average feature vector length for each method.

to small characteristic sentences and expressions to patterns of change in parts of speech. Fig. 3 shows similar trends for the Random Forest classifier, although with relatively lower accuracy for all features. Note that RFs are normally more expensive than the linear PMSVM learning model. Error bars were calculated for both plots, but were too small to visualize.

At this point, one could wonder what would be the impact of using other character n-grams instead of only character 4-grams in the attribution task. Our choice for the character 4-grams was motivated by previous work in the area [116], [158], [166]. In addition, previous experiments have shown that these features reasonably capture the idiosyncrasies commonly used in Internet parlance albeit leading to a higher processing and memory footprint, which may prevent their deployment when considering large pools of messages and authors. Therefore we now turn our attention to comparing different character n-grams feature sets.

In all the experiments, we consider PMSVM as the base classifier over different character n-grams feature sets, with $n \in 1 \ldots 5$. The test setup comprises 50 users and 500 tweets per user, all of them randomly sampled. This process was repeated 10 times in order to have 10 different runs for each set of users/messages. The reported results are the average classification accuracies, and computational time and average feature vector length required to analyze each pool of authors.

Fig. 4 depicts the classification results for different char n-grams, while Fig. 5 shows the results for the training time spent by the different methods and the length of the final generated feature vector. These experiments were performed on a 12-core i7-5820K CPU with 3.30GHz, and 96GB of RAM memory. The reported computational times are related to training an attribution model for 50 authors with 500 messages per author. This means the reported times refer to the total processing time for approximately $50 \times 500 = 25{,}000$ messages.

First of all, aligned with previous studies in the literature, `Char 4-grams` shows the best individual performance in terms of accuracy, followed closely by the `Char 3-grams` features. However, `Char 3-grams` offers a competitive accuracy with less than half of the number of features (c.f. Fig. 5). In turn, `Char 5-grams`, although encompassing much more features, is outweighed by `Char 3-grams` and `Char 4-grams`.

Combining some of the char n-grams into a more complex model also shows potential for future investigations with `All Char-grams` outperforming all individual char-gram methods and also improving the best result obtained in Fig. 2 when incorporating additional char n-grams (`All Features + All Char-grams`) although at the cost of almost doubling the size of the feature set representation. In this case, the `All Features` technique reported in Fig. 2 has approximately 100,000 features while its augmented version with all char n-grams variations accounts for almost 180,000 features. This doubling effect might be even worse when dealing with more authors and messages in the training stage, as more characters and words combinations would be present, potentially exploding the number of features and rendering the analysis intractable.

We close this section concluding that the strategy of using all word/POS-tags n-grams and only the character 4-grams offers a competitive performance with significant less cost in terms of space and processing time. Regardless of these requirements, when dealing with Internet parlance in social media like Twitter (full of emoticons, emojis, slangs and exaggerated punctuation), character n-grams play a crucial role in the author discrimination and should be more studied.

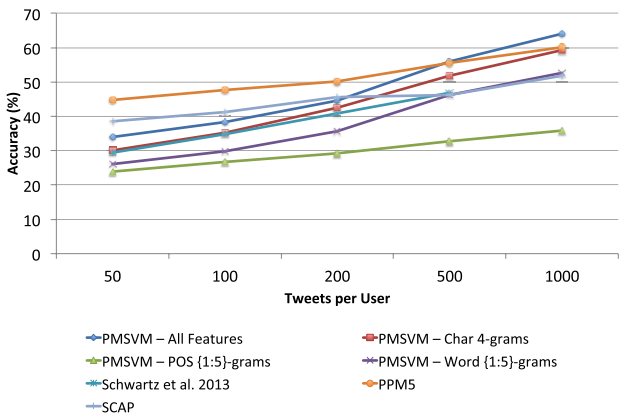


Fig. 6. Classification accuracy comparison between PMSVM trained with all of the features from Fig. 2, SCAP (Sec. III-D4), PPM-5 (Sec. III-D5) and the method of Schwartz et al. [166]. All experiments consider 50 users and a variable number of tweets per user. Some methods trade off in performance with increasing numbers of tweets per user.

C. Comparison of Different Classifiers

While `Char 4-grams` can help improve classification accuracy, their use generates a greater number of features to process. This, coupled with the need for large amounts of training data due to the small size of each sample, means that we expect any approach making use of a common classifier like linear SVM to be slow in the best case, and not converge in a reasonable amount of time in the worst case. This is where the PMSVM, SCAP, and PPM-5 methods show an advantage. For instance, relying on PMSVM, a larger number of features has demonstrably less impact on the classification time, and significantly improves accuracy.

When considering all character- and word-level n-grams, as well as POS n-grams (see `PMSVM - All Features` in Fig. 6) as part of the feature representation, PMSVM outperforms the method proposed by Schwartz et al. [166] by an average margin of 10 percentage points for 50 users regardless of the number of training tweets per user. Moreover, the linear SVM classifier used in the method of Schwartz et al. is not as efficient as PMSVM. Therefore, for all additional experiments with the feature sets proposed by Schwartz et al. (`Char 4-grams and Word {1:5}-grams`), we opted to replace the linear SVM classifier with PMSVM.

Fig. 6 also depicts the performance of the SCAP and PPM-5 methods. The first thing to notice is that PPM-5 outperforms the other methods when 500 or fewer training tweets per user are present. However, as more messages per user are added to the training set, the representation incorporating character- and word-level n-grams, as well as POS n-grams presents better results (see `PMSVM - All Features`). This is expected, as more data is available to capture the author's style. In addition, the PPM-5 and SCAP methods perform better than the method of Schwartz et al., especially for small numbers of tweets per user.

Another interesting trend present in the results is the steep slope in the classification curve for `PMSVM - All Features`. This suggests that the more tweets we can gather for training, the better for this configuration. In a way, this is aligned with the classical literature on authorship attribution, which has already shown that long-form writing is easier to assess than short-form writing. If a very large set of small samples is collected for a specific author, it functions similarly to a long-form text, from which many samples can be produced for more traditional authorship attribution tasks. However, for attribution tasks involving social media, we cannot set any expectations beyond the size of the samples from the medium itself (*e.g.*, 140 characters for Twitter).

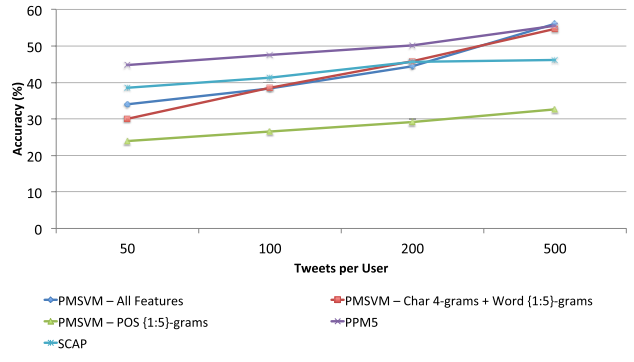
D. Efficiency and Search-Space Reduction

We now turn our attention to the effect of adding more authors (or classes) of interest and evaluate how the methods handle an increasing number — up to 1,000 authors per test. Fig. 7(a:c) shows the results for these experiments. Error bars were calculated for both plots, but were too small to visualize. PMSVM handles hundreds of users while continuing to increase in accuracy as more training messages are used per user. This shows that investigators could keep adding training examples if they exist in an attempt to improve accuracy — something that is not possible with all classifiers. With small numbers of tweets per user, and many users, the classification accuracy for all methods is significantly lower compared to more favorable settings using fewer users.

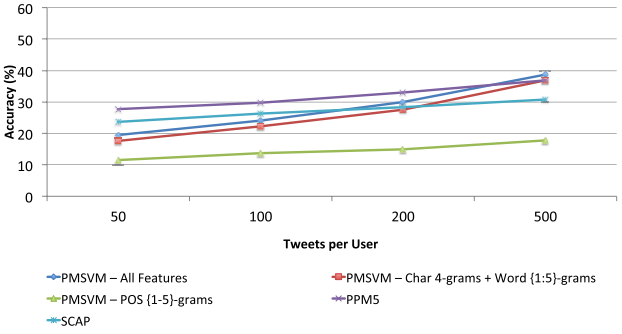
When considering the SCAP and PPM-5 methods, SCAP is competitive with PMSVM only when very few tweets per user are available for training. On the other hand, PPM-5 appears to be more capable than PMSVM in that same scenario. However, as already observed in the case of 50 authors, when a large number of tweets per author is available, the PMSVM method using character- and word-level n-grams fused with POS-tag n-grams is typically more effective. In all scenarios, however, it is clear that there is plenty of room for improvement as the best attribution results are still below 40 percentage points for our experiments with 500 or more authors. Although these results are much better than chance they are far from useful as courtroom evidence, especially when considering the most difficult scenario in Fig. 7(c).

Although traditional authorship attribution intends to find the exact author of a given text, achieving that goal may not always be possible when considering social media. By their very nature, short messages are very difficult to attribute to a single author. Instead of searching for the most probable author, we can rank all of the known authors according to the output function, and then show how well we can reduce the search space of the problem. We tested this method for the PMSVM with 500 Twitter users and a varying number of tweets per user. The Cumulative Match Curve (CMC) in Fig. 8 shows the accuracy of finding the author of a tweet considering the top N users.

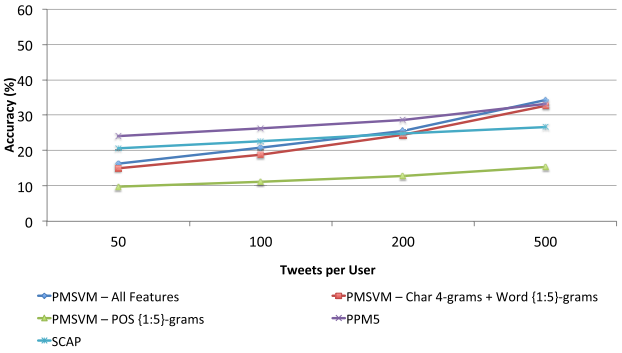
The classifier starts with a classification accuracy below 25% when using 50 tweets per user (the purple curve in Fig. 8). Considering the random baseline of 0.2% (randomly guessing the author of a Tweet in a universe with 500 users), this result directly conveys to the research community how difficult the problem is. In more than 65% of the cases, the correct user will be among the top 50 users (out of 500) when we use 200 tweets per user (the red curve in Fig. 8).



