Proceedings of the Sixteenth International AAAI Conference on Web and Social Media (ICWSM 2022)

Using Authorship Verification to Mitigate Abuse in Online Communities

Janith Weerasinghe[1], **Rhia Singh**[2], **Rachel Greenstadt**[1]

[1] New York University,
[2] Macaulay Honors College (Hunter CUNY)
janith@nyu.edu, rhia.singh@macaulay.cuny.edu, greenstadt@nyu.edu

Abstract

Social media has become an important method for information sharing. This has also created opportunities for bad actors to easily spread disinformation and manipulate public opinion. This paper explores the possibility of applying Authorship Verification on online communities to mitigate abuse by analyzing the writing style of online accounts to identify accounts managed by the same person. We expand on our similarity-based authorship verification approach, previously applied on large fanfictions, and show that it works in open-world settings, shorter documents, and is largely topic-agnostic. Our expanded model can link Reddit accounts based on the writing style of only 40 comments with an AUC of 0.95, and the performance increases to 0.98 given more content. We apply this model on a set of suspicious Reddit accounts associated with the disinformation campaign surrounding the 2016 U.S. presidential election and show that the writing style of these accounts are inconsistent, indicating that each account was likely maintained by multiple individuals. We also apply this model to Reddit user accounts that commented on the WallStreetBets subreddit around the 2021 GameStop short squeeze and show that a number of account pairs share very similar writing styles. We also show that this approach can link accounts across Reddit and Twitter with an AUC of 0.91 even when training data is very limited.

Introduction

Social media and online forums play a critical role in the current society. They allow people to come together and discuss various topics that can influence real-world events. One such recent example is the short squeeze of GameStop stocks, mainly organized by the Reddit community on the WallStreetBets subreddit (Herrman 2021). While discussions and activism facilitated on online communities are vital to society, social media platforms are also being used to spread misinformation and disinformation (Jiang et al. 2020), manipulate public opinion (Starbird 2017; Weld, Glenski, and Althoff 2021), and harass other users (Blackwell et al. 2018; Redmiles, Bodford, and Blackwell 2019). often, bad actors create multiple accounts for these activities. However, there are many legitimate reasons why an individual might maintain multiple accounts on the same platform such as to separate work-related and private content or to separate content based on interests. The ability to identify clusters of accounts maintained by the same person (often referred to as sockpuppet accounts), detect ban evaders (users who create new profiles when they get banned), and linking accounts across social media platforms belonging to the same person for investigations, might help mitigate these abuses.

An author makes conscious and unconscious decisions about the words they choose, the structure of their sentences, and other aspects of language, distinct from the content of their writing. These differences, which form one's writing style, can uniquely identify the author of a document. The study of analyzing the linguistic style is called Stylometry. One area of interest in stylometry is authorship verification (AV). This is the task of comparing the writing style of documents to predict if the same person wrote them. A variety of AV approaches have been presented over the past years (Stamatatos 2009; Koppel, Schler, and Argamon 2009). In this paper, we expand upon our simple yet effective similarity-based approach (Weerasinghe et al. 2020; 2021) for AV that was shown to perform well at previous PAN authorship verification shared tasks (Kestemont et al. 2020). PAN is a stylometry-related workshop series that allows researchers to present models that solves various stylometry-related tasks. The AV shared task was to predict, given two fanfictions, if they were written by the same author. While AV has been applied successfully to larger documents such as novels and blog posts, its application on social media platforms and online forums is still challenging. Previous research (Halvani, Graner, and Regev 2020b) have shown that some approaches perform poorly when tested under more challenging conditions, such as on shorter documents, and diverse topics. Therefore we evaluate and extend our previous work to show that it works well on online communities, perform well with smaller documents and less training data, is largely topic-agnostic, and maintains its performance in settings with a moderate class imbalance.

