Use Fewer Instances of the Letter "i": Toward Writing Style Anonymization

Andrew W.E. McDonald, Sadia Afroz, Aylin Caliskan, Ariel Stolerman, and Rachel Greenstadt

Drexel University, Philadelphia, PA
{awm32,sa499,ac993,ams573,greenie}@cs.drexel.edu

Abstract. This paper presents Anonymouth, a novel framework for anonymizing writing style. Without accounting for style, anonymous authors risk identification. This framework is necessary to provide a tool for testing the consistency of anonymized writing style and a mechanism for adaptive attacks against stylometry techniques. Our framework defines the steps necessary to anonymize documents and implements them. A key contribution of this work is this framework, including novel methods for identifying which features of documents need to change and how they must be changed to accomplish document anonymization. In our experiment, 80% of the user study participants were able to anonymize their documents in terms of a fixed corpus and limited feature set used. However, modifying pre-written documents were found to be difficult and the anonymization did not hold up to more extensive feature sets. It is important to note that Anonymouth is only the first step toward a tool to acheive stylometric anonymity with respect to state-of-the-art authorship attribution techniques. The topic needs further exploration in order to accomplish significant anonymity.

Keywords: stylometry, privacy, anonymity, machine learning

1 Introduction

The Privacy Enhancing Technologies community has long been interested in tools that enable people to participate in anonymous or pseudonymous speech[1]. Current anonymity and circumvention systems focus strongly on location-based privacy but do not address many avenues for the leakage of identification through the content of data. In particular, writing style as a marker of identity is not addressed in current circumvention tools. Given the high accuracy of even basic stylometry systems this is not a topic that can afford to be overlooked.

Stylometry is a form of authorship recognition that relies on the linguistic information found in a document. While stylometry existed before computers and artificial intelligence, the field is currently dominated by AI techniques such as neural networks and statistical pattern recognition. State-of-the-art stylometry

[1] Selected Papers in Anonymity: http://freehaven.net/anonbib/

approaches can identify individuals in sets of 50 authors with over 90% accuracy [1]. Recent work has scaled stylometry methods to over 100,000 authors [2]. Stylometry is currently used in intelligence analsyis and forensics. The 2009 Technology Assessment for the State of the Art Biometrics Excellence Roadmap (SABER) commissioned by the FBI stated that, "As non-handwritten communications become more prevalent, such as blogging, text messaging and emails, there is a growing need to identify writers not by their written script, but by analysis of the typed content [3]."

The stylometry field has focused on creating new methods that attempt to classify unknown works using known sets of authors, with little attention being given to the question of what happens when an adversary tries to intentionally circumvent the classification system that has been established. This paper aims to provide a framework for researching and accomplishing writing style anonymization: Anonymouth. Work by Brennan and Greenstadt has shown that non-expert human subjects can defeat stylometry simply by consciously hiding their writing style or imitating the style of another author [4]. However, when analyzing the Brennan-Greenstadt Adversarial Stylometry Corpus, we find that some authors are more capable of composing anonymous documents than others. Furthermore, a case study of the long-term pseudonymous blog, "A Gay Girl in Damascus," showed that even when authors were skilled at hiding their style, doing so with consistency was difficult [5]. There is currently active research in finding stylometry methods that work on adversarial passages such as those in the Brennan-Greenstadt corpus. Even if these methods succeed at identifying these adversarial passages, they should be benchmarked against an adaptive attack where the adversary has access to the features and tools used to identify the text. Lastly, the limited research in circumventing stylometry has focused on creating anonymous documents, whereas the Anonymouth framework provides a mechanism to study the modification/anonymization of documents that were written without anonymity in mind.

This paper includes three key contributions.

1. Our Anonymouth framework defines the steps necessary to anonymize documents. Anonymouth's novel feature clustering and prioritization algorithms enable it to identify the changes necessary to anonymize a document relative to a set of author documents and a set of linguistic features. We show that modifying the features as suggested does result in anonymized documents.
2. We have implemented this framework via two tools, JStylo and Anonymouth, that have been released under an open source license (GPL 3) and can serve as a research platform for stylometry and adversarial stylometry[2]. This software not only performs authorship attribution, but also calculates the features that are most identifying and the ways the feature vectors must change to provide anonymity. The software also provides suggestions to users to help them anonymize their style. We found that 80% of user study participants were able to anonymize their documents in terms of a data corpus and feature set that is known to and chosen by the user before anonymization.

[2] Available at https://psal.cs.drexel.edu/index.php/JStylo-Anonymouth

3. We have performed a user study to investigate whether and how users can edit previously written documents so they obscure their authorship. We show that this problem is harder than starting from scratch. Anonymouth can suggest the right changes, but they are difficult to implement. The methods used to aid a user in anonymizing his document need further development in order for Anonymouth to be effective against state-of-the-art feature sets.

2 Related Work

Anonymization plays an important role in data privacy. Perfect anonymity is hard to achieve. Private information about an individual can be revealed not only from his name and physical and virtual addresses, but also from browser configuration [6], netflix movie ratings [7], and even from the public outputs of a recommender system [8]. Anonymity at the network level can be achieved through onion routing systems like Tor [9]. But the privacy concerns of writing style are still not well-analyzed. Writing style is a serious threat to anonymity and free speech. With the improvement of authorship recognition techniques, it is possible to identify authorship of a document even among 100,000 authors [2].

Authorship attribution can be circumvented by changing writing style. All authorship attribution techniques are based on the fact that people always write in their regular style. Brennan et al. showed that current authorship attribution techniques perform less than random chance if people hide their writing style by imitating someone else or by obfuscating their regular style [4]. Though it is possible to change writing style, it is hard to maintain a separate style consistently in anonymous writings [5].

Rao and Rohatgi suggested round trip machine translation (for example, English → German → English) as a possible method for document anonymization [10]. But because of improvement in machine translation, empirical results have shown that round trip machine translation is not effective in obfuscating writing style[3].

Anonymization by obfuscating writing style was first explored by Kacmarcik et al. [11]. Their approach was to identify the features that a typical authorship attribution method uses to attribute authorship and then adjust the frequencies of these features to make them less effective. They used the Federalist papers and found that 14 changes per 1000 words are sufficient to reduce the likelihood of identifying an author as himself. Our work differs from this work in several ways. First, they did not change the actual documents, only modified the feature sets to prove obfuscation is possible to circumvent attribution. Anonymouth helps the user to change the actual document. Second, their feature set was limited, only function words were used. Anonymouth supports both the Basic-9 [4] feature set and the Writeprints [1] features. Third, only the 12 disputed Federalist papers were analyzed, whereas Anonymouth allows obfuscation of any written document.