(a)



(b)



(c)

Fig. 7. Classification accuracy comparison between PMSVM trained with all of the features from Fig. 2, the method of Schwartz et al. [166], PPM-5 and SCAP for 50, 500, and 1,000 test authors. Note that it was not possible to run some of the methods for 500+ users on a standalone Intel i7 5820k machine with 96GB of memory. Cloud-based infrastructure is a possible workaround for this. (a) Performance for 50 test authors. (b) Performance for 500 test authors. (c) Performance for 1,000 test authors.

Assuming generalization, this would reduce the number of suspects to 10% of the original size in more than half of the scenarios. 500 messages per user (blue curve) brings a further reduction.

E. Feature Importance

When combining different features to solve a problem it is natural to ask which ones are effectively contributing to the solution. As we discussed earlier, there are different ways of assessing feature importance in a classification scenario. Here we chose the random forests classifier to perform this task according to the procedure described in Sec. III-D3.

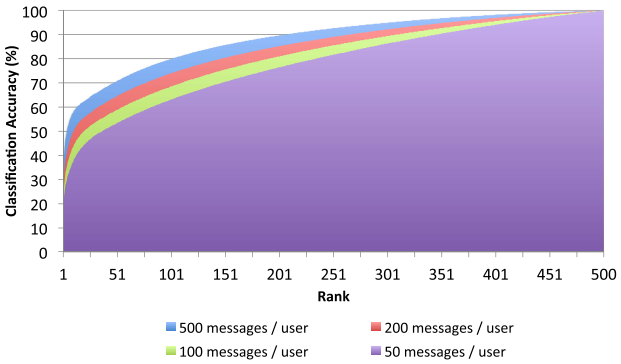


Fig. 8. Cumulative Match Curves for PMSVM for 500 Twitter users with a varying number of tweets per user.

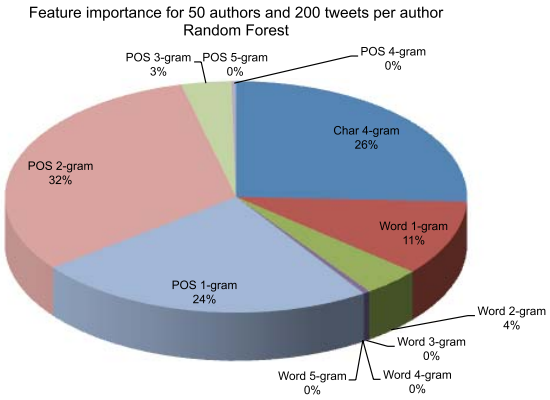


Fig. 9. Feature importance determined by a Random Forest classifier for a fused vector consisting of different word-level n-grams, character-level 4-grams, and different POS n-grams for 50 authors and 200 training tweets per author. Note that the entries marked as 0% are actually very close to zero and not literally zero.

Fig. 9 shows the result of analyzing 50 authors and 200 tweets per author in the training set. We consider the best attribution setup, which takes in features from word-level n-grams (for $n \in \{1, \ldots, 5\}$), character-level 4-grams, and POS n-grams (for $n \in \{1, \ldots, 5\}$). The random forest's inherent ability to assign feature importance gives us a direct indication of the weights for each feature type. The main conclusion from the results discussed in Sec. IV-B was that combining different feature groups leads to a better attribution accuracy even though the individual performance of some of them was poor (*e.g.*, `Word 5-grams` in Fig. 2). The feature importance values help us analyze this conclusion in more detail.

The results in Fig. 9 are in line with those in Sec. IV-B. An interesting observation, however, is that although POS n-grams are not very effective by themselves (see `POS {1:5}-grams` in Fig. 2), when combined with other feature types they are weighted more heavily in importance. Together, POS 1- and 2-grams represent over 50% of the feature importance. Character-level 4-grams, by themselves, led to good classification results in one of our earlier experiments (see `Char 4-gram` in Fig. 2) and this is reflected in the feature set's relatively high importance of 26% in this experiment. Unigrams (`Word 1-gram` and `POS 1-gram`)

represent 43% of the feature importance. Although somewhat neglected in the literature, unigrams can have merit in some cases. Finally, some feature groups were not judged to be important (word-level 3-, 4-, and 5-grams along with POS 4- and 5-grams). This type of anlaysis could help in a posterior analysis of feature complementarity and subsequent reduction. In this case, POS 1- and 2-grams along with character-level 4-grams and word-level 1-grams account for 93% of the feature importance.

F. Open Set Attribution

Thus far, we have only evaluated feature and classifier approaches in a closed set context, where all authors seen at testing time were known at training time. What kind of behavior can be expected from classifiers operating in an open set context, where large numbers of unknown authors posting to social media will appear as inputs? For this experiment, we follow the feature extraction and open set authorship verification methodology of Sec. III-D2, computing results over five randomly sampled folds at each evaluation point.

50 known authors were randomly chosen from the corpus and fixed across folds for this experiment. For each fold, positive training samples for these authors were generated by randomly sampling 600 matching tweet pairs (*i.e.*, two different tweets from the same author), and negative samples generated by sampling 600 non-matching tweet pairs (*i.e.*, two different tweets from two different authors). This training data was used to create three separate verification models that can tell if any two tweets came from the same author. For the models, the open set-specific W-SVM [163] classifier was considered, along with RBF SVM [38] and Logistic Regression [54] for comparison. Parameters for each classifier were tuned via cross-validation during training. A visual overview of the training process is shown at the top of Fig. 10.

The first test set establishes a closed set baseline. It contains 200 positive and 200 negative tweet pairs for each fold, all coming from the 50 authors known at training time, with no samples overlapping with the training set. It is marked "0" on the x-axis of Fig. 11. The subsequent tests start with 100 positive and 100 negative tweet pairs from a new sampling of tweet pairs from the 50 known authors, and add, in increments, 100 positive and 100 negative tweet pairs from an additional 50 unknown authors. Three of these open set increments are generated, from 50 to 150 unknown authors. Thus the test sets grow in both size and number of unknown authors considered. A visual overview of the testing process is shown in the bottom of Fig. 10. The results for each classifier are shown in Fig. 11.

As the problem grows to become more open by adding additional authors, accuracy for the W-SVM and RBF SVM classifiers drops accordingly until performance begins to plateau around 100 unknown authors. A slight advantage is demonstrated for the W-SVM, which is expected, given its inherent ability to minimize the risk of the unknown. Logistic regression, which does not benefit from the maximum margin principle, is significantly worse than the other two classifiers in all cases. While these results show some feasibility for

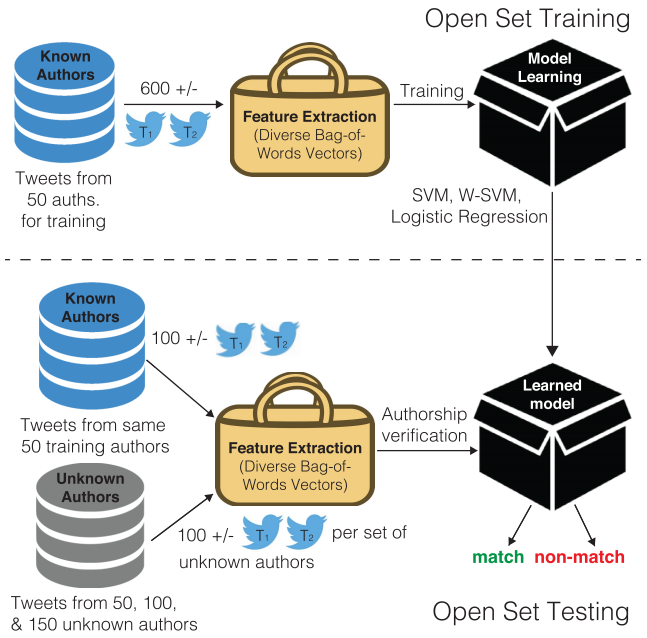


Fig. 10. An overview of open set authorship verification experiment. Pairs of tweets are sampled from sets where the authors are either known (training and testing) or unknown (just testing). A matching tweet pair consists of two tweets by the same author; a non-matching tweet pair consists of two tweets by different authors. Features from Table III are extracted with respect to the information contained in both tweets from the pair. In the training phase, a verification model is learned from matching and non-matching tweet pairs from the known authors. The testing phase evaluates matching and non-matching tweet pairs from both known and unknown authors. In a real forensic scenario, one of the two tweets in the pair during testing could be from a known author in order to make an attribution determination, or the models could be used to simply identify common authorship.

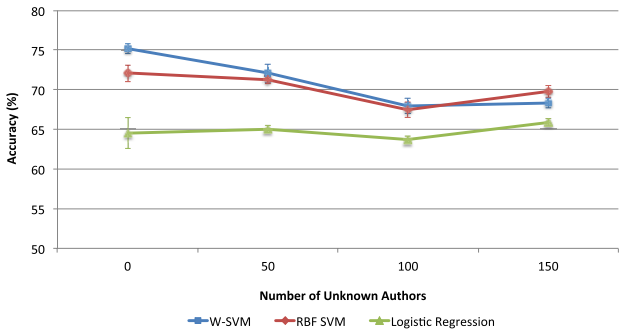


Fig. 11. Open set classification accuracy for the W-SVM [163], RBF SVM [38], and Logistic Regression [54] algorithms. In all cases, samples from 50 known authors are present. Each point on a curve reflects average classification accuracy over five-testing folds. As the number of unknown authors generating samples is increased during testing from 0 to 150, a decrease in accuracy can be seen right away (*i.e.*, the drop in accuracy between a closed set and 50 unknown authors).

classifiers that have been applied to open set problems in other domains [163], note that pair matching for verification is the most basic scenario for authorship attribution. These results make it clear that much more work must be done before full multi-class open set attribution incorporating hundreds of known authors can be achieved.

V. Conclusion and Future Directions

The enormous popularity of social media means that it is now a conduit for both legitimate and illegitimate messages

targeted at the broadest possible audience. Correspondingly, new forensic challenges have appeared related to this form of new media, triggering the need for effective solutions and requiring the attention of the information forensics community. A primary problem in this area has been authorship attribution for short messages.