We conduct our experiments on two primary datasets, the PAN fanfictions dataset and a Reddit comments dataset that we collected, and show that our approach can achieve AUCs greater than 0.95 and up to 0.98. Our Reddit models achieved an AUC of 0.95 on documents with only 40 Reddit comments, performing better than previous models (Halvani, Graner, and Regev 2020b). We also show that our model remains largely topic-agnostic by ensuring that our

features are less likely to encode topic-specific concepts and investigating misclassifications to verify that topic-related biases are minimal. We also perform a feature analysis that shows that top features do not include any content-specific features. This analysis also reveals interesting facts about what linguistic signals remain consistent across different documents by the same author. We also explore how our models work in settings with high class imbalance, a problem that is common in many abuse detection settings, in which there are fever instances of the positive class. Our experiments show that our model performs well under moderate class imbalances, but the performance degrades scenarios with extreme class imbalance. This analysis helps us understand the scenarios in which our model can be applied reliably.

We apply our approach to scenarios where AV can play a role in mitigating online abuse. We show our AV approach can link accounts across Reddit and Twitter with an AUC of 0.91. Our cross-platform model was trained on a much smaller dataset of 680 users showing that this approach is able to handle challenging datasets gracefully. We then apply our model to two datasets on Reddit to identify potential sockpuppet accounts based on similar writing styles. We applied our model to a list of suspicious user accounts identified by Reddit around the 2016 US Presidential election (Reddit 2017) believed to be operated by the Russian IRA. Our results show that the writing style of these accounts are not consistent even across the same user, indicating that each account was probably maintained by multiple people. While the details around how the IRA operated these troll farms are scarce, we believe that our research sheds some light on how these accounts were maintained. We also applied our model to user accounts that participated on Reddit's WallStreetBets subreddit to identify sockpuppets. Our model also predicts that 205 user account pairs out of the 7,318 accounts evaluated on the WallStreetBets subreddit had very similar writing styles.

Related Work

Stylometry and Authorship Verification: Stylometry is the analysis of the writing style of a document. Authorship Attribution, the task of identifying the author of a given document from a set of authors (Stamatatos 2009; Koppel, Schler, and Argamon 2009) and authorship verification, the task of predicting if a document pair (or a pair of document sets) is written by the same person, are two sub-problems in stylometry. Bouanani and Ismail (2013) and Tempestt et al. (2017) provide surveys of the current state of stylometry and identify several sub-tasks related to stylometry. The PAN workshop series (Bevendorff et al. 2020; Daelemans et al. 2019), which organizes stylometry-related shared tasks enabled researchers to compare models effectively.

Both Stamatatos (2016) and Halvani et al. (2019) assess existing AV approaches and characterizes them based on various properties. Multiple approaches such as outlier detection (Nirkhi, Dharaskar, and Thakare 2016), threshold learning methods (Potha and Stamatatos 2014), deep-learning approaches (Boenninghoff, Nickel, and Kolossa 2021), meta-learning (Koppel and Schler 2004; Kestemont et al. 2012), and compression models (Veenman and Li 2013) have been used for AV with varying degrees of success.

In this work, we build upon the approach we introduced at the PAN 2020 and 2021 shared tasks that uses a vector-similarity-based model. We will describe this approach in detail in the next section. Several other AV approaches (Burrows 2002; Hoover 2004) use a similarity-measure in their model. The most similar to our approach is TAVeer (Halvani, Graner, and Regev 2020a), which uses absolute vector difference as a similarity measure, learns thresholds for each feature, and aggregate them to make predictions. Halvani et al. (2020b) compared their approach with ten AV approaches on four datasets, including a Reddit posts-based dataset, and found that their approach is close or outperforms the other approaches. The same approach was submitted to the PAN 2020 shared task (Halvani, Graner, and Regev 2020a), in which our submission performed better. We believe this difference could be attributed to the stylometric features in our approach and the use of a classifier instead of the thresholding-based method. The current state-of-the-art AV approach, based on PAN 2021 results (Kestemont et al. 2021) is AdHominem by Boenninghoff et al. (2021), in which they present a deep learning method using a siamese network. This network consists of two identical neural networks whose output is fed into a distance measure which is then used to make the final prediction. While this approach outperformed our model for the larger dataset, our model for the smaller dataset outperformed the AdHominem model. Current AV research indicates that vector-similarity-based approaches and siamese-network-based deep learning methods perform best. While deep learning methods perform better given large amounts of data, simple vector-similarity-based methods are able to achieve similar performance with smaller datasets which is useful in domains with less training data.