[3] http://events.ccc.de/congress/2009/Fahrplan/events/3468.en.html

There is considerable prior work in authorship attribution [1, 12, 13]. But currently no tool is available to allow circumvention of the authorship attribution techniques to achieve anonymity. Anonymouth is the first research to explore the idea of changing writing style to anonymize a written document. Anonymouth makes a user aware of the idiosyncrasies of his writing style. It allows users to choose a background corpus in terms of which a document can be anonymized. It indicates features that are unique to the user and suggests how a feature value can be adjusted to achieve a sufficient level of anonymity.

3 Problem Statement

An author A has a document D that he wants to anonymize. The author selects a set of his own writing D_{pre} and a set B of N authors where $A \notin B$. Author A also chooses a feature set F and authorship attribution method M. The goal is to create a new document $D\prime$ from D where the feature values F are changed sufficiently so that $D\prime$ does not appear to be written by A. To evaluate the level of anonymity, D_{pre} is used. $D\prime$ is anonymized if a classifier trained on D_{pre} and documents written by B, B_{pre}, attributes authorship of $D\prime$ to A with a probability p less than random chance, i.e. $p \leq \frac{1}{N+1}$.

4 Approach

Our writing style anonymization framework consists of two platforms: JStylo and Anonymouth. JStylo is a standalone platform for authorship attribution. It is used as an underlying feature extraction and authorship attribution engine for the anonymization framework. Anonymouth is the writing style anonimization platform. It uses the extracted stylometric features and classification results obtained through JStylo and provides suggestions to users to anonymize their writing style.

4.1 JStylo: An Authorship-Attribution Platform

JStylo uses NLP techniques to extract linguistic features from documents, and supervised machine learning methods to classify those documents based on the extracted features. JStylo first "learns" the style of known candidate authors based on documents of those authors, and the style of a given set of anonymous documents. It then attributes authorship of the anonymous documents to any of the known authors. JStylo is a Java-based open-source software with a graphic user interface and an extendable API.

Structure and Usage. The main work-flow of JStylo consists of four consecutive phases: defining a problem set, defining a feature set, selecting classifiers and running the analysis.

A problem set is defined by a training corpus, constructed of documents of all potential authors (as it is supervised learning), and a set of documents of unknown authorship whose authorship are to be determined.

A feature set is defined by a set of various stylistic features to be extracted from the text. Currently there are just above 50 different configurable features available, spanning over different levels of the text, like parts-of-speech in the syntactic level or word frequencies in the lexical level.

The current version of JStylo supports three pre-defined feature sets: Basic-9, Writeprints, and Writeprints (limited). The Basic-9 feature set consists of the nine features that were used in the neural network experiments in [4]. The Writeprints feature set consists of the features used for the Writeprints technique [1]. The Writeprints (Limited) feature set consists of the same features used for Writeprints, where feature classes with potential of exceeding 50 features (e.g. letter bigrams, with a potential of 26^2 features) are limited to the top 50 features. The documents in the training set are mined for the selected features, which are later used for training the classifier, basically profiling the stylistic characteristics of each candidate author. The same features are mined in the test set, for later classification by the trained classifiers.

Each feature is defined by 1) optional text pre-processing tools that allow various filtering methods, to be applied before the feature extraction (e.g. stripping all punctuation); 2) the "core" of the feature which is the feature extractor itself; 3) optional feature post-processing tools to be applied on the features after extraction (e.g. picking the top features frequency-wise); and 4) optional normalization baselines and factoring multipliers (e.g. normalizing over the number of words in each document). The components in 1-3 are based on the JGAAP API [14].

The classifiers available for selection are a subset of Weka [15] classifiers commonly used, such as support vector machine, Naïve Bayes, decision tree, etc. There are several analysis configurations available, the main choice being either to run a 10-fold cross validation analysis over the training corpus or to train the classifiers using a training corpus and classifying the test documents.

JStylo as a Stylometry Research Platform. The main advantages and novelties of JStylo are 1) allowing integration of multiple features to represent various stylistic characteristics of documents, and 2) a high level of feature-set customizability, where each feature can be configured with its own text pre-processing tools, feature extractors, feature post-processing tools and normalization methods. Its user-friendly graphic interface and Java API allow a high level of usage across both linguistic researchers and computer scientists, providing a convenient platform for stylometry research.

Details of the performance and accuracy of JStylo as a stylometry research platform are discussed in section 6.1.

4.2 Anonymouth: An Authorship–Anonymization Framework

Anonymouth aims to use the tools of authorship attribution to systematically render them ineffective on a text, while preserving the message of the document in question to the highest degree possible. The task of actively changing the document is however, at this point, left to the user. For Anonymouth to be able to read in a document and output an anonymized version satisfying the constraint that the meaning be preserved, it would need a deep understanding of the structure of the English language (assuming English text), knowledge of almost all words, and a reasonable grasp of things like metaphors and idioms - which is quite a tall order.

After initialization via JStylo, Anonymouth performs an iterative two-step approach to achieve writing style anonymization. These steps are: 1) feature clustering and preferential ordering, and 2) feature modification and document reclassification.

Initialization. Anonymouth requires[4] the user (A) to input three sets of documents: 1) a single document consisting of approximately 500 ± 25 words, the `documentToAnonymize` (D); 2) a set (at least 2, though preferably more) of sample documents written by the user, totaling 6500 ± 500 words, the `userSampleDocuments` (D_{pre}); and 3) a corpus — preferably made up of at least 3 different authors — of sample documents, *not* written by the user, containing 6500 ± 500 words per author, the `otherSampleDocuments` (B_{pre}). The `userSampleDocuments` are used to determine where the `documentToAnonymize`'s features should *not* be, while the `otherSampleDocuments` are used to determine where the `documentToAnonymize`'s features could be moved to.