In this vein, this study showed that for popular services like Twitter, we face the dilemma of simultaneously having an enormous overall corpus, yet scarcity of information for individual users. This suggests that we should consider strategies that are a bit different than traditional authorship attribution algorithms for long form writing. When working with highly constrained forms of writing like tweets, the problem size grows rapidly due to the large number of users and messages involved. One way to address this problem is to compute very low-level lexical statistics, which easily leads to high-dimensional spaces. Moreover, the problem is exacerbated by the unconventional punctuation, abbreviations, and character-based signifiers common in Internet culture. There is a need for better learning tools that will help us avoid the so-called curse of dimensionality [21].

As discussed throughout this article, there are several complementary paths one can follow when solving the authorship attribution problem for social media forensics. The methods we introduced are just the beginning, as this problem is far from being solved. This point is reinforced by the classification results we observed in Sec. IV: they are far from perfection. The PMSVM algorithm used in conjunction with diverse and complementary features is a good advance over the state-of-the-art methods, and opens the door for other researchers to explore feature sets that have been avoided out of computational concerns. In addition, the cumulative matching analysis showed that current attribution methods can greatly reduce the number of users to be analyzed in a real situation. In the rest of this section, we provide a brief rundown of some of the more difficult aspects of this problem that the community should turn its attention to.

A. Real-World Use

While expert witnesses well versed in forensic authorship attribution are being used in court [10], the results from automated authorship attribution methods are not commonplace. However, this will likely change in the next several years as the relevant technologies become more accessible. The methods we have discussed can be used directly by investigators, especially for suspect search-space reduction or helping to assemble more conventional evidence that is admissible in court.

The results presented in this article (*e.g.*, 70% accuracy for 50 authors) are just a beginning, but they do show that the techniques we introduced may be feasible to deploy when looking for clues in a real investigation. For further clues, researchers should also look closely at the graphical structure of a social network. Such an analysis might surface additional suspects that are implicated via their association to an identified author. Researchers interested in continuing or building upon our work herein can obtain the source-code from our GitHub repository: `http://tinyurl.com/zvxsavl`. Although the Twitter raw data cannot be freely shared (Twitter data extraction policy), its processed versions might be obtained under request.

B. Social Network-Scale Data

The amount of data present on social media services is growing larger each day. As a consequence, scalable classifiers are as important as discriminative features. The experiments in this article indicate that by using more training data, better results can be achieved. Methods like PMSVM, which are custom-tailored to high-dimensional spaces, represent a large improvement over prior methods using traditional learning methods.

Social network-scale data can be characterized by the three Vs of big data: volume, velocity and variety. The volume of data found on social media sites is clearly large, and the velocity at which it is delivered is remarkably fast. In this work, we only dealt with textual features, but the variety of data found on social media is much richer than just this. Future work should also look at meta-data, images, and videos posted online to further hone the accuracy of attribution by pursuing a hybrid approach that extends beyond just stylometry.

At this point, one would wonder why not just focusing on features that are related to social media messages rather than building a larger feature set as we exploited in this work. We believe that an active adversary would more easily attack a solution if it were only based on explicit features related to social media. In the past, we have seen this enough with specific features for detecting e-mail SPAMS. Some specific features might be attacked and mimicked thus preventing proper identification of an author. The features we studied in this work take into account some elements that are native to social media, outside the realm of proper writing, such as hashtags, web links and user directed messages. However, there are still many other native features to be explored in social media, including the actual social graph of a suspect. Other meta-data such as timestamps, device records, and browser records could also be exploited, though they are often unreliable and, in some cases, easily forged. Therefore, a more generalized solution that incorporates knowledge of the problem, general and data-driven features and, possibly, network connectivity and the social network of suspects would be, certainly, more difficult to tamper with. We leave this as another opportunity for the research community to delve into.

Regardless, the need for algorithms that can exploit context is paramount. Today's machine learning algorithms are not able to identify salient pieces of information related in time and space to a tweet with any meaningful degree of accuracy. Bayesian reasoning is a good start, but more efficient probabilistic models taking into account entire social networks will be necessary in difficult attribution cases. For example, when a message appears to be unique, but is actually a newly propagating meme.

C. Dense Features

The methodology we introduced consisted of dynamic features extracted from raw text, and messages like tweets contain

few words, thus the resulting feature vectors are always sparse. In the course of this work, we experimented with standard techniques such as PCA and random selection of features to reduce the dimensionality of the feature vectors. Random selection always performed worse, no matter the size of the extracted features, and PCA failed to converge in some cases due to the overwhelming number of examples and dimensions and the limits of the available hardware.

Just to put things into perspective, PCA might lead to worse performance both in terms of accuracy and speed. The reason for the drop in accuracy is that useful information may be discarded when reducing dimensionality, while the reason for the reduction in speed is likely due to the interplay of the resulting dense representation and the classification algorithm processing it. In this article, we looked at optimal algorithms for *sparse* representations, which speed up the processing given the appropriate input.

As an example, the classification accuracy for 50 authors, 50 tweets per author in the training set, with `All Features` is 34% (see Fig. 2). The PCA-reduced version keeping 90% of the variance (from 14,500 to about 700 features) is 27.5% — a difference of nearly seven percentage points. This gap becomes smaller as more tweets are present in the training set at the cost of more computational time for PCA. For 200 tweets per author during training, PCA reduces the 50,000-d feature vector of `All Features` in Fig. 2 to 2,000 (keeping 90% variance) and the performance difference is 2.5 percentage points. Although the number of features considered is much smaller, the time to compute such a reduction needs to be considered along with the possible drop in performance.

Future work could explore other feature selection and dimensionality reduction techniques such as random projection when computational power is a bottleneck, which might improve the classification task to avoid computationally expensive sparse feature representations. When using Random Forests, which creates different sets of trees with each one using just a sub-set of features (usually $\sqrt{n}$, with n being the original number of features), the performance in terms of classification accuracy is similar but an implementation optimized for sparse vectors and additive kernels still runs faster (*e.g.*, PMSVM). For efficiency, better multi-class from binary techniques [156] could also be used instead of the expensive one-vs-all techniques.

D. Representativeness

Unlike other text classification tasks, in authorship attribution it is not always possible to assume that training and evaluation texts will share the same properties. For instance, in a forensic examination of suspicious tweets, for some of the suspects we may not find authentic tweets by them. However, we might be able to find other types of text like email messages, blog posts, etc. Certainly, the available texts by each suspect may be on a completely different topic in comparison to the texts under investigation. In all these cases, the training dataset is not representative enough with respect to the documents under investigation. Thus, unlike other text classification tasks, we need authorship attribution models that remain useful in cross-genre and cross-topic conditions. Recently, attempts have been made to estimate the effectiveness of attribution models in such challenging conditions. But the lack of large-scale data covering multiple and heterogeneous genres and topics by the same authors limits the generality of the conclusions [160], [176], [177].

Another problem that affects the representativeness of the training data is the distribution of that data over the candidate authors. In a forensic examination, when class imbalance exists (the training texts are unequally distributed over candidate authors), the suspect for whom we have plenty of training

TABLE IV

FUNCTION WORDS USED FOR CREATING THE AUTHORSHIP DATA SET

a	about	above	after	all	although
am	among	an	and	another	any
anybody	anyone	anything	are	around	as
at	be	because	before	behind	below
beside	between	both	but	by	can
cos	do	down	each	either	enough
every	everybody	everyone	everything	few	following
for	from	have	he	her	him
i	if	in	including	inside	into
is	it	its	latter	less	like
little	lots	many	me	more	most
much	must	my	near	need	neither
no	nobody	none	nor	nothing	of
off	on	once	one	onto	opposite
or	our	outside	over	own	past
per	plenty	plus	regarding	same	several
she	should	since	so	some	somebody
someone	something	such	than	that	the
their	them	these	they	this	those
though	through	till	to	toward	towards
under	unless	unlike	until	up	upon
us	used	via	we	what	whatever
when	where	whether	which	while	who
whoever	whom	whose	will	with	within
without	worth	would	yes	you	

TABLE V

PART-OF-SPEECH TAGS CONSIDERED IN THIS WORK (ADAPTED FROM [65])

Tag	Meaning
A	adjective (J*)
B	proper noun (NNP, NNPS)
C	interjection (UH)
D	determiner (WDT, DT, WP$, PRP$)
E	emoticon
F	coordinating conjunction (CC)
G	other abbreviations, foreign words, possessive endings, symbols, garbage (FW, POS, SYM, LS)
H	hashtag (indicates topic/category for tweet)
I	at-mention (indicates another user as a recipient of a tweet)
J	discourse marker, indications of continuation of a message across multiple tweets
K	numeral
L	nominal + verbal
M	proper noun + verbal
N	common noun (NN, NNS)
O	pronoun (personal/WH; not possessive; PRP, WP)
P	pre- or postposition, or subordinating conjunction (IN, TO)
R	adverb (R*, WRB)
S	nominal + possessive
T	verb particle (RP)
U	URL or email address
V	verb incl. copula, auxiliaries (V*,MD)
W	punctuation
X	existential there, predeterminers (EX, PDT)
Y	X + verbal
Z	proper noun + possessive

TABLE VI
UNSUPERVISED CLASSIFICATION FOR AUTHORSHIP ATTRIBUTION

Source	Features Used	Classifier	Corpus
Burrows 1987 [28], 1989 [29], 1992 [30], [31]	Small set of function words	Multivariate analysis	English prose
Ledger and Merriam 1994 [117]	Character-level n-grams	Multivariate analysis	English drama
Mealand 1995 [132]	Function words	Multivariate analysis	Greek prose
Holmes and Forsyth 1995 [75]	Words	Multivariate analysis	English prose
Baayen et al. 1996 [13]	Syntax	Multivariate analysis	English prose
Merriam 1996 [137]	Function words	Multivariate analysis	English drama
Tweedie and Baayen 1998 [186]	Function words	Multivariate analysis	Latin prose
Binongo and Smith 1999 [20]	Function words	Multivariate analysis	English drama
Holmes et al. 2001 [76]	Function words	Multivariate analysis	Journalism
Baayen et al. 2002 [12]	Function words	Multivariate analysis	Dutch prose
Hoover 2003 [80]	Word-level n-grams	Multivariate analysis	English prose
Binongo 2003 [19]	Function words	Multivariate analysis	English prose
Kestemont et al. 2015 [99]	Function words	Multivariate analysis	Latin prose

TABLE VII
DISTANCE-BASED AND SIMPLE MODEL-BASED CLASSIFICATION FOR AUTHORSHIP ATTRIBUTION (SORTED BY CLASSIFICATION TYPE)