Stylometry in Account Linking: Both authorship attribution and verification methods have been used to link accounts with similar writing styles. Doppelgänger Finder (Afroz, Brennan, and Greenstadt 2012) links accounts by training multiple authorship attribution models by excluding each author and seeing how often the documents of two authors are classified to each other. Several other studies (Almishari et al. 2014; Solorio, Hasan, and Mizan 2013; Vosoughi, Zhou, and Roy 2015; Bu, Xia, and Wang 2013; Johansson, Kaati, and Shrestha 2013) have used linguistics and other attributes such as temporal patterns to successfully link user accounts. Kumar et al. (2017) analyzed sockpuppet accounts on Disqus, and used activity, community, and linguistic features to identify sockpuppet accounts on a balanced dataset with an AUC of 0.91. Cross-platform account linking is a more difficult problem due to the linguistic differences across different social media platforms. Overdorf et al. (2016) used a modified version of the Doppelgänger Finder to link accounts across Twitter, blogs, and Reddit.

Compared to previous approaches, which were mostly evaluated under one domain setting, we will show that our

	Mean	Std. dev
Reddit Dataset (46,465 users):		
Subreddits per user	2.68	1.17
Comments per subreddit	111.55	138.65
Tokens per comment	31.31	57.61
Characters per comment	150.58	282.96
Tokens per document	3457.36	6155.25
Characters per document	16734.67	32164.10
PAN Dataset (550,972 documents)		
Tokens per document	4682.40	574.59
Characters per document	21389.12	2334.69

Table 1: Statistics about the datasets we used

AV approach is generalizable to multiple domains such as fanfictions, Reddit comments, and Tweets, can make predictions both in-platform and cross-platform, and make fairly accurate predictions even on smaller datasets.

Authorship Verification Approach

We will start by describing our AV approach, introduced at the PAN workshops (Weerasinghe and Greenstadt 2020; Weerasinghe, Singh, and Greenstadt 2021). This approach assumes that the verification task is to predict, given **a pair of documents**, if the same person wrote them . We will refer to this approach as the **single-document-pair** approach (**single-pair** for short). Later, we extend this approach to a **multi-document** approach where we have a **pair of document sets** which allows us to extract features from multiple documents and run multiple predictions. We also evaluate how the document size and content similarity (as opposed to the writing style) affects our models. We also perform a feature analysis to assess any biases of our models, and uncover interesting linguistic features that differentiate same-author documents from different-author ones. The source code for our experiments and the trained models are available at: https://github.com/janithnw/authorship_verification.

Data Sets

We will be primarily using two datasets in our work. Summary statistics about these datasets are shown in Table 1.

PAN: This dataset was used at the PAN 2020 and 2021 AV shared tasks. The dataset provided for this task was compiled by Bischoff et al. (2020) and contains English documents from fanfiction.net. Each record in the dataset consists of two documents which may or may not be written by the same person. The ground truth specifies the author identifiers and the label indicating if the two documents were written by the same person. This is a roughly balanced dataset with 275,486 document pairs in which 54% of the records are same-author document pairs and the rest are different-author document pairs.

Reddit: This dataset was created using Reddit comments. Our goal was to create a dataset in which we can treat comments made by one user on two separate subreddits as two separate documents written by the same author. Our assumption was that the content on two subreddits would have enough topical variation so that our model would learn style-related similarities between two documents. This assumption is not perfect because Reddit users may post on similar topics even if it is on two subreddits. We collected this dataset from the public Reddit comments using Reddit data dumps made available through pushshift[1]. We downloaded the comment dumps for the time period of October 2019 to December 2019. Then we selected users with 200 to 5,000 comments. The upper limit helps to reduce the number of spam and bot accounts that post multiple of comments in a short period of time. We set this threshold after looking at a sample set of users that has a large number of comments. We also excluded widely accepted bot accounts[2] and users who post highly repetitive content. To determine if a user's content is repetitive, we computed the compression-ratio of their content by taking the ratio of the lengths between a gzip-compressed version of the text and the original text. If this ratio is less than $.25$, indicating that their content can be compressed very efficiently due to it's repetitiveness, we removed that user from the dataset. Out of these users, we further selected users that have posted at least 40 comments on at-least two subreddits. Our final dataset contains $46,465$ Reddit users, with each user having more than two *documents* where each document is a concatenation of all the comments a user made on one subreddit.