After an initial classification has been produced (by JStylo), four groups of features result: 1) those extracted from D, `toAnonymizeFeatures`; 2) those extracted from D_{pre}, `userSampleFeatures`; 3) those extracted from B_{pre}, `otherSampleFeatures`; and 4) a combination of the two previous groups, `userAndOtherFeatures`. Anonymouth then runs Weka's information gain method on the `userAndOtherFeatures` to select the top f features according to information gain. These top f features will be used in the subsequent computations to generate suggestions for changing writing style. Among the top f features, any that *are not present* in D are excluded from the suggestions Anonymouth deliveres. Resultantly, f becomes $f\prime$. This is done to provide effective suggestions because it cannot be freely assumed that any given feature can be reasonably added to the document. This only applies to JStylo's Writeprints feature sets, where without excluding the non-existing features from suggestions (as an extreme example), a user might be asked to include the word, "Electromagnetic" — when that particular word has no business appearing in the document the user is interested in anonymizing.

[4] In its present state as a research platform rather than a software designed for an end-user, this is the case. However, these limitations are by no means absolute.

Feature Clustering and Preferential Ordering. Knowing what features to change in order to anonymize a document says nothing about how much to change them, nor does it indicate how much they can be changed and still represent a coherent document that adheres to the rules of grammar. The cause–and–effect relationship among the stylometric features is comparable to that of a field of Dominoes: altering the sentence count of a text will impact the average sentence length; which will affect the Gunning-Fog Readability Index — which happens to be inversely related to the Flesch-Kincaid Reading Ease Score; all of which will inevitably change the character count and will (probably) change the number of occurrences of the three letter word "and". Because of this, it is hard to decide exactly what changes can/should be made in an existing document. However, individually grouping the values of every feature across all B_{pre} seems to provide a fairly decent guideline. It allows Anonymouth to decide how to change each of the $f\prime$ features based upon where the 'real' document's features lie with respect to both one another as well as the user's normal distribution for each feature. The clustering of all instances of each feature assists Anonymouth in selecting physically realizable 'target' values to represent the 'suggested' final document configuration that the user should aim to achieve in order to evade authorship detection. The mechanism behind this selection process is presented through the rest of this section.

Objects containing `otherSampleFeatures` and their respective document names are then fed into a modified k-means clustering algorithm (described in Algorithm 1). The algorithm clusters the objects with respect to each Object's value with, $k = numAuthors$ (where $numAuthors$ is the total number of authors), means, using a modified k-means++ [16] initialization algorithm on a per feature basis spanning across all documents represented by `otherSampleFeatures`. The most significant change to the k-means algorithm is that if any clusters exist with less than three elements after the algorithm converges, the algorithm is re-initialized with $k = k - 1$ means. A more accurate representation might be ak-means. The reasoning behind this adjustment is: because target values for the `documentToAnonymize` are chosen as the centroids of clusters, more elements weighing in on the target value (centroid) increases the potential for anonymization — as opposed to having a single element cluster and effectively copying another's writing style[5]. It remains to be seen whether it would be beneficial to scale the minimum cluster size limit as the number of documents increases; as of now, the value remains fixed.

Implementing these changes in the k-means++ and k-means algorithms creates a safety net that allows Anonymouth to deal with many potential issues that may arise while analyzing an unknown number of documents with unknown similarities/differences. Anonymouth assumes that the documents it receives will

[5] There is no guarantee that each cluster will contain documents from more than one author. However, limiting the minimum cluster size helps increase the chances of this happening. In practice, clusters have been observed to contain documents by more than one author more often than not.

be easily clustered. It will adapt if this is not the case, and produce the most beneficial output it can.

Algorithm 1 The ak-means Clustering Algorithm (Done on a per-feature basis)

```
1. Initialization:
   (a) run k-means++ algorithm to initialize cluster's based on otherSampleFeatures, with the following exceptions:
       i. If 10,000 numbers have been tried before finding a new centroid, restart.
       ii. If all remaining unchosen values are the same, update the number of total centroids to number of current centroids, set $maxCentroidsFound = True$, and exit initialization; nothing else can be done.
   (b) Assign all instances of the current feature (one per document) from otherSampleFeatures to the centroid nearest to it based on one-dimensional euclidean distance. These are the initial clusters.
2. Update Centroids:
   (a) Calculate the average of the elements (features) contained within each cluster, and update that cluster's centroid with the calculated average.
3. Reorganization:
   (a) Calculate the linear distance between each element, and each existing centroid.
   (b) Assign each element to its closest centroid based on the distance calculation in (a).
   (c) If no elements moved:
       i. If $maxCentroidsFound$ is $True$, or there are at least two clusters with no less than 3 elements per cluster, algorithm has converged.
       ii. If there is only one cluster and $maxCentroidsFound$ is $False$, increment $numMeans$, and Initialize.
       iii. If there are any clusters with less than 3 elements and $maxCentroidsFound$ is $False$, decrement $numMeans$, and Initialize.
   (d) Else if elements did move:
       i. Update centroids.
```

Once the ak-means algorithm has converged, clusters are assigned a preference value based on the primary preference calculation, and placed into an $i \times j$ array after being sorted (from least to greatest).

$$p_{i,j} = numElements_{i,j} \times \mid centroid_{i,j} - userSampleMean_i \mid \tag{1}$$

where: $p_{i,j}$ is the primary preference of feature i's jth cluster; $numElements_{i,j}$ is the number of elements in feature i's jth cluster; $centroid_{i,j}$ is the average of feature i's jth cluster's elements; and $userSampleAvg_i$ is the average of the user's sample documents, `userSampleDocuments`, for feature i. The purpose of taking the number of elements into account rather than basing a cluster's preference value off its distance from the user's average values alone is to avoid attempting to modify a user's `documentToAnonymize` to take the form of a document who's features lie in the extremes due to specific content, while refraining from unwittingly restricting the pool of potential target values due to a single feature. Ordering each feature's clusters in such a way that the most desirable cluster has the highest value also lays the groundwork that allows cluster groups to be ordered by a secondary preference calculation.

The secondary preference calculation weights features with respect to their information gain ranking, and ensures that cluster groups that appear with high frequency take precedence over those that appear less often. Because the most desirable cluster, as determined by the primary preference calculation in Eq. (1), has the highest value, weighting the secondary preference calculation in this manner is intended to assign the greatest cluster group preference to the most common cluster group that has the most impact on the features with high information gain. The centroids of the cluster group with the highest ranking are likely to be the best target values for the `documentToAnonymize`. However, because the primary and secondary preference calculations have not been completely optimized, it is possible that the actual best target cluster will be found slightly further down the list of cluster group preferences. For this reason, as well as to help validate the approach by graphically displaying the workings of Anonymouth, the Clusters tab was created.