Bissel 1995 [22]	Weighted cum. sum of lexical statistics	Statistical hypothesis test	English prose
Somers 1998 [170]	Weighted cum. sum of lexical statistics	Statistical hypothesis test	English prose
Chaski 2001 [39]	Syntax and punctuation	Statistical hypothesis test	English prose
Somers and Tweedie 2003 [171]	Weighted cum. sum of lexical statistics	Statistical hypothesis test	English prose
Merriam 1979 [134], 1980 [135], 1982 [136]	Word positions	Statistical hypothesis test	English drama
Grieve 2007 [68]	Words, syntactic structures, and character-level n-grams	Statistical hypothesis test	English prose
Kjell 1994 [104]	Character-level n-grams	Cosine Similarity	English prose
Hoover 2004 [82]	Function words	Delta	English prose and poetry
Kestemont et al. 2015 [99]	Function words	Delta	Latin prose
Kukushkina et al. 2001 [114]	Character-level n-grams and grammatical word classes	Markov models	Russian prose
Khmelev and Tweedie 2002 [101]	Character-level n-grams	Markov models	English prose
Khmelev and Teahan 2003 [100]	Character-level n-grams	Markov models	English journalism
Zhao et al. 2006 [197]	Parts of speech	Kullback-Leibler Divergence	English novels and journalism
Zhao and Zobel 2007 [196]	Function words and part-of-speech tags	Kullback-Leibler Divergence	English prose and drama
Teahan and Harper 2003 [183]	Character streams	Cross-entropy	English journalism
Juola and Baayen 2005 [90]	Character streams and function words	Cross-entropy	Dutch prose
Kjell et al. 1995 [105]	Character-level n-grams	K-NN	English journalism
Hoorn et al. 1999 [78]	Character-level n-grams	K-NN and Naïve-Bayes	Dutch poetry
Keselj et al. 2003 [97]	Character-level n-grams	K-NN	English prose and Greek journalism
Zhao and Zobel 2005 [195]	Function words	K-NN and Naïve-Bayes	English journalism
Mosteller and Wallace 1964 [141]	Small set of function words	Naïve-Bayes	English prose
Clement and Sharp 2003 [42]	Character-level n-grams	Naïve-Bayes	English prose
Peng et al. 2004 [150]	Character- and word-level n-grams	Naïve-Bayes	Greek journalism
Savoy 2013 [161]	Function words	Naïve-Bayes	English prose
Stamatatos et al. 2000 [179], 2001 [180]	Syntactic chunks	Linear Discrim. Analysis	Greek journalism
Chaski 2005 [40]	Character- and word-level n-grams	Linear Discrim. Analysis	English prose
Jockers and Witten 2010 [87]	Words and word-level bigrams	Regularized Discrim. Analysis, Delta, and K-NN	English prose

samples should not be considered more likely than the suspect for whom only a few training documents are available. The attribution models should make efforts not to favor the majority authors even when the imbalance ratio is extremely high. Certain machine learning methods dealing with the class imbalance problem can be adopted for authorship attribution [53], [174] and this problem needs to be studied more thoroughly.

E. Open Set Recognition

The confounding problems related to the authorship attribution task such as hoaxes, impersonations and identity theft lead to an open set scenario, whereby a candidate author may not be among the known suspects [181]. As we learned in this article, a traditional multi-class classifier will always return an answer pointing to a known suspect, which will not be correct in many instances. This suggests that the effort should be placed on reducing and prioritizing the known suspects rather than always pointing to a single culprit.

In open set recognition problems, we face the dilemma of not having information about all of the existing authors on social media. We must always learn from an incomplete set of authors when training a classifier. However, in the testing phase, many different authors might appear, even authors that

TABLE VIII
MODEL-BASED CLASSIFICATION FOR AUTHORSHIP ATTRIBUTION (SORTED BY CLASSIFICATION TYPE)

Matthews and Merriam 1994 [129]	Small set of function words	Neural Networks	English drama
Merriam and Matthews 1994 [138]	Function words	Neural Networks	English drama
Kjell 1994 [102], [103]	Character-level n-grams	Neural Networks	English prose
Lowe and Matthews 1995 [120]	Function words	Neural Networks	English drama
Martindale and McKenzie 1995 [127]	Words	Neural Networks	English prose
Kjell et al. 1995 [105]	Character-level n-grams	Neural Networks	English journalism
Tweedie et al. 1996 [187]	Function words	Neural Networks	English prose
Hoorn et al. 1999 [78]	Character-level n-grams	Neural Networks	Dutch poetry
Waugh et al. 2000 [190]	Function words	Neural Networks	English prose
Zheng et al. 2006 [198]	Characters, function words and syntax	Decision Trees, Neural Networks, and SVM	English and Chinese newsgroups
Li et al. 2006 [118]	Lexical, syntactic, structural, and content-specific features	Neural Networks and SVM	English and Chinese newsgroups
Tearle et al. 2008 [184]	Lexical, syntactic, structural, and content-specific features	Neural Networks	English prose and English drama
Jockers et al. 2008 [88]	Words	Nearest Shrunken Centroid	English prose
Jockers and Witten 2010 [87]	Words and word-level bigrams	Nearest Shrunken Centroid and SVM	English prose
Schaalje and Fields 2011 [162]	Word-level statistics	Nearest Shrunken Centroid	English prose
Fung 2003 [63]	Function words	SVM	English prose
Diederich et al. 2003 [47]	Function words	SVM	German journalism
Gamon [64]	Function words, syntactic and semantic features	SVM	English prose
Koppel et al. 2005 [112]	Function words and part-of-speech tags	SVM	English prose
Koppel et al. 2006 [110]	tf-idf over words and characters	SVM	English web posts
Argamon et al. 2007 [9]	Functional lexical features	SVM	English prose
Pavelec et al. 2007 [148]	Conjunction types	SVM	Portuguese journalism
Koppel et al. 2007 [111]	Function words, syntactic structures, part-of-speech tags, complexity and richness measures, and syntactic and idiosyncratic usage	SVM	English prose
Stamatatos 2008 [174]	Character-level n-grams	SVM	English and Arabic journalism
Forstall and Scheirer 2009 [57]	Character-level n-grams	SVM	English prose, and English and Latin poetry
Escalante et al. 2011 [53]	Character-level n-grams	SVM	English journalism
de Vel et al. 2001 [46]	Capitalization, white space, and punctuation	SVM	English e-mail
Hedegaard et al. 2011 [73]	Word- and character-level n-grams and semantic features	SVM	English and (translated) Russian prose
Savoy 2013 [161]	Function words	SVM	English prose
Sidorov et al. 2014 [168]	Syntactic n-grams	SVM	English prose
Sapkota et al. 2015 [159]	Character-level n-grams	SVM	English electronic communication and journalism
Abbasi and Chen 2005 [1]	Lexical, syntactic, and structural features	Decision Trees and SVM	Arabic and English web posts
Argamon et al. [8]	Function words and part-of-speech tags	Decision Trees	English journalism
Koppel and Schler 2003 [106]	Function words, part-of-speech tags, idiosyncratic usage	Decision Trees and SVM	English e-mail
Popescu and Grozea 2012 [151]	Character-level n-grams	Random Forest	AAAC data set [89]
Bartoli et al. 2015 [15]	Lexical, syntactic, structural, and content-specific features	Random Forest	English, Dutch, Greek and Spanish prose
Maitra et al. 2015 [123]	Lexical, syntactic, structural, and content-specific features	Random Forest	English, Dutch, Greek and Spanish prose
Pacheco et al. 2015 [147]	Lexical, semantic, syntactic, structural, and content-specific features	Random Forest	English, Dutch, Greek and Spanish prose
Caliskan-Islam 2015 [33]	Lexical and syntactic features	Random Forest	Source code

were not present in the training set. Traditional classifiers are not designed for this scenario, but a new class of open set classifiers is able to produce results that indicate positive classification or rejection in the case of an unknown author.

In this review, we introduced a basic methodology for open set attribution incorporating an algorithm that is known to minimize the risk of the unknown [163], and presented an experiment that highlighted the difficulty of the problem. It is worth mentioning that a number of other existing open set machine learning algorithms could be considered [44], [86], [164]. Certainly this is another avenue of possible theoretical and practical work.

F. Decision-Level Fusion

Finally, due to the complexity of the problem, combining different models is a promising path. Here we investigated combining different character- and word-level n-gram feature sets along with POS-tag n-grams, which turned out

TABLE IX
WORKS IN SHORT TEXT AUTHORSHIP ATTRIBUTION (SORTED BY TYPE OF CORPUS)

Source	Features Used	Classifier	Corpus
Sanderson and Guenter 2006 [158]	Character and word sequences	Character-level Sequence Kernel, Markov chains and SVM	English short text samples
Hirst and Feiguina 2007 [74]	Syntactic labels	SVM	English short text samples
Koppel et al. 2007 [111]	Function words	SVM	English essays
Forstall et al. 2011 [56]	Character-level n-grams	SVM	Latin poetry
Anderson 2001 [6]	Capitalization, white space, and punctuation	SVM	English e-mail
de Vel et al. 2001 [46]	Capitalization, white space, and punctuation	SVM	English e-mail
Koppel and Schler 2003 [106]	Function words, part-of-speech tags, idiosyncratic usage	Decision Trees and SVM	English e-mail
Layton et al. 2012 [115]	Character-level n-grams	SCAP	English electronic communication
Brocardo et al. 2013 [27]	Character-level n-grams	*Ad hoc* similarity measure	English e-mail
Koppel et al. 2011 [109]	Character-level n-grams	Cosine similarity	English web posts
Koppel and Winter 2014 [113]	Character- and word-level n-grams	SVM	English web posts
Qian et al. 2014 [152]	Word and character-level n-grams and syntactic features	SVM	English web posts
Afroz et al. 2015 [3]	Lexical, syntactic and domain-specific features	SVM	Russian, English and German web posts
Frantzeskou et al. 2006 [60]	Byte-level n-grams	SCAP	Source code
Frantzeskou et al. 2007 [59]	Byte-level n-grams	SCAP	Source code
Hayes 2008 [72]	Lexical features	Multivariate analysis and Linear Discrim. Analysis	Source code
Burrows and Tahaghoghi 2007 [32]	Token-level n-grams	Statistical hypothesis test	Source code
Caliskan-Islam et al. 2015 [34], [35]	Lexical and syntactic features	Random Forest	Source and compiled code