Preprocessing and Feature Extraction

We will briefly discuss the preprocessing and feature extraction process that we use on all of our models. During the preprocessing stage, the NLTK Treebank Word Tokenizer tokenizes the documents and NLTK's default Perceptron-based POS-tagger computes the part of speech tag for each token. Then a pre-trained POS tag chunker (based on the example given in the NLTK book by Bird (2009)) is used to group POS tags into verb phrases (VP), noun phrases (NP), and prepositional phrases (PP).

The set of features we used are commonly used in most previous stylometry work (Stamatatos 2009; Abbasi and Chen 2008). They include TF-IDF values of character tri-grams, POS tag tri-grams, punctuation marks, function words and the ratio between all POS tag pairs. Similar to other prior work (Hirst and Feiguina 2007; Luyckx and Daelemans 2005), the syntactic structure of sentences is captured using POS Tag chunks. This includes TF-IDF values of POS tag chunk tri-grams and features that encode how each POS tag chunk is constructed. To capture stylistic information about word order while also preventing topic related biases, we replaced all words (except function words) with their part-of-speech tag and computed the TF-IDF values of the tri-grams from this modified text. Similar methods of *text distortion* have been used successfully in previous studies (Bergsma, Post, and Yarowsky 2012; Stamatatos 2017). We also included features that encode the fraction of commonly misspelled English words, common typos, com-

[1] https://files.pushshift.io/

[2] https://www.reddit.com/r/autowikibot/wiki/redditbots

mon errors with determiners, British spelling of words, and popular online abbreviations, vocabulary richness, and word length.

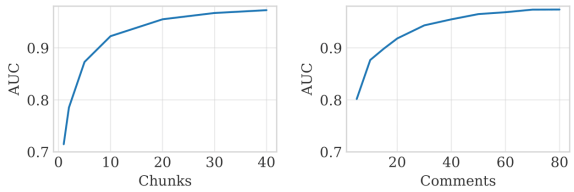
Classifier

Following our earlier work, we used a Logistic Regression classifier to make predictions. Since some of the datasets are large and cannot be trained in-memory, we train the classifier incrementally with a stochastic gradient descent (SGD) algorithm with a logarithmic loss function. After the feature extraction process we scaled the feature matrix to have a zero mean and unit variance. To train and test our classifier, we use the pair of documents (D_A and D_B, and their feature vectors X_A and X_B), and the label Y indicating if the document pair was written by the same author ($Y = 1$) or not ($Y = 0$). We then take the absolute difference of feature vectors ($X = |X_A - X_B|$) to construct a feature vector representing the document pair. Since the SGD algorithm requires a scaled feature vector for optimal performance, we scaled the final feature vector (different from our initial scaling) before passing it to the classifier.

Training and testing: Each record in the PAN dataset is a document pair with 54% of the pairs from the same author and the rest from different-author pairs. We sampled 70% of this dataset to be used as the training set and the rest as the test set. When splitting the dataset, we ensured that authors in the training set are not included in the test set.

Similarly, we split our Reddit dataset into two sets with a training set with 70% of the users and a test set with 30% of the users, and made sure that the test set does not include any user that was included in the training set, which makes this an open-world setting. To construct the same-user pairs, for each user in our dataset, we consider all the unique pairwise comparisons across the documents as same-author training records. If a user has n documents, we will generate $n(n-1)/2$ same-author document pairs. We included two types of different-author pairs in our dataset. These are document pairs from different users that talk about the same topic and about different topics. This allows our classifier to learn features that are more relevant to writing style and avoid topic-related biases. To generate different-author-same-topic training samples, we match each user document with a random document belonging to the same subreddit from another user. Similarly, to generate different-author-different-topic training samples, we match each user document with a random document from another user from a different subreddit. This process generates a large number of different-author document pairs. We then sample a subset of these document pairs in order to create a balanced dataset with 50% same-author pairs, 15% different-author-different-topic pairs and 35% different-author-same-topic pairs. We sampled more different-author-same-topic pairs to make the dataset more challenging for the learning algorithm and avoid topic related confounders. Our primary test set contains a similar mixture of records.