The "Clusters" tab, as seen in Fig. 1, displays the clusters formed by the Algorithm 1, represented by the empty green ellipses which contain clusters of blue dots representing the `otherSampleFeatures`. A shaded purple ellipse displays the user's confidence interval (CI) for a given feature. CI is computed using the following formula,

$$CI = D_{pre_{mean}} \pm 1.96 \times \sigma \tag{2}$$

where, $D_{pre_{mean}}$ = average of all `userSampleDocuments`($\mathrm{D_{pre}}$), and σ = standard deviation from the mean. The visible red dot displays the present value of the same feature in the `documentToAnonymize`, which Anonymouth tries to 'put' in the most populated location as far away from the purple shaded ellipse as possible. By selecting one of the available cluster configurations from the drop-down menu, the user may view configurations from, P_{CG_0} (which should provide the greatest potential to anonymize the document) to $P_{CG_{u-1}}$ (which should provide the least potential to anonymize the document), where u is the number of unique document cluster configurations, and P_{CG_n} is the nth document cluster group's cluster group preference. Upon choosing a configuration, one cluster per feature will be shaded green, and is the target cluster for that feature within that configuration. When a cluster configuration is selected, each target cluster's centroid — represented by the empty black circle — is set to be the target value for each feature (respectively) within the `documentToAnonymize`.

One might ask, why not simply pick the cluster farthest away from the author's average value for each feature? The danger in doing this, as has been determined experimentally, is that many features are co-dependent upon one another; so, it may be physically impossible to modify a document to be represented by a set of independently chosen features. For example, it is impossible to increase the average sentence length of a document, while increasing the number of sentences (assuming the idea is to keep the document more or less the same length). Target values for features must be selected while being mindful of other features. Why, then, not just use the document with a cluster group configuration farthest from the author's standard set of values? This is done because ideally it is more feasible to alter an existing document to look 'generic' than it is to attempt to drive it toward an extreme, which may only be that way as a result of content (including unusual errors that one might have a hard time trying to, or would not want to, reproduce). If many documents share more or less the same configuration, there is a greater chance that any given document can also be fit to share that configuration while maintaining readability. Furthermore, changing a document to look more like many other documents should be more effective in making it anonymous than simply altering it to look as much unlike the true author's work as possible.

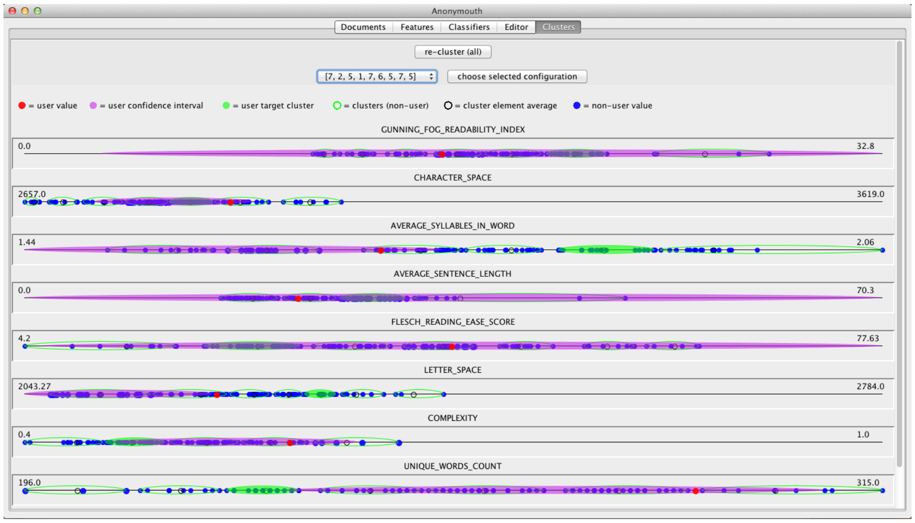
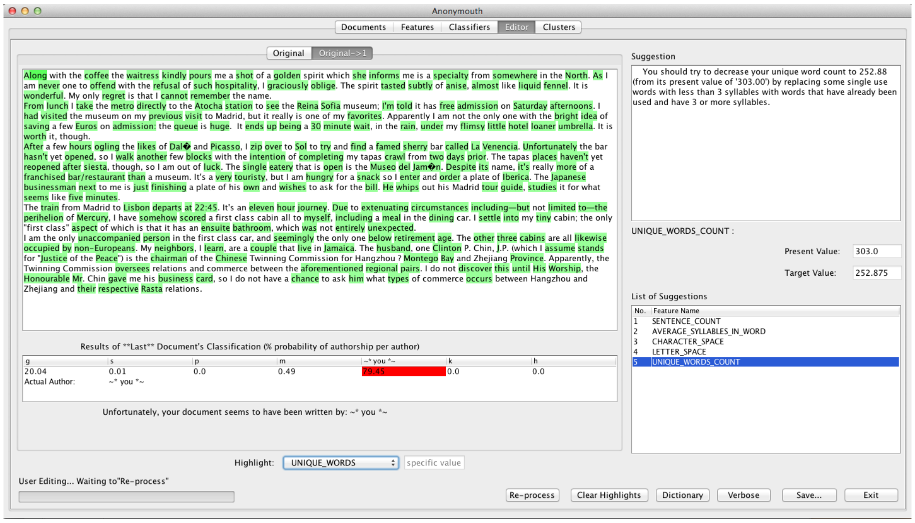


Fig. 1. Anonymouth Clusters tab with cluster group P_{CG_0} selected, using the Basic-9 feature set, with 6 'otherSample' authors. The red circles display the present value of each feature within the user's 'documentToAnonymize', the purple ellipses depict the user's confidence interval (assuming a normal distribution) for each feature, and the shaded green ellipses show where the user's feature's will fall if all features are modified as recommended by Anonymouth.

Feature Modification and Document Reclassification. Once the targets are selected, the user is presented with a clickable list of features to change. When a feature is selected, a suggestion appears that aids the user in changing the present value of the feature to its target value. The suggestions for the Basic-9 feature set have been optimized to guide the user to change the elements in their document that will have the greatest overall impact on its classification.