TABLE X
MODELS FOR EASTERN AND NEAR EASTERN LANGUAGES, WORKS IN AUTHORSHIP ATTRIBUTION FOR SOCIAL MEDIA, AND SEMANTIC ANALYSIS FOR SHORT TEXTS

Source	Features Used	Classifier	Corpus
Peng et al. 2003 [149]	Character-level n-grams	Naïve-Bayes	Chinese prose
Kešelj et al. 2003 [97]	Byte-level n-grams	Dissimilarity measure	Chinese prose
Abbasi and Chen 2005 [1]	Lexical, syntactic, and structural features	Decision Trees and SVM	Arabic and English web posts
Abbasi and Chen 2008 [2]	Static and dynamic features	PCA and Karhunen-Loeve transforms	English email and web posts
Layton et al. 2010 [116]	Character-level n-grams	SCAP	English social media
Boutwell 2011 [23]	Character-level n-grams	Naïve-Bayes	English social media
Silva et al. 2011 [169]	Idiosyncratic usage	SVM	Portuguese social media
Green and Sheppard 2013 [67]	Lexical and syntactic features	SVM	English social media
Keretna et al. 2013 [96]	Part of Speech Features	Jacard's coefficient index	English social media
Mikros et al. 2013 [139]	Author's multilevel n-gram profile	SVM	Greek social media
Schwartz et al. 2013 [166]	Character- and word-level n-grams	SVM	English social media
Bhargava et al. 2013 [18]	Lexical features	SVM	English social media
Almishari et al. 2014 [5]	Character-level n-grams	Naïve-Bayes	English social media
Okuno et al. 2014 [144]	Part of Speech n-grams	cosine similarity	English social media
Arakawa et al. 2014 [7]	Retweet features	Variable Importance	Japanese social media
Igawa et al. 2015 [85]	Word-level n-grams	*Ad hoc* similarity measure	English social media
Albadarneh et al. 2015 [4]	TF-IDF fearures	Naïve-Bayes	Arabic social media
Azarbonyad et al. 2015 [11]	Character-level n-grams	SCAP	English social media
Titov and McDonald 2008 [185]	Latent Dirichlet Allocation and Probabilistic Latent Semantic Analysis	Topic Assignment	Web reviews
Li et al. 2010 [119]	Latent Dirichlet Allocation	Topic Assignment	Web post summaries
Hong and Davison 2010 [77]	Latent Dirichlet Allocation and Author-Topic Model	Topic Assignment	English social media
Zhao et al. 2011 [194]	Latent Dirichlet Allocation	Topic Assignment	English social media
Seroussi et al. 2012 [167]	Words and Latent Dirichlet Allocation	Probabilistic Model	English e-mail and web posts
Owoputi et al. 2013 [146]	Words	Brown Clustering	English social media

to be effective. However, according to our analysis, not all features are equally important, which shows that investigating how each group of features affects the performance is an important research direction. Future work could explore combining different problem-solving paradigms at the decision-level. For instance, combining the output of the PMSVM,

Random Forest, SCAP and PPM models to ideally arrive at a more accurate result, compared to examining each model in isolation.

APPENDIX

G. Function Words Used for Creating the Dataset

See Table IV.

H. Part-of-Speech Tags

See Tables V–X.

REFERENCES

[1] A. Abbasi and H. Chen, "Applying authorship analysis to extremist-group Web forum messages," *IEEE Intell. Syst.*, vol. 20, no. 5, pp. 67–75, Sep. 2005.

[2] A. Abbasi and H. Chen, "Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace," *ACM Trans. Inf. Syst.*, vol. 26, no. 2, Mar. 2008, Art. no. 7.

[3] S. Afroz, A. C. Islam, A. Stolerman, R. Greenstadt, and D. McCoy, "Doppelgänger finder: Taking stylometry to the underground," in *Proc. IEEE Secur. Privacy*, May 2014, pp. 212–226.

[4] J. Albadarneh *et al.*, "Using big data analytics for authorship authentication of arabic tweets," in *Proc. IEEE/ACM Int. Conf. Utility Cloud Comput.*, Dec. 2015, pp. 448–452.

[5] M. Almishari, M. A. Kaafar, E. Oguz, and G. Tsudik, "Stylometric linkability of tweets," in *Proc. Workshop Privacy Electron. Soc.*, 2014, pp. 1–4.

[6] A. M. Anderson, M. W. Corney, O. de Vel, and G. M. Mohay, "Multi-topic e-mail authorship attribution forensics," in *Proc. ACM Conf. Comput. Secur.*, 2001, pp. 1–8.

[7] Y. Arakawa, A. Kameda, A. Aizawa, and T. Suzuki, "Adding Twitter-specific features to stylistic features for classifying tweets by user type and number of retweets," *J. Assoc. Inf. Sci. Technol.*, vol. 65, no. 7, pp. 1416–1423, Jul. 2014.

[8] S. Argamon-Engelson, M. Koppel, and G. Avneri, "Style-based text categorization: What newspaper am I reading?" in *Proc. AAAI Workshop Text Categorization*, 1998, pp. 1–4.

[9] S. Argamon, C. Whitelaw, P. Chase, S. R. Hota, N. Garg, and S. Levitan, "Stylistic text classification using functional lexical features," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 58, no. 6, pp. 802–822, Apr. 2007.

[10] P. Juola, "Computational analysis of authorship and identity for immigration," Juola Assoc., Munhall, PA, USA, White Paper, 2016. [Online]. Available: http://juolaassociates.com/publication/computational-analysis-of-authorship-and-identity-for-immigration/

[11] H. Azarbonyad, M. Dehghani, M. Marx, and J. Kamps, "Time-aware authorship attribution for short text streams," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2015, pp. 727–730.

[12] H. Baayen, H. van Halteren, A. Neijt, and F. Tweedie, "An experiment in authorship attribution," in *Proc. 6th Journées Int. d'Analyse Statist. Données Textuelles*, Mar. 2002, pp. 29–37.

[13] H. Baayen, H. van Halteren, and F. Tweedie, "Outside the cave of shadows: Using syntactic annotation to enhance authorship attribution," *Literary Linguistic Comput.*, vol. 11, no. 3, pp. 121–132, 1996.

[14] D. Bagnall, "Author identification using multiheaded recurrent neural networks," in arXiv https://arxiv.org/pdf/1506.04891, pp. 1—11, 2015.

[15] A. Bartoli, A. Dagri, A. De Lorenzo, E. Medvet, and F. Tarlao, "An author verification approach based on differential features," in *Proc. CLEF Eval. Labs*, 2015, pp. 1–7.

[16] D. Benedetto, E. Caglioti, and V. Loreto, "Language trees and zipping," *Phys. Rev. Lett.*, vol. 88, no. 4, p. 048702, Jan. 2002.

[17] Y. Bengio, *Learning Deep Architectures for AI*. Breda, The Netherlands: Now Publishers, 2009.

[18] M. Bhargava, P. Mehndiratta, and K. Asawa, "Stylometric analysis for authorship attribution on Twitter," in *Proc. 2nd Int. Conf. Big Data Anal.*, 2013, pp. 37–47.

[19] J. N. G. Binongo, "Who wrote the 15th book of Oz? An application of multivariate analysis to authorship attribution," *Chance*, vol. 16, no. 2, pp. 9–17, 2003.

[20] J. N. G. Binongo and M. W. A. Smith, "The application of principal component analysis to stylometry," *Literary Linguistic Comput.*, vol. 14, no. 4, pp. 445–466, 1999.

[21] C. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

[22] A. F. Bissell, "Weighted cumulative sums for text analysis using word counts," *J. Roy. Statist. Soc.*, vol. 158, no. 3, pp. 525–545, 1995.

[23] S. R. Boutwell, "Authorship attribution of short messages using multimodal features," M.S. thesis, Naval Postgraduate School, Monterey, CA, USA, 2011.

[24] J. Boyd-Graber, D. Mimno, and D. Newman, "Care and feeding of topic models: Problems, diagnostics, and improvements," in *Handbook of Mixed Membership Models and Their Applications* (CRC Handbooks of Modern Statistical Methods). E. M. Airoldi, D. Blei, E. A. Erosheva, and S. E. Fienberg, Eds. Boca Raton, FL, USA: CRC Press, 2014.

[25] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.

[26] M. Brennan, S. Afroz, and R. Greenstadt, "Adversarial stylometry: Circumventing authorship recognition to preserve privacy and anonymity," *ACM Trans. Inf. Syst. Secur.*, vol. 15, no. 3, Nov. 2012, Art. no. 12.

[27] M. L. Brocardo, I. Traore, S. Saad, and I. Woungang, "Authorship verification for short messages using stylometry," in *Proc. Int. Conf. Comput., Inf. Telecommun. Syst.*, May 2013, pp. 1–6.

[28] J. F. Burrows, "Word-patterns and story-shapes: The statistical analysis of narrative style," *Literary Linguistic Comput.*, vol. 2, no. 2, pp. 61–70, 1987.

[29] J. F. Burrows, "'An ocean where each kind. . .': Statistical analysis and some major determinants of literary style," *Comput. Humanities*, vol. 23, no. 4, pp. 309–321, Aug. 1989.

[30] J. F. Burrows, "Computers and the study of literature," *Comput. Written Texts*, vol. 1, no. 1, pp. 167–204, 1992.

[31] J. F. Burrows, "Not unless you ask nicely: The interpretative nexus between analysis and information," *Literary Linguistic Comput.*, vol. 7, no. 2, pp. 91–109, 1992.

[32] S. Burrows and S. M. M. Tahaghoghi, "Source code authorship attribution using *n*-grams," in *Proc. Austral. Document Comput. Symp.*, Melbourne, VIC, Australia, 2007, pp. 32–39.

[33] A. Caliskan-Islam, "Stylometric fingerprints and privacy behavior in textual data," Ph.D. dissertation, Drexel Univ., Philadelphia, PA, USA, 2015.

[34] A. Caliskan-Islam *et al.*, "De-anonymizing programmers via code stylometry," in *Proc. 24th USENIX Secur. Symp.*, 2015, pp. 255–270.

[35] A. Caliskan-Islam *et al.*, "When coding style survives compilation: De-anonymizing programmers from executable binaries," *CoRR* abs/1512.08546, 2015.