We trained two models for 100 iterations on the two datasets using the SGD algorithm. We evaluated the performance of each model by measuring the AUC (Area under the ROC curve) for the test set predictions. The results were remarkably consistent. The PAN model achieved an AUC of **0.970** and the Reddit model achieved an AUC of **0.981**.



(a) PAN (b) Reddit

Figure 1: Classifier performance (AUC) at varying document lengths.

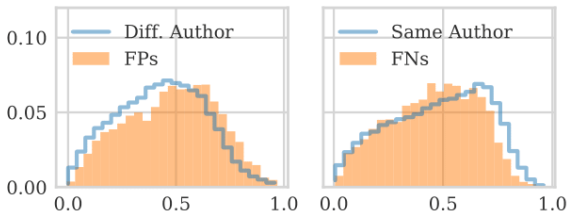
As we discussed before, prior to applying this AV approach in real-world scenarios, it is important to understand how it would perform under challenging scenarios. Therefore, in the next sub-sections, we will evaluate how this approach works on varying document lengths, analyze the effect that content similarity has towards model accuracy, perform a feature analysis, and evaluate the model on an imbalanced test set.

Varying Document Lengths

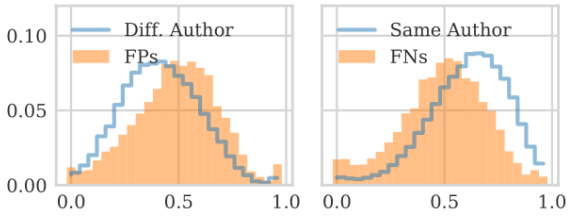
One of the challenges in applying AV to social media is the smaller length of social media content such as Tweets and comments. We evaluated models with varying document lengths to determine how far we can push our models without compromising the performance. We split the documents from the PAN dataset into chunks with each chunk on average containing 125 tokens. For the Reddit dataset, we varied the document size by varying the number of comments included in each document. On average each comment consists of 31 tokens. All steps of the pipeline, except the preprocessing step, are re-run for each data point. The number of training and test samples are held constant for each data point. As expected the performance of our model continues to increase when the size of documents increases (Figure 1). However, the marginal increase in AUC seems to decrease as we increase the document length. The PAN model performs fairly accurately with an AUC of 0.92 with documents with 10 chunks (1,250 tokens), and our Reddit model has an AUC of 0.95 on with just 40 comments (1,240 tokens).

Content Similarity and Accuracy

An ideal AV approach should only consider an author's writing style. However, disentangling the writing style from the content of documents is difficult. A model that includes features such as character or word n-grams could learn content-specific features about an author that may not generalize to other authors who write about different topics. Models could incorrectly learn that similarity in content is an important feature in same-author documents. However, in certain scenarios, the similarity in content could play an important role because it is more likely for an author to write about similar topics than about vastly different topics. The solution would



(a) PAN Topic Similarity



(b) Reddit Topic Similarity

Figure 2: Topic similarity across the document pairs. Blue lines show the histograms of the topic similarity in the ground truth. The filled orange histograms show the topic similarity for misclassifications: false positives (FPs) and false negatives (FNs). Note that the histograms are normalized and the actual misclassifications are a very small fraction of the data.

be to create models that are not too biased towards content similarity. One way to evaluate a model's bias is to measure the content similarity in misclassifications.

Approach: In this section, following Kestemont et al.'s (2020) approach, we measure how similarity in content affects our models. We trained topic models on our corpora using Non-Negative Matrix Factorization (NMF) and then measured the topic representations' similarity in our document pairs. Specifically, we trained two NMF models with 150 topics using a randomly sampled subset of our two datasets. We trained each model on a TF-IDF-normalized bag-of-words representation that only included nouns, adjectives, and verbs. Similar to Kestemont et al., the models were trained on 300-token smaller documents that were created by splitting the larger documents into shorter ones.