An example of this is, "[replace] some single use words with less than 3 syllables with words that have already been used and have 3 or more syllables", as seen in Fig. 2. Once the document has been changed so that its present values reflect the target values, the document is reclassified. If the document has reached a sufficiently low classification[6] the document is considered anonymized. Until that point, the process loops back to 'feature clustering and preferential ordering.' Every time the features are clustered, slightly different clusters may result; which leads to changing target values. We found that in some cases (especially for the Writeprints features) clustering the features only once is a better alternative to continually re-clustering the features upon every classification.



Along with the coffee the waitress kindly pours me a shot of a golden spirit which she informs me is a specialty from somewhere in the North. As I am never one to offend with the refusal of such hospitality, I graciously oblige. The spirit tasted subtly of anise, almost like liquid fennel. It is wonderful. My only regret is that I cannot remember the name.
From lunch I take the metro directly to the Atocha station to see the Reina Sofia museum; I'm told it has free admission on Saturday afternoons. I had visited the museum on my previous visit to Madrid, but it really is one of my favorites. Apparently I am not the only one with the bright idea of saving a few Euros on admission: the queue is huge. It ends up being a 30 minute wait, in the rain, under my flimsy little hotel loaner umbrella. It is worth it, though.
After a few hours ogling the likes of Dal� and Picasso, I zip over to Sol to try and find a famed sherry bar called La Venencia. Unfortunately the bar hasn't yet opened, so I walk another few blocks with the intention of completing my tapas crawl from two days prior. The tapas places haven't yet reopened after siesta, though, so I am out of luck. The single eatery that is open is the Museo del Jam�n. Despite its name, it's really more of a franchised bar/restaurant than a museum. It's a very touristy, but I am hungry for a snack so I enter and order a plate of Iberica. The Japanese businessman next to me is just finishing a plate of his own and wishes to ask for the bill. He whips out his Madrid tour guide, studies it for what seems like five minutes.
The train from Madrid to Lisbon departs at 22:45. It's an eleven hour journey. Due to extenuating circumstances including—but not limited to—the perihelion of Mercury, I have somehow scored a first class cabin all to myself, including a meal in the dining car. I settle into my tiny cabin; the only "first class" aspect of which is that it has an ensuite bathroom, which was not entirely unexpected.
I am the only unaccompanied person in the first class car, and seemingly the only one below retirement age. The other three cabins are all likewise occupied by non-Europeans. My neighbors, I learn, are a couple that live in Jamaica. The husband, one Clinton P. Chin, J.P. (which I assume stands for "Justice of the Peace") is the chairman of the Chinese Twinning Commission for Hangzhou ? Montego Bay and Zhejiang Province. Apparently, the Twinning Commission oversees relations and commerce between the aforementioned regional pairs. I do not discover this until His Worship, the Honourable Mr. Chin gave me his business card, so I do not have a chance to ask him what types of commerce occurs between Hangzhou and Zhejiang and their respective Rasta relations.

Results of **Last** Document's Classification (% probability of authorship per author)

g	s	p	m	~* you *~	k	h
20.04	0.01	0.0	0.49	79.45	0.0	0.0
Actual Author:	~* you *~					

Unfortunately, your document seems to have been written by: ~* you *~

Suggestion

You should try to decrease your unique word count to 252.88 (from its present value of '303.00') by replacing some single use words with less than 3 syllables with words that have already been used and have 3 or more syllables.

UNIQUE_WORDS_COUNT :

Present Value: 303.0

Target Value: 252.875

List of Suggestions

No.	Feature Name
1	SENTENCE_COUNT
2	AVERAGE_SYLLABLES_IN_WORD
3	CHARACTER_SPACE
4	LETTER_SPACE
5	UNIQUE_WORDS_COUNT

Highlight: UNIQUE_WORDS

User Editing... Waiting to"Re-process"

Fig. 2. Anonymouth Editor tab showing the 'Unique Words Count' suggestion selected, with unique words (only used once) highlighted, and an initial classification attributing authorship of the displayed text to the user of the Anonymouth with 79.5% probability.

The Editor tab contains a 'Dictionary' which brings up an interface to Princeton's WordNet 3.0, allowing a user to search for synonyms and words containing various continuous character strings (e.g. 'ie'). A 'verbose' button will bring up a window that prints Anonymouth's standard output and error streams in real time as well. Finally, should the user want to revert back to a previous copy of the `documentToAnonymize`, tabs that display where each copy of the document originated from permit the user to trace back through processed changes, while viewing each document's classification results.

[6] In this case, a sufficiently low classification means at or below random chance, which is $1/(numAuthors)$, where $numAuthors$ is the total number of authors.

5 Anonymouth User Study

We performed a user study with 10 participants to understand the effectiveness of Anonymouth in changing writing style. We evaluated Anonymouth based on its effectiveness in anonymizing a document and its ease of use from a user's perspective.

We asked participants to anonymize their pre-written writing samples. Anonymizing a pre-written text is more difficult than writing in a changed style from the start. Stylometry methods fail to attribute authorship when people write in a different style [4]. We wanted to evaluate how much anonymity can be achieved by changing the writing style of a pre-existing document. The subjects were asked to submit 6500 words of pre-existing writing, along with a document of approximately 500 words to modify. The 6500-word sample was used as the subject's training sample. We chose to use 6500 words of writing as it has been found to be enough to leak the identity of an author [10]. As a background corpus, we used regular writing samples of six authors from the Brennan-Greenstadt adversarial corpus. Anonymouth allows the user to choose any corpus as a background corpus. But for the purpose of the study we fixed the background corpus for all users.

During the experiment, each user was asked to use the Basic-9 feature set and SMO SVM classifier as authorship attribution method. Anonymouth provides the option of choosing any feature set and any classifier. The reason we fixed the feature set is because changing an existing document with Writeprints features is very hard, and after the first two participants failed to follow the suggestion for changing Writeprints features we decided not to use it in this study. We excluded the Writeprints result of the first two participants. We only used all of the 10 participants results with Basic-9 features. SMO SVM is used for its high accuracy. The users were asked to perform classification of their document, choose the appropriate cluster from the clustering window and change the document based on Anonymouth's suggestion. We suggested the users to repeat the process for one hour, or until the result of the classification goes below random chance (which was 14% accuracy in our case).

After a user successfully anonymized his/her document or used Anonymouth for an hour, we asked them to rate several aspect of Anonymouth on a 10-point Likert scale. The survey asked basic demographics questions.