[36] D. Cerra and M. Datcu, "Algorithmic relative complexity," *Entropy*, vol. 13, no. 4, pp. 902–914, 2011.

[37] D. Cerra, M. Datcu, and P. Reinartz, "Authorship analysis based on data compression," *Pattern Recognit. Lett.*, vol. 42, pp. 79–84, Jun. 2014.

[38] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 2:27:1–27:27, Apr. 2011. [Online]. Available: http://tinyurl.com/d6f624

[39] C. E. Chaski, "Empirical evaluations of language-based author identification techniques," *Forensic Linguistics*, vol. 8, no. 1, pp. 1–65, 2001.

[40] C. E. Chaski, "Who's at the keyboard? authorship attribution in digital evidence investigations," *Int. J. Digit. Evidence*, vol. 4, no. 1, pp. 1–13, 2005.

[41] A. Chen. (Jun. 2015). *The Agency—The New York Times Magazine*, accessed on Jul. 1, 2015. [Online]. Available: http://www.nytimes.com/2015/06/07/magazine/the-agency.html

[42] R. Clement and D. Sharp, "Ngram and Bayesian classification of documents for topic and authorship," *Literary Linguistic Comput.*, vol. 18, no. 4, pp. 423–447, 2003.

[43] S. Coles, *An Introduction to Statistical Modeling of Extreme Values*. London, U.K.: Springer, 2001.

[44] F. de O. Costa, E. Silva, M. Eckmann, W. J. Scheirer, and A. Rocha, "Open set source camera attribution and device linking," *Pattern Recognit. Lett.*, vol. 39, pp. 91–101, Apr. 2014.

[45] M. Coulthard, "On admissible linguistic evidence," *J. Law Policy*, vol. 21, no. 2, p. 441, 2012.

[46] O. de Vel, A. Anderson, M. Corney, and G. Mohay, "Mining e-mail content for author identification forensics," *ACM SIGMOD Rec.*, vol. 30, no. 4, pp. 55–64, Dec. 2001.

[47] J. Diederich, J. Kindermann, E. Leopold, and G. Paass, "Authorship attribution with support vector machines," *Appl. Intell.*, vol. 19, no. 1, pp. 109–123, Jul. 2003.

[48] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," in *Proc. 13th Conf. USENIX Secur. Symp.*, 2004, p. 21.

[49] M. Eder, "Does size matter? Authorship attribution, small samples, big problem," *Digit. Scholarship Humanities*, vol. 30, no. 2, pp. 162–167, Jun. 2015.

[50] M. Eder, M. Kestemont, and J. Rybicki, "Stylometry with R: A suite of tools," in *Proc. Digit. Humanities*, 2013, pp. 1–4.

[51] M. Eder, J. Rybicki, and M. Kestemont. (2015). *Stylo: Functions for a Variety of Stylometric Analyses*. [Online]. Available: https://cran.r-project.org/web/packages/stylo/index.html

[52] A. Ellegård, "A Statistical method for determining authorship," *Junius Lett.*, vol. 40, no. 1, pp. 85–90, Jan./Mar. 1964.

[53] H. J. Escalante, T. Solorio, and M. Montes-y-Gómez, "Local histograms of character *N*-grams for authorship attribution," in *Proc. Annu. Meeting Assoc. Comput. Linguistics, Human Lang. Technol.*, 2011, pp. 288–298.

[54] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, Jun. 2008.

[55] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems?" *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3133–3181, Jan. 2014.

[56] C. W. Forstall, S. L. Jacobson, and W. J. Scheirer, "Evidence of intertextuality: Investigating Paul the Deacon's *Angustae Vitae*," *Literary Linguistic Comput.*, vol. 26, pp. 285–296, Sep. 2011.

[57] C. Forstall and W. Scheirer, "Features from frequency: Authorship and stylistic analysis using repetitive sound," *Proc. Chicago Colloq. Digit. Humanities Comput. Sci.*, 2009, pp. 7–9.

[58] G. Frantzeskou, S. G. MacDonell, and E. G. Stamatatos, "Source code authorship analysis for supporting the cybercrime investigation process," in *Handbook of Research on Computational Forensics, Digital Crime, and Investigation: Methods and Solutions*. Hershey, PA, USA: Information Science Reference, 2006, pp. 470–495.

[59] G. Frantzeskou, E. Stamatatos, S. Gritzalis, C. E. Chaski, and B. S. Howald, "Identifying authorship by byte-level n-grams: The source code author profile (SCAP) method," *Int. J. Digit. Evidence*, vol. 6, no. 1, pp. 1–18, 2007.

[60] G. Frantzeskou, E. Stamatatos, S. Gritzalis, and S. Katsikas, "Effective identification of source code authors using byte-level information," in *Proc. Int. Conf. Softw. Eng.*, 2006, pp. 893–896.

[61] L. Fridman *et al.*, "Multi-modal decision fusion for continuous authentication," *Comput. Elect. Eng.*, vol. 41, pp. 142–156, Jan. 2015.

[62] L. Fridman, S. Weber, R. Greenstadt, and M. Kam, "Active authentication on mobile devices via stylometry, application usage, Web browsing, and GPS location," *IEEE Syst. J.*, to be published.

[63] G. Fung, "The disputed federalist papers: SVM feature selection via concave minimization," in *Proc. Conf. Diversity Comput.*, 2003, pp. 42–46.

[64] M. Gamon, "Linguistic correlates of style: Authorship classification with deep linguistic analysis features," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2004, p. 611.

[65] K. Gimpel *et al.*, "Part-of-speech tagging for Twitter: Annotation, features, and experiments," in *Proc. Annu. Meeting Assoc. Comput. Linguistics, Human Lang. Technol.*, vol. 2. 2011, pp. 42–47.

[66] T. Grant, "TXT4N6: Method, consistency, and distinctiveness in the analysis of SMS text messages," *J. Law Policy*, vol. 21, p. 467, Sep. 2012.

[67] R. M. Green and J. W. Sheppard, "Comparing frequency-and style-based features for Twitter author identification," in *Proc. FLAIRS Conf.*, 2013, pp. 1–6.

[68] J. Grieve, "Quantitative authorship attribution: An evaluation of techniques," *Literary Linguistic Comput.*, vol. 22, no. 3, pp. 251–270, 2007.

[69] The Grugq. (Dec. 2013). *In Search of OPSEC Magic Sauce, Hacker OPSEC Blog*. accessed on Jul. 1, 2015. [Online]. Available: http://grugq.github.io/blog/2013/12/21/in-search-of-opsec-magic-sauce/

[70] Y. Guo, T. Hastie, and R. Tibshirani, "Regularized linear discriminant analysis and its application in microarrays," *Biostatistics*, vol. 8, no. 1, pp. 86–100, Jan. 2007.

[71] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York, NY, USA: Springer, 2009.

[72] J. F. Hayes, "Authorship attribution: A principal component and linear discriminant analysis of the consistent programmer hypothesis," *Int. J. Comput. Appl.*, vol. 15, no. 2, pp. 79–99, 2008.

[73] S. Hedegaard and J. G. Simonsen, "Lost in translation: Authorship attribution using frame semantics," in *Proc. Annu. Meeting Assoc. Comput. Linguistics, Human Lang. Technol.*, vol. 2. 2011, pp. 65–70.

[74] G. Hirst and O. Feiguina, "Bigrams of syntactic labels for authorship discrimination of short texts," *Literary Linguistic Comput.*, vol. 22, no. 4, pp. 405–417, 2007.

[75] D. I. Holmes and R. S. Forsyth, "The *Federalist* revisited: New directions in authorship attribution," *Literary Linguistic Comput.*, vol. 10, no. 2, pp. 111–127, 1995.

[76] D. I. Holmes, M. Robertson, and R. Paez, "Stephen crane and the New-York tribune: A case study in traditional and non-traditional authorship attribution," *Comput. Humanities*, vol. 35, no. 3, pp. 315–331, Aug. 2001.

[77] K. Hong and B. D. Davison, "Empirical study of topic modeling in Twitter," in *Proc. Workshop Social Media Anal.*, 2010, pp. 80–88.

[78] J. F. Hoorn, S. L. Frank, W. Kowalczyk, and F. van der Ham, "Neural network identification of poets using letter sequences," *Literary Linguistic Comput.*, vol. 14, no. 3, pp. 311–338, 1999.

[79] D. L. Hoover, "Frequent word sequences and statistical stylistics," *Literary Linguistic Comput.*, vol. 17, no. 2, pp. 157–180, 2002.

[80] D. L. Hoover, "Multivariate analysis and the study of style variation," *Literary Linguistic Comput.*, vol. 18, no. 4, pp. 341–360, 2003.

[81] D. L. Hoover, "Delta prime?" *Literary Linguistic Comput.*, vol. 19, no. 4, pp. 477–495, 2004.

[82] D. L. Hoover, "Testing Burrows's delta," *Literary Linguistic Comput.*, vol. 19, no. 4, pp. 453–475, 2004.

[83] K. Hornik and B. Grün, "topicmodels: An R package for fitting topic models," *J. Statist. Softw.*, vol. 40, no. 13, pp. 1–30, 2011.

[84] C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S. Keerthi, and S. Sundararajan, "A dual coordinate descent method for large-scale linear SVM," in *Proc. Int. Conf. Mach. Learn.*, 2008, pp. 408–415.

[85] R. A. Igawa, A. M. G. de Almeida, B. B. Zarpelão, and S. Barbon, Jr, "Recognition of compromised accounts on Twitter," in *Proc. Annu. Conf. Brazilian Symp. Inf. Syst., Inf. Syst.*, 2015, pp. 1–6.

[86] L. P. Jain, W. J. Scheirer, and T. E. Boult, "Multi-class open set recognition using probability of inclusion," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 393–409.

[87] M. L. Jockers and D. M. Witten, "A comparative study of machine learning methods for authorship attribution," *Literary Linguistic Comput.*, vol. 25, no. 2, pp. 215–223, 2010.

[88] M. L. Jockers, D. M. Witten, and C. S. Criddle, "Reassessing authorship of the *Book of Mormon* using delta and nearest shrunken centroid classification," *Literary Linguistic Comput.*, vol. 23, no. 4, pp. 465–491, Oct. 2008.

[89] P. Juola, "Authorship attribution," *Found. Trends Inf. Retr.*, vol. 1, no. 3, pp. 233–334, Dec. 2006.