We can then measure the topic similarity between two given documents by splitting them into 300-token chunks, running the NMF model to obtain the topic representation for each chunk, averaging the representation across the chunks to obtain the topic representation for the entire document, normalizing it, and then computing the cosine similarity between the two topic representations. Following this approach, we measured the topic similarity between all document pairs in our test datasets. We then compared these topic similarities with the predictions made in the previous section.

Results: We find that, while our model is mostly topic agnostic, some of the misclassifications may be caused by content-similarity-related biases. We see that the topic-similarity between the same-author and different-author pairs (histograms represented by the blue lines in Figure 2) for the PAN dataset is mostly identical. To identify if there is a significant bias caused by topic-similarity, we can measure the difference between the topic-similarity distribution of the ground truth and the misclassifications. We can quantify this by computing the Cohen's d value between the distributions, by taking the difference between the two means of the distributions and dividing it by the pooled standard deviation. Generally a Cohen's d value between 0.2 and 0.4 is considered to be a small effect, 0.4 and 0.8 is considered a medium effect, and a value greater than 0.8 is considered to be a large effect. When considering the PAN misclassifications, the topic similarities of false positives and true positives are very similar (Cohen's d of 0.15), which suggests that there is no topic-related biases in the false positive predictions. Based on the right plot in Figure 2 (a), some same-author pairs that were predicted as different-authors (false-negatives) have a slightly lower topic similarity than the ground truth (Cohen's d value is 0.20) which suggests that there is a small topic-related bias.

As for our Reddit dataset, Figure 2 (b) shows that our different-author documents have a low similarity when compared to the same-author document pairs (represented by the blue lines on the two histograms). Upon further analysis, we found that that same-author document pairs from different subreddits still had a higher topic similarity than different-author document pairs from the same subreddit. We believe that this is because, users usually participate in different subreddits about similar topics (such as r/bitcoin, r/btc) and that even on the same subreddit there could be a variety of related content. This further highlights the difficulty in disentangling ones writing style from their content. The different-author pairs that were falsely predicted as same-author documents (false positives) has a slightly higher topic similarity than the rest of the different-author document pairs (Cohen's d of 0.29). The false-negative predictions had a lower topic similarity than actual same-author document pairs (Cohen's d of 0.69). This shows that our model has a moderate bias for false-negatives. A future improvement we can make is to create a topic-similarity-balanced dataset, as we do in our Twitter-Reddit model. This can be done by computing the topic similarity between all possible training pairs and sample them such that the topic similarity distribution is uniform and similar between same-author and different-author pairs. However, doing so naively by first creating all possible pairs can be prohibitively expensive on a large dataset.

Feature Analysis

Next we performed a feature analysis of our models to find out which features were the most important in making predictions. We measured the importance of each feature using Shapley Additive Explanations, or SHAP values (Lundberg and Lee 2017). The SHAP value for a given feature value on a prediction instance shows the degree to which that feature influenced the final prediction. For linear models, the SHAP value can be computed using the deviation of each feature from the mean feature value, scaled by the feature weight.

Avg SHAP value		Feature Name	
PAN:	0.112	Char n-gram	[,"]
	0.105	Char n-gram	[..]
	0.103	Masked stopwords	[, and]
	0.100	Char n-gram	[beg]
	0.097	Masked stopwords	[. but]
	0.095	Char n-gram	[,"]
	0.087	POS tag chunk n-gram	[, NNP .]
	0.084	Vocab richness Brunet's W	
	0.083	Masked stopwords	[, but]
	0.080	Freq. stopwords	[towards]
Reddit:	0.237	Special Character	[']
	0.235	Char n-gram	[']
	0.231	Char n-gram	[.]
	0.185	Char n-gram	[,]
	0.160	Special Character	[*]
	0.143	POS tag chunk n-gram	[, NP VP]
	0.134	Char n-gram	[']
	0.132	Masked stopwords	[, but]
	0.130	Char n-gram	[..]
	0.124	Special Character	[,]

Table 2: Top-10 features for each dataset

The SHAP value summary for a model is obtained by taking the average of absolute SHAP values across a test set.