6 Evaluation and Results

This section discusses the results of the Anonymouth user study. The following subsections explain the effectiveness of JStylo in attributing authorship, effectiveness of Anonymouth in anonymizing a document, the effect of the choice of background corpus and feature set on anonymity, which features were changed by the users to achieve anonymity, and the user satisfaction survey.

6.1 Effectiveness of JStylo

To evaluate the effectiveness of JStylo as a sufficiently accurate authorship attribution engine for Anonymouth and as an authorship attribution research platform in general, we conducted experiments using the Brennan-Greenstadt Adversarial Stylometry Corpus, which includes 13 authors with 5000-word documents each. We then compared the results with those of two other state-of-the-art authorship attribution methods in the literature: the Writeprints method and the synonym-based approach [17]. The experiments with JStylo were conducted using a SVM classifier, over two feature sets: the Basic-9 and the Writeprints (Limited). All experiments were evaluated using 10-folds cross-validation. The results are summarized in table 1.

Table 1. Authorship attribution results using Writeprints, Synonym-based and JStylo.

Method	*Accuracy*
Writeprints	73.33%
Synonym-based	89.61%
JStylo with Basic-9	53.98%
JStylo with Writeprints (Limited)	92.03%

Although the Basic-9 feature set did not produce as high results as the other methods, it is still much higher than random chance (7.69%), and is used only as baseline for authorship attribution features in JStylo, or anonymization features baseline in Anonymouth. It is notable that using the Writeprints (Limited) feature set with JStylo produced the highest results across all four experiments.

6.2 Effectiveness of Anonymouth

Figure 3 shows authorship attribution accuracy of the modified and unmodified documents. Using the Basic-9 features 80% participants were able to anonymize their documents in terms of the corpus used. The first participant's (s1) original document was not attributed to him as an author. The second participant (s2) made no changes to his document. All other participants were able to anonymize their documents.

6.3 Effect of the Background Corpus on Anonymity

The background corpus, or set of reference authors and documents, is important for document anonymization with Anonymouth as the tool calculates the average value of each feature based on the background corpus and suggests changes to users based on the average feature values.

We tested if documents anonymized in terms of one background corpus are also anonymized against a different background corpus. To test this, we used a

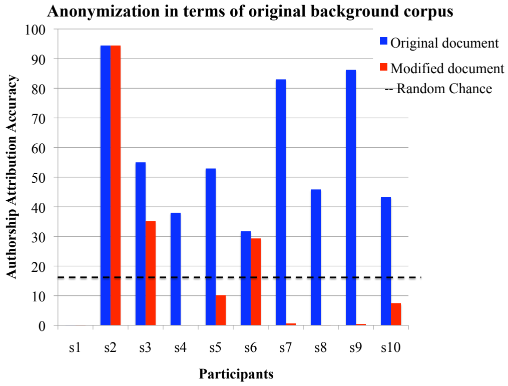


Fig. 3. Authorship attribution accuracy of modified and original documents using the original background corpus. The Basic-9 feature set and SMO SVM classifer were used. All subjects who made changes were able to anonymize their documents (8/10).

different six author subset from the Brennan-Greenstadt adversarial corpus. We also tested the results using the whole 13-author corpus. Results are shown in Figure 6.3 (a) and Figure 6.3 (b). The effectiveness of the anonymization changes if the background corpus is changed. Unfortunately, the basic 9-Feature set is not very effective at stylometry. Where possible, we pre-selected documents that were correctly classified for the anonymization with respect to the original background corpus. However, when we switched to the new background corpus, only four of these were correctly classified. Of these four, 50% (2) of the authors' documents were still anonymized even in terms of a different corpus of six authors and the others remained anonymized (as shown in Figure 6.3 (a)). For the corpus of 13 authors, 5 subjects' original documents were classified correctly and all modified documents were classified incorrectly (as shown in Figure 6.3(b)).

6.4 Effect of Feature Set on Anonymity

We wanted to see if documents anonymized with one authorship attribution approach are detectable by another approach. Unfortunately in every case documents anonymized with Basic-9 features were attributed to the real author when Writeprints (Limited) feature were used. The Writeprints feature set is much larger than Basic-9, contains around 700 linguistic, content specific and structural features. Most of these features are very low level features, for example, frequencies of character uni-/bi-/tri-grams. Providing effective suggestions for such low level features is challenging. Changing existing documents by following those suggestions to hide author specific features is also very difficult. For this reason, none of the participants in our study were able to anonymize themselves using the Writeprints (Limited) features.

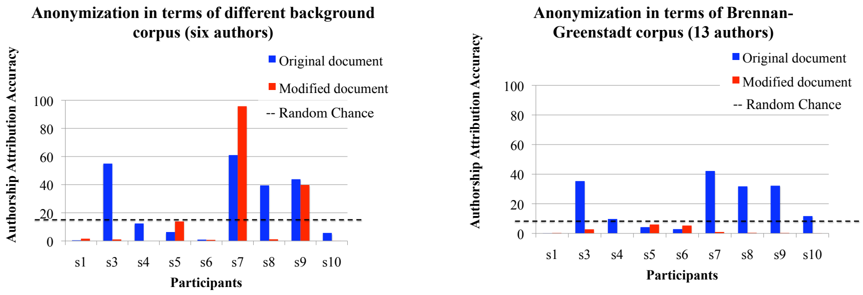


Fig. 4. Authorship attribution accuracy of modified and original documents using six different author samples as background corpus 6.3 (a) and 13 authors as background corpus 6.3 (b). The Basic-9 feature set and SMO SVM classifier were used.

We wanted to evaluate functionality of Anonymouth using the Writeprints (Limited) features to find out the minimum number of features that need to be changed to anonymize a document. To do this, we first ranked the features based on information gain ratio [18]. Then we calculated clusters of feature values using Anonymouth. We chose the top K features based on information gain ratio and changed their values with those of the first cluster, where K= 25, 50, 75, ..., 300. Result of the experiment is shown in Figure 5.