[90] P. Juola and R. H. Baayen, "A controlled-corpus experiment in authorship identification by cross-entropy," *Literary Linguistic Comput.*, vol. 20, pp. 59–67, Jun. 2005.

[91] P. Juola and E. Stamatatos, "Overview of the author identification task at PAN 2013," in *Proc. Notebook Papers CLEF Labs Workshops*, 2013, pp. 1–20.

[92] P. Juola and D. Vescovi, "Analyzing stylometric approaches to author obfuscation," in *Advances in Digital Forensics VII*. Springer, 2011, pp. 115–125.

[93] D. Jurafsky, *Speech and Language Processing*, 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2008.

[94] G. Kacmarcik and M. Gamon, "Obfuscating document stylometry to preserve author anonymity," in *Proc. COLING/ACL Main Conf. Poster Sessions*, 2006, pp. 444–451.

[95] R. Kelly. (2014). *Pyenchant: A spellchecking Library for Python*. [Online]. Available: https://pythonhosted.org/pyenchant/

[96] S. Keretna, A. Hossny, and D. Creighton, "Recognising user identity in Twitter social networks via text mining," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2013, pp. 3079–3082.

[97] V. Kešelj, F. Peng, N. Cercone, and C. Thomas, "N-gram-based author profiles for authorship attribution," in *Proc. Conf. Pacific Assoc. Comput. Linguistics*, 2003, pp. 255–264.

[98] M. Kestemont, "Function words in authorship attribution from black magic to theory?" in *Proc. 14th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2014, pp. 59–66.

[99] M. Kestemont, S. Moens, and J. Deploige, "Collaborative authorship in the twelfth century: A stylometric study of Hildegard of Bingen and Guibert of Gembloux," *Digit. Scholarship Humanities*, vol. 30, pp. 199–224, Jun. 2015.

[100] D. V. Khmelev and W. J. Teahan, "A repetition based measure for verification of text collections and for text categorization," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2003, pp. 104–110.

[101] D. V. Khmelev and F. Tweedie, "Using Markov chains for identification of writer," *Literary Linguistic Comput.*, vol. 16, no. 3, pp. 299–307, 2002.

[102] B. Kjell, "Authorship attribution of text samples using neural networks and Bayesian classifiers," in *Proc. IEEE Intl. Conf. Syst., Man, Humans, Inf. Technol.*, vol. 2. Oct. 1994, pp. 1660–1664.

[103] B. Kjell, "Authorship determination using letter pair frequency features with neural network classifiers," *Literary Linguistic Comput.*, vol. 9, no. 2, pp. 119–124, 1994.

[104] B. Kjell, W. A. Woods, and O. Frieder, "Discrimination of authorship using visualization," *Inf. Process. Manage.*, vol. 30, no. 1, pp. 141–150, Jan./Feb. 1994.

[105] B. Kjell, W. A. Woods, and O. Frieder, "Information retrieval using letter tuples with neural network and nearest neighbor classifiers," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, vol. 2. Oct. 1995, pp. 1222–1226.

[106] M. Koppel and J. Schler, "Exploiting stylistic idiosyncrasies for authorship attribution," in *Proc. Workshop Comput. Approaches Style Anal. Synth.*, vol. 69. 2003, pp. 72–80.

[107] M. Koppel and J. Schler, "Authorship verification as a one-class classification problem," in *Proc. Int. Conf. Mach. Learn.*, 2004, p. 62.

[108] M. Koppel, J. Schler, and S. Argamon, "Computational methods in authorship attribution," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 60, no. 1, pp. 9–26, Jan. 2009.

[109] M. Koppel, J. Schler, and S. Argamon, "Authorship attribution in the wild," *Language Resour. Eval.*, vol. 45, no. 1, pp. 83–94, Mar. 2011.

[110] M. Koppel, J. Schler, S. Argamon, and E. Messeri, "Authorship attribution with thousands of candidate authors," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2006, pp. 659–660.

[111] M. Koppel, J. Schler, and E. Bonchek-Dokow, "Measuring differentiability: Unmasking pseudonymous authors," *J. Mach. Learn. Res.*, vol. 8, pp. 1261–1276, Dec. 2007.

[112] M. Koppel, J. Schler, and K. Zigdon, "Determining an author's native language by mining a text for errors," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2005, pp. 624–628.

[113] M. Koppel and Y. Winter, "Determining if two documents are written by the same author," *J. Assoc. Inf. Sci. Technol.*, vol. 65, no. 1, pp. 178–187, Jan. 2014.

[114] O. V. Kukushkina, A. A. Polikarpov, and D. V. Khmelev, "Using literal and grammatical statistics for authorship attribution," *Problems Inf. Transmiss.*, vol. 37, no. 2, pp. 172–184, Apr. 2001.

[115] R. Layton, S. McCombie, and P. Watters, "Authorship attribution of IRC messages using inverse author frequency," in *Proc. Cybercrime Trustworthy Comput. Workshop*, Oct. 2012, pp. 7–13.

[116] R. Layton, P. Watters, and R. Dazeley, "Authorship attribution for Twitter in 140 characters or less," in *Proc. Cybercrime Trustworthy Comput. Workshop*, Jul. 2010, pp. 1–8.

[117] G. Ledger and T. Merriam, "Shakespeare, Fletcher, and the two noble Kinsmen," *Literary Linguistic Comput.*, vol. 9, no. 3, pp. 235–248, 1994.

[118] J. Li, R. Zheng, and H. Chen, "From fingerprint to writeprint," *Commun. ACM*, vol. 49, no. 4, pp. 76–82, Apr. 2006.

[119] P. Li, J. Jiang, and Y. Wang, "Generating templates of entity summaries with an entity-aspect model and pattern mining," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2010, pp. 640–649.

[120] D. Lowe and R. Matthews, "Shakespeare vs. Fletcher: A stylometric analysis by radial basis functions," *Comput. Humanities*, vol. 29, no. 6, pp. 449–461, Dec. 1995.

[121] K. Luyckx, *Scalability Issues in Authorship Attribution*. Brussels, Belgium: Brussels Univ. Press, 2010.

[122] K. Luyckx and W. Daelemans, "Authorship attribution and verification with many authors and limited data," in *Proc. Int. Conf. Comput. Linguistics*, 2008, pp. 513–520.

[123] P. Maitra, S. Ghosh, and D. Das, "Authorship verification: An approach based on random forest," in *Proc. CLEF Eval. Labs*, 2015, pp. 1–9.

[124] M. B. Malyutov, "Authorship attribution of texts: A review," in *General Theory of Information Transfer and Combinatorics*. Springer, 2006, pp. 362–380.

[125] L. M. Manevitz and M. Yousef, "One-class SVMs for document classification," *J. Mach. Learn. Res.*, vol. 2, pp. 139–154, Mar. 2002.

[126] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*. Cambridge, MA, USA: MIT Press, 1999.

[127] C. Martindale and D. McKenzie, "On the utility of content analysis in author attribution: *The Federalist*," *Comput. Humanities*, vol. 29, no. 4, pp. 259–270, Aug. 1995.

[128] C. Mascol, "Curves of Pauline and pseudo-Pauline style I," *Unitarian Rev.*, vol. 30, pp. 453–460, Nov. 1888.

[129] R. A. J. Matthews and T. V. N. Merriam, "Neural computation in stylometry I: An application to the works of shakespeare and Fletcher," *Literary Linguistic Comput.*, vol. 8, no. 4, pp. 203–209, 1994.

[130] A. K. McCallum. (2002). *Mallet: A Machine Learning for Language Toolkit*. [Online]. Available: http://mallet.cs.umass.edu

[131] A. W. E. McDonald, S. Afroz, A. Caliskan, A. Stolerman, and R. Greenstadt, "Use fewer instances of the letter 'i': Toward writing style anonymization," in *Privacy Enhancing Technologies*. Springer, 2012, pp. 299–318.

[132] D. L. Mealand, "Correspondence analysis of Luke," *Literary Linguistic Comput.*, vol. 10, no. 3, pp. 171–182, 1995.

[133] T. C. Mendenhall, "The characteristic curves of composition," *Science*, vol. NS-9, no. 214S, pp. 237–246, 1887.

[134] T. Merriam, "What Shakespeare wrote in Henry VIII (part I)," *Bard*, vol. 2, pp. 81–94, 1979.

[135] T. Merriam, "What Shakespeare wrote in Henry VIII (part II)," *Bard*, vol. 2, pp. 11–28, 1980.

[136] T. Merriam, "The authorship of Sir Thomas More," *ALLC Bull. Assoc. Library Linguistic Comput.*, vol. 10, no. 1, pp. 1–7, 1982.

[137] T. Merriam, "Marlowe's hand in *Edward III revisited*," *Literary Linguistic Comput.*, vol. 11, no. 1, pp. 19–22, 1996.

[138] T. V. N. Merriam and R. A. J. Matthews, "Neural computation in stylometry II: An application to the works of Shakespeare and Marlowe," *Literary Linguistic Comput.*, vol. 9, no. 1, pp. 1–6, 1994.

[139] G. K. Mikros and P. Kostas, "Authorship attribution in Greek tweets using authors multilevel n-gram profiles," in *Proc. AAAI Spring Symp. Series*, 2013, pp. 17–23.

[140] A. Q. Morton, *Literary Detection: How to Prove Authorship and Fraud in Literature and Documents*. London, U.K.: Bowker, 1978.

[141] F. Mosteller and D. L. Wallace, *Inference and Disputed Authorship: The Federalist*. Reading, MA, USA: Addison-Wesley, 1964.

[142] A. Narayanan *et al.*, "On the feasibility of Internet-scale author identification," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2012, pp. 300–314.

[143] B. O'Connor, M. Krieger, and D. Ahn, "TweetMotif: Exploratory search and topic summarization for Twitter," in *Proc. ICWSM*, 2010, pp. 1–2.

[144] S. Okuno, H. Asai, and H. Yamana, "A challenge of authorship identification for ten-thousand-scale microblog users," in *Proc. IEEE Int. Conf. Big Data*, Oct. 2014, pp. 52–54.

[145] W. Oliveira, Jr., E. Justino, and L. S. Oliveira, "Comparing compression models for authorship attribution," *Forensic Sci. Int.*, vol. 228, nos. 1–3, pp. 100–104, May 2013.