Table 2 shows the top-10 features with the highest average SHAP values. We inspected the top-50 features in our experiments and no obvious topic-specific features were included as top features. The top features comprise of function words, punctuation marks, and POS tag related features. We did observe several interesting linguistic patterns. For example, the character n-gram feature [,"] is an important feature and shows that same-author documents tend to have a high similarity for this feature value. A possible explanation is that most American style guides such as MLA and AP suggest placing commas inside the closing quotation marks whereas in British English, the comma can be placed either inside or outside a quotation. The features [, and] were among the top 15 features for both PAN and Reddit datasets, and suggest that authors tend to use (or not use) the Oxford comma consistently. Similarly the feature [. but] may be an indication that authors either prefer of do not prefer starting sentences with the word 'But', which is a debated topic among English writers. Apart from these features, features that encode punctuation marks, ellipses, different POS tag combinations, and sentence structures are among the important features identified by our analysis. Note that these results show several features that are essentially the same. This is because the same entry is computed using two different feature sets. For example, the character ['] is captured in both the special character and character n-gram feature sets. The classifier coefficient and the SHAP value for these two are slightly different due to the different normalization settings used in the different feature sets. However, these similar features are highly correlated. This correlation would not effect the predictions of the classifier, but would have an impact if we were to use the classifier coefficients as a measure for feature importance. We opted to use SHAP values to avoid this problem.

Multi-document Predictions

In this section we expand our initial approach to scenarios where we have multiple documents for an author (or splitting a large document to several smaller documents). We investigate whether it is better to concatenate the documents into a single document, or to treat them as separate documents and aggregate classifier prediction scores. We explore how the results change based on the amount of training data. We show that multiplying predictions from the single-pair model and the multi-document model improves performance.

The output of our classifier, given two documents, can be considered as a measure of *stylometric similarity* between the documents. Let us assume we have two authors A and B, each with a **set** of documents, D_A and D_B where $|D_A| = n$ and $|D_B| = m$. We can run multiple predictions that will reflect the stylometric similarity between document sets in the following manner:

- Intra-author similarity of A, denoted as $intra(A)$: We can run the model on all possible (non-duplicate) pairs of documents from D_A resulting in $n(n-1)/2$ predictions.
- Intra-author similarity of B, denoted as $intra(B)$ where $|intra(B)| = m(m-1)/2$.
- Inter-author similarity, denoted as $inter(A, B)$: We can run the model between all the pairs of documents between D_A and D_B. This would result in $m * n$ predictions.

We can then use the aggregate classifier scores to gain insight about the authors. For example, if the mean score for intra-author predictions for author A (denoted by $mean(intra(A))$) is high, this suggests that our model is able to accurately predict that documents in D_A have a similar writing style. On the other hand, a lower mean suggests that either the performance of our model is poor for author A or that author A has a very inconsistent writing style. Similarly a high inter-author similarity mean (denoted by $mean(inter(A, B))$), suggests that the two authors A and B has a similar writing style.

Approach: We experimented on multi-document predictions on both our PAN and Reddit datasets under varying settings. First, we assess the impact of having less training data. For Reddit data, we test this by sampling 50% of the users and limiting the number of comments for each subreddit by each user to 40, significantly reducing the amount of content we have per user. For the PAN dataset, we sampled 25% of the dataset and limited content to a maximum of 15 chunks (The chunk size is the same as in our previous experiments, leading to a average document length of 1,875 tokens).

We also wanted to identify the best approach to aggregate user content. Would having a large number of small documents, a small number of large documents, or one large document yield the best performance? For these experiments, we kept the amount of content the same and varied the size of each document. For the Reddit experiments, we

formed smaller documents by concatenating 10 comments and larger documents by concatenating 20 comments. For PAN experiments, we formed smaller documents by concatenating 5 chunks (averaging to 625 tokens) and larger documents by concatenating 10 chunks (averaging to 1250 tokens). We decided on these document sizes based two criteria: the need to have a meaningful number of documents in each document set, especially for the limited data experiments and the need for each document to have enough content to so that the models would have a reasonably high performance based on our results from our previous document size related experiments (Figure 1). We also applied our single-pair models to the test set.