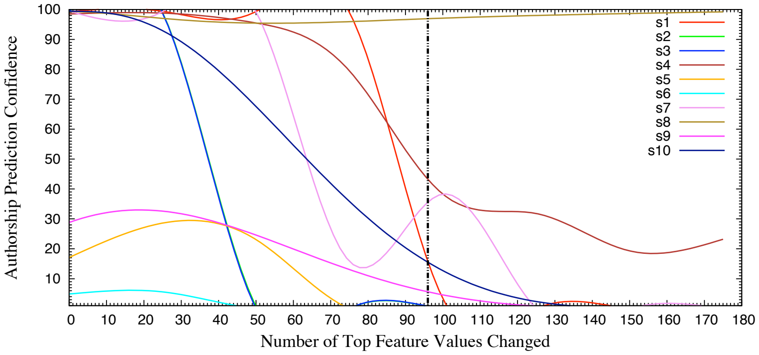


Fig. 5. Number of Writeprints (Limited) features needs to be changed to achieve anonymity. Authorship prediction accuracy decreases as the top feature values are replaced with those predicted by Anonymouth. Sufficient anonymity can be achieved after changing 13.79% (96 out of 696) of the features.

The result shows that authorship prediction accuracy decreases as the top feature values are replaced with the values predicted by Anonymouth. After changing 13.79% of the features, 90% of the documents were anonymized. This experiment shows that the core approach of Anonymouth works successfully to anonymize a document even against a robust feature set like Writeprints.

6.5 Change in features

We compare the frequencies of different features to understand which ones people change to anonymize their writing style. The changes made to features are shown in Figure 6. We only used samples of the participants who were successful in anonymizing their documents. This graph illustrates the changes in frequencies for each feature. The y-axis contains a list of features that have been adjusted in the passages and x-axis of the graph denotes the change in each feature. We compute the change in feature using the following formula:

Change in Feature f, $C_f = (f_{mod} - f_{ori})/(f_{ori})$
where,
f_{mod} = Value of feature f in the modified document.
f_{ori} = Values of feature f in the original document.

The amount to the right of the y-axis represents the increases in a feature and the amount to the left represents the decreases. 87.5% of the successful participants (7/8) increased average sentence length and decreased sentence count. Average syllable count was increased in 75% of the cases. Increase in complexity was also noticed in every anonymized document. This indicates that most participants made their language complicated to anonymize their documents, which is also evident by the increase of the Gunning-Fog (GF) readability index.

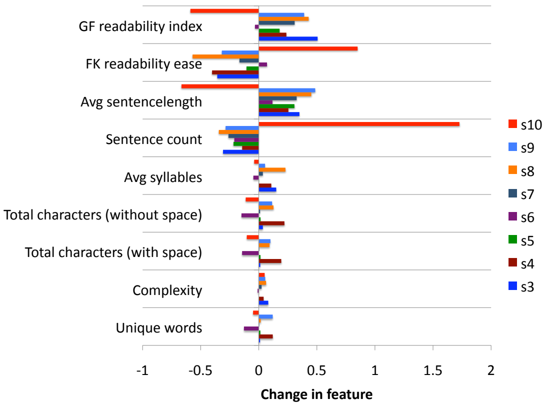


Fig. 6. Feature changes to achieve anonymity

This result differs from the feature changes in the Brennan-Greenstadt adversarial documents where participants used simpler language to obfuscated their document [5]. Ideally a document can be anonymized by using a language that is either more complex or less complex than the original writing style of an author. As seen in [5], people usually use less complex language while obfuscating their writing style, which is easily distinguishable from regular writings. Anonymouth allows user to choose his own background corpus and provide suggestions to change his writing style. Thus by choosing a diverse background corpus an author can hide both his writing style and the indication of changing style.

6.6 User Experience Survey

We had 10 participants in the study within 18-45 age limit who are daily computer users. 2 of them were females and 8 of them were males. On average, the users considered themselves to be moderately good writers. The participants all either had or were working on college degrees, most with different majors, and 90% of them were native English speakers. None of the subjects had any previous knowledge of linguistics or stylometry.

The detailed evaluation of Anonymouth was covered in the first part of the survey which had a 10-point scale, with '0' being the lowest/worst and '9' being the highest/best. The following graph summarizes the reaction of the subjects to Anonymouth that was captured in the first part of the survey.

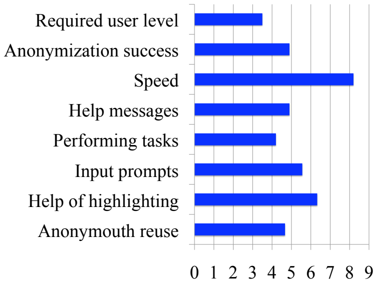


Fig. 7. Anonymouth user experience survey

On average, participants found Anonymouth was user-friendly and that it did not require any specific background knowledge to use. Anonymouth's word highlighting feature was highly rated as helpful. The speed of Anonymouth was considered very fast. Participants felt Anonymouth was moderately successful in anonymizing documents (rated 4.9 on Likert chart). 7 of the participants said they would recommend Anonymouth to other people.

7 Discussion

Although it appears to be quite challenging for a user to implement the changes that Anonymouth asks for, even when only using Brennan and Greenstadt's 'Basic-9' feature set, preliminary results suggest that when users are able to do what is asked, they *can* successfully anonymize their documents — with respect to that feature set. As shown in Fig. 3, 80% of participants were able to reduce the accuracy of the SVM classifier used with respect to the original background corpus used. Furthermore, 60% of participants succeeded in achieving a final classification probability below random chance, which for a total of 7 authors is just under 14.3%.

Initial user tests using the 'Writeprints (Limited)' feature set implemented by JStylo suggested less usability than existed when using the Basic-9 feature set in terms of users being able to perform the actions requested by Anonymouth. Due to the complex nature of the Writeprints (Limited) feature set, the user is asked to do things like add more of the letter 'i' to his/her document, or to decrease the number of occurrences of a part of speech n-gram. While no one was able to anonymize their document with respect to the Writeprints (Limited) feature set, it has been shown that in general, if approximately 15% of possible features are changed to the values determined by Anonymouth, a document initially classified as having been written by its actual author with 98% probability, will — about 80% of the time — end up being classified as having been written by another author with over 95% probability.

This suggests that the core of Anonymouth — the methods used to determine what and how much should be changed within a document — have some merit. That is not to say that Anonymouth's core has either been optimally adjusted or is in fact the best way to decide how to anonymize a document. There is a clear separation between knowing the degree to which certain things need to be changed, and being able to execute those changes. Resulting from finding that Anonymouth's suggestions to the user regarding how to make these changes need re-working, it is quite possible that Anonymouth's algorithms will need to be re-worked as well.