[146] O. Owoputi, B. O'Connor, C. Dyer, K. Gimpel, N. Schneider, and N. A. Smith, "Improved part-of-speech tagging for online conversational text with word clusters," in *Proc. NAACL*, 2013, pp. 1–11.

[147] M. L. Pacheco, K. Fernandes, and A. Porco, "Random forest with increased generalization: A universal background approach for authorship verification," in *Proc. CLEF Eval. Labs*, 2015, pp. 1–8.

[148] D. Pavelec, E. Justino, and L. S. Oliveira, "Author identification using stylometric features," *Inteligencia Artif., Rev. Iberoamericana Inteligencia Artif.*, vol. 11, no. 36, pp. 59–66, 2007.

[149] F. Peng, D. Schuurmans, V. Kešelj, and S. Wang, "Language independent authorship attribution using character level language models," in *Proc. Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2003, pp. 267–274.

[150] F. Peng, D. Schuurmans, and S. Wang, "Augmenting Naïve Bayes classifiers with statistical language models," *Inf. Retr.*, vol. 7, no. 3, pp. 317–345, Sep. 2004.

[151] M. Popescu and C. Grozea, "Kernel methods and string kernels for authorship analysis," in *Proc. CLEF Eval. Labs*, 2012, pp. 1–12.

[152] T. Qian, B. Liu, L. Chen, and Z. Peng, "Tri-training for authorship attribution with limited training data," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2014, pp. 345–351.

[153] F. Y. Rashid and M. Eddy. (Jun. 2013). *Simple Ways to Make Anonymous Phone Calls, Secure Your Data*, PC Magazine, accessed on Jul. 1, 2015. [Online]. Available: http://www.pcmag.com/article2/0,2817,2420131,00.asp

[154] M. G. Reed, P. F. Syverson, and D. M. Goldschlag, "Anonymous connections and onion routing," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 4, pp. 482–494, May 1998.
[155] R. Řehůřek and P. Sojka, "Software framework for topic modelling with large corpora," in *Proc. Workshop New Challenges NLP Frameworks*, Valletta, Malta, 2010, pp. 45–50. [Online]. Available: http://is.muni.cz/publication/884893/en
[156] A. Rocha and S. Klein Goldenstein, "Multiclass from binary: Expanding one-versus-all, one-versus-one and ECOC-based approaches," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 2, pp. 289–302, Feb. 2014.
[157] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. New York, NY, USA: McGraw-Hill, 1986.
[158] C. Sanderson and S. Guenter, "Short text authorship attribution via sequence kernels, Markov chains and author unmasking: An investigation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2006, pp. 482–491.
[159] U. Sapkota, S. Bethard, M. Montes-y-Gómez, and T. Solorio, "Not all character n-grams are created equal: A study in authorship attribution," in *Proc. Annu. Conf. North Amer. Chapter ACL Human Lang. Technol.*, 2015, pp. 93–102.
[160] U. Sapkota, T. Solorio, M. Montes-y-Gómez, S. Bethard, and P. Rosso, "Cross-topic authorship attribution: Will out-of-topic data help?" in *Proc. Int. Conf. Comput. Linguistics, Tech. Papers*, 2014, pp. 1228–1237.
[161] J. Savoy, "The *Federalist Papers* revisited: A collaborative attribution scheme," *Amer. Soc. Inf. Sci. Technol.*, vol. 50, no. 1, pp. 1–8, 2013.
[162] G. B. Schaalje and P. J. Fields, "Extended nearest shrunken centroid classification: A new method for open-set authorship attribution of texts of varying sizes," *Literary Linguistic Comput.*, vol. 26, pp. 71–88, Jan. 2011.
[163] W. J. Scheirer, L. P. Jain, and T. E. Boult, "Probability models for open set recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2317–2324, Nov. 2014.
[164] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult, "Toward open set recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 7, pp. 1757–1772, Jul. 2013.
[165] B. Schölkopf, J.C. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, 2001.
[166] R. Schwartz, O. Tsur, A. Rappoport, and M. Koppel, "Authorship attribution of micro-messages," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2013, pp. 1880–1891.
[167] Y. Seroussi, F. Bohnert, and I. Zukerman, "Authorship attribution with author-aware topic models," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2012, pp. 264–269.
[168] G. Sidorov, F. Velasquez, E. Stamatatos, A. Gelbukh, and L. Chanona-Hernández, "Syntactic N-grams as machine learning features for natural language processing," *Expert Syst. Appl.*, vol. 41, no. 3, pp. 853–860, Feb. 2014.
[169] R. S. Silva, G. Laboreiro, L. Sarmento, T. Grant, E. Oliveira, and B. Maia, "'Twazn me!!! Automatic authorship analysis of micro-blogging messages," in *Natural Language Processing and Information Systems*. Springer, 2011, pp. 161–168.
[170] H. Somers, "An attempt to use weighted cusums to identify sublanguages," in *Proc. Joint Conf. New Methods Lang. Process. Comput.*, 1998, pp. 131–139.
[171] H. Somers and F. Tweedie, "Authorship attribution and pastiche," *Comput. Humanities*, vol. 37, no. 4, pp. 407–429, Nov. 2003.
[172] E. H. Spafford and S. A. Weeber, "Software forensics: Can we track code to its authors?" *Comput. Secur.*, vol. 12, no. 6, pp. 585–595, Oct. 1993.
[173] Spirit. (2014). *Guess Language: Guess the Natural Language of a Text*, accessed on Jul. 1, 2015. [Online]. Available: https://bitbucket.org/spirit/guess_language
[174] E. Stamatatos, "Author identification: Using text sampling to handle the class imbalance problem," *Inf. Process. Manage.*, vol. 44, no. 2, pp. 790–799, Mar. 2008.
[175] E. Stamatatos, "A survey of modern authorship attribution methods," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 60, no. 3, pp. 538–556, Mar. 2009.
[176] E. Stamatatos, "On the robustness of authorship attribution based on character n-gram features," *J. Law Policy*, vol. 21, no. 2, pp. 421–725, 2013.
[177] E. Stamatatos *et al.*, "Overview of the author identification task at PAN 2015," in *Proc. Conf. Labs Eval. Forum (CLEF)*, 2015, pp. 1–17.
[178] E. Stamatatos *et al.*, "Overview of the author identification task at PAN 2014," in *Proc. Conf. CLEF*, 2014, pp. 877–897.
[179] E. Stamatatos, N. Fakotakis, and G. Kokkinakis, "Automatic text categorization in terms of genre and author," *Comput. Linguistics*, vol. 26, no. 4, pp. 471–495, Dec. 2000.
[180] E. Stamatatos, N. Fakotakis, and G. Kokkinakis, "Computer-based authorship attribution without lexical measures," *Comput. Humanities*, vol. 35, no. 2, pp. 193–214, May 2001.
[181] A. Stolerman, R. Overdorf, S. Afroz, and R. Greenstadt, "Classify, but verify: Breaking the closed-world assumption in stylometric authorship attribution," in *Proc. Annu. IFIP WG Int. Conf. Digit. Forensics*, 2014, pp. 1–17.
[182] A. Subramanya, S. Petrov, and F. Pereira, "Efficient graph-based semi-supervised learning of structured tagging models," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2010, pp. 167–176.
[183] W. J. Teahan and D. J. Harper, "Using compression-based language models for text categorization," in *Language Modeling for Information Retrieval* (The Springer International Series on Information Retrieval), vol. 13, W. B. Croft J. Lafferty, Eds. Dordrecht, The Netherlands: Springer, 2003, pp. 141–165.
[184] M. Tearle, K. Taylor, and H. Demuth, "An algorithm for automated authorship attribution using neural networks," *Literary Linguistic Comput.*, vol. 23, pp. 425–442, Oct. 2008.
[185] I. Titov and R. McDonald, "Modeling online reviews with multi-grain topic models," in *Proc. Int. Conf. World Wide Web*, 2008, pp. 111–120.
[186] F. J. Tweedie, D. I. Holmes, and T. N. Corns, "The provenance of De Doctrina Christiana, attributed to John Milton: A statistical investigation," *Literary Linguistic Comput.*, vol. 13, no. 2, pp. 77–87, 1998.
[187] F. J. Tweedie, S. Singh, and D. I. Holmes, "Neural network applications in stylometry: The *Federalist Papers*," *Comput. Humanities*, vol. 30, no. 1, pp. 1–10, 1996.
[188] H. Van Halteren, "Linguistic profiling for author recognition and verification," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2004, p. 199.
[189] S. V. N. Vishwanathan and A. J. Smola, "Fast kernels for string and tree matching," in *Proc. Neural Inf. Process. Syst.*, 2003, pp. 1–8.
[190] S. Waugh, A. Adams, and F. Tweedie, "Computational stylistics using artificial neural networks," *Literary Linguistic Comput.*, vol. 15, no. 2, pp. 187–198, 2000.
[191] J. Wu, "A fast dual method for HIK SVM learning," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 552–565.
[192] J. Wu, "Power mean SVM for large scale visual classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2344–2351.
[193] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, 2014,Ç pp. 818–833.
[194] W. X. Zhao *et al.*, "Comparing Twitter and traditional media using topic models," in *Proc. Eur. Conf. Inf. Retr.*, 2011, pp. 338–349.
[195] Y. Zhao and J. Zobel, "Effective and scalable authorship attribution using function words," in *Proc. Asian Inf. Retr. Symp.*, 2005, pp. 174–189.
[196] Y. Zhao and J. Zobel, "Searching with style: Authorship attribution in classic literature," in *Proc. Austral. Conf. Comput. Sci.*, vol. 62. 2007, pp. 59–68.
[197] Y. Zhao, J. Zobel, and P. Vines, "Using relative entropy for authorship attribution," in *Proc. Asia Inf. Retr. Symp.*, 2006, pp. 92–105.
[198] R. Zheng, J. Li, H. Chen, and Z. Huang, "A framework for authorship identification of online messages: Writing-style features and classification techniques," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 57, no. 3, pp. 378–393, 2006.
[199] X. S. Zhou and T. S. Huang, "Relevance feedback in image retrieval: A comprehensive review," *Multimedia Syst.*, vol. 8, no. 6, pp. 536–544, Apr. 2003.
[200] G. K. Zipf, *Selected Studies of the Principle of Relative Frequency in Language*. Cambridge, MA, USA: Harvard Univ. Press, 1932.

Authors' photographs and biographies not available at the time of publication.