During the training phase, for each document set pair, we are able to generate multiple training records by pairing up all the (now smaller) documents across the two document sets. For example, given two authors, A with m documents, and B with n, we can generate $m * n$ training records. We then use the same approach as before to train new models by scaling the feature vectors, taking the absolute difference between each vector in a training record, and scaling the resulting difference vector. We repeat this process for all the different training set size and document length settings we discussed above.

During the testing phase, for each document set pair, we used the model to make intra-author and inter-author predictions. We experimented with the following methods to combine the intra-author and inter-author classifier scores to arrive at an aggregate classifier score.

- **Inter-author Mean -** $mean(inter(A, B))$ is computed by taking the average of the classifier scores of all $m * n$ document pairs across the two authors.
- **Intra-Inter Author Similarity** is computed as $1-|mean(intra(A), intra(B))-mean(inter(A, B))|$ where $mean(intra(A), intra(B))$ is the pooled mean of all the intra-author classifier scores. The intuition behind this measure is that, we hypothesize that if the two authors are in fact the same person, their inter-author similarity and intra-author similarity would have a very similar values.
- **Intra-Inter Author Standardized Similarity** is the intra-inter author similarity, divided by the pooled standard deviation of all the classifier scores.

Results and Discussion: In general, our results (Table 3) show that the multi-document AV approach is more effective than the single-pair approach. We believe that this is because, unlike in the single-pair approach, by splitting the content into multiple smaller documents, the multi-document approach classifier is exposed to orders of magnitude more training records which results in a better performing classifier. As seen in Figure 1, the marginal increase in classifier performance as the document size increases becomes small. We see that splitting larger documents and creating more training records leads to better results, especially in scenarios where there is limited training records. We also see that combing both the single-pair classifier score and the multi-document classifier score generally gives the best performance, especially in the Reddit dataset. While these differences are small in absolute terms, they are improving an already strong model. Furthermore, the differences become larger in more challenging scenarios such as Reddit-Twitter AV (Table 5).

Data Imbalance: Similar to most previous studies on AV, up to this point, we have only tested our model on a balanced dataset, in which the amount of same-author and different-author documents are the same. However, when applying this model to abuse detection scenarios and in scenarios where this model is being applied to all user pairs, the amount of same-author pairs would be significantly smaller than different-author pairs. To evaluate how our Reddit model would perform on imbalanced datasets, we tested it on test sets with varying number of same-author pairs. We computed the predictions using the combination of single-pair and the 20-comment multi-document models that were trained on the full training dataset which gave the best performance previously (Table 3). We kept the total number of test records fixed (at 79,000) and varied the fraction of positive (same-author) pairs. We then measured the area under the ROC-curve, the average precision (same as the area under the precision-recall curve), the recall when the model's operating point is set such that the precision is 0.9 (R@P90), and the precision and recall values when the operating point is set to 0.5 (Table 4).

Our model performs quite well on test sets with a moderate class imbalance (with 5% - 25% same-author pairs), with high average precision values and a good balance between precision and recall. In most abuse detection scenarios, we would select a machine learning model's operating point based on constraints such as cost of false positives (usually incurred when a false-positive decision has to be reviewed by a human), and how many positives we are willing to let go undetected. Often, this operating point is set to a value that gives us a reasonable recall under a fixed "false-positive budget". We included the R@P90 measure to capture this idea. For example, when 5% of account pairs are same-author pairs, we can expect our model to perform at 90% precision while detecting 42% of the same-account pairs. We find that scenarios with even higher class imbalances (1% to 0.5% same-author pairs) are challenging to our model. In these scenarios, we will have to set the model operating point to a very high value to maintain a high precision while sacrificing recall. While the average precision values of our model are low for high-class-imbalance settings, they are still orders of magnitude better than a random classifier. We also tried two approaches that are widely suggested in literature (Leevy et al. 2018; Ali, Shamsuddin