7.1 Future Work

In general, it seems as though the information presented to the user should be of a higher level, such as, "re-write this sentence using the third person and in the past tense". Of course, doing just this is not the solution. In attempting to anonymize a document via a set of naïve algorithms, there appears to be a trade-off between anonymity and affect. Assuming that the author of a document has written that document in a style that he usually writes in, it is very difficult for that author to go back to another document and modify it to then appear in a different style, while retaining the document's meaning (it is assumed that in order to retain meaning, the imagery and tone would have to create the same end result). Simply stripping descriptive words, modifying tense, and altering the point of view (e.g. from third to first person) would certainly increase anonymity; though clearly at the expense of the documents impact on the audience (affect). While this is one approach that may be taken, it seems far from ideal, and as though it ought to be considered as a last resort.

To achieve its goal, Anonymouth must be able to understand what a sentence/passage means to the extent necessary to enable it to produce an output passage expressed using language constructs foreign to the original author's work that can *at least* capture the main idea and tone of the original passage. While a perfect system would be quite challenging to implement, constructing a system that offers a list of potentially reasonable alternatives to a given passage seems to be a realizable goal.

Adding these features to Anonymouth would resolve the current usability problem that limits the application of Anonymouth.

8 Conclusion

This paper presents Anonymouth, a novel framework for anonymizing writing style.Without accounting for style, anonymous authors risk identification. This framework is necessary to provide a tool for testing the consistency of anonymized writing style and a mechanism for adaptive attacks against stylometry techniques. Our framework defines the steps necessary to anonymize documents and implements them via Anonymouth and JStylo. These are (1) Analysis of the documents using authorship attribution techniques relative to a corpus of text and a set of linguistic features (implemented via JStylo), (2) Determining the features that need to be changed (using information gain), (3) Ordering the features to be changed and determining where they need to go (using our modified k-means clustering approach), (4) Suggesting changes to achieve these changes (via Anonymouth). We have shown that these steps are effective, that users who make the suggested changes do anonymize their documents relative to the suggested feature sets, and that Anonymouth does help users reduce the accuracy of stylometry techniques. However, our user studies suggest that step 4 is quite difficult and significant research remains to determine the best way to suggest changes that are easy to apply, especially for the large and complex feature sets that result in the highest accuracy. It is not so easy to use fewer 'i's.

This paper presents the first study that evaluates modifying pre-written documents to anonymize style. We found that this was much more difficult than creating anonymous documents from scratch. Human subjects can create a document that evades multiple state-of-the-art authorship techniques in 30-60 minutes without access to those techniques [4]. However, an hour was sometimes not enough to anonymize pre-existing documents in reference to a limited feature set and the changes made did not transfer to analysis with other feature sets, showing that the choice of feature set is critical when using a tool like Anonymouth. More research is needed to better anonymize pre-existing documents as people do not often write with anonymity in mind and may wish to publish documents previously written without compromising their identity.

Acknowledgements. We thank DARPA (grant N10AP20014) and the Intel Science and Technology Center (ISTC) for Secure Computing for supporting this work. We are grateful to our colleagues Pavan Kantharaju, and Michael Brennan for their invaluable suggestions on the project. We also thank our shepherd Arvind Narayanan and the anonymous reviewers for their useful feedback.

References

1. Abbasi, A., Chen, H.: Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace. ACM Trans. Inf. Syst. **26**(2) (2008) 1–29
2. Narayanan, A., Paskov, H., Gong, N., Bethencourt, J., Stefanov, E., Shin, R., Song, D.: On the feasibility of internet-scale author identification. In: Proceedings of the 33rd conference on IEEE Symposium on Security and Privacy, IEEE (2012)
3. Wayman, J., Orlans, N., Hu, Q., Goodman, F., Ulrich, A., Valencia, V.: Technology assessment for the state of the art biometrics excellence roadmap. http://www.biometriccoe.gov/SABER/index.htm (March 2009)
4. Brennan, M., Greenstadt, R.: Practical attacks against authorship recognition techniques. In: Proceedings of the Twenty-First Innovative Applications of Artificial Intelligence Conference. (2009)
5. Afroz, S., Brennan, M., Greenstadt, R.: Detecting hoaxes, frauds, and deception in writing style online. In: Proceedings of the 33rd conference on IEEE Symposium on Security and Privacy, IEEE (2012)
6. Eckersley, P.: How unique is your web browser? In: Privacy Enhancing Technologies, Springer (2010) 1–18
7. Narayanan, A., Shmatikov, V.: Robust de-anonymization of large sparse datasets. In: Proceedings of the 29th IEEE Symposium on Security and Privacy, IEEE (2008) 111–125
8. Calandrino, J., Kilzer, A., Narayanan, A., Felten, E., Shmatikov, V.: You might also like: Privacy risks of collaborative filtering. In: Proceedings of the 32th IEEE Symposium on Security and Privacy, IEEE (2011) 231–246
9. Dingledine, R., Mathewson, N., Syverson, P.: Tor: The second-generation onion router. In: Proceedings of the 13th conference on USENIX Security Symposium-Volume 13, USENIX Association (2004) 21–21
10. Rao, J., Rohatgi, P.: Can pseudonymity really guarantee privacy. In: Proceedings of the Ninth USENIX Security Symposium. (2000) 85–96
11. Kacmarcik, G., Gamon, M.: Obfuscating document stylometry to preserve author anonymity. In: Proceedings of the COLING/ACL on Main conference poster sessions, Association for Computational Linguistics (2006) 444–451
12. Juola, P.: Authorship attribution. Foundations and Trends in information Retrieval **1**(3) (2008) 233–334
13. Uzuner, U., Katz, B.: A comparative study of language models for book and author recognition. In: IJCNLP. (2005) 969
14. Juola, P.: Jgaap, a java-based, modular, program for textual analysis, text categorization, and authorship attribution
15. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.: The weka data mining software: an update. ACM SIGKDD Explorations Newsletter **11**(1) (2009) 10–18
16. Arthur, D., Vassilvitskii, S.: k-means++: the advantages of careful seeding. In: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. SODA '07, Philadelphia, PA, USA (2007) 1027–1035
17. Clark, J.H., Hannon, C.J.: A classifier system for author recognition using synonym-based features. In: Lecture Notes in Computer Science. Volume 4827., Springer (2007) 839–849
18. Quinlan, J.: Induction of decision trees. Machine learning **1**(1) (1986) 81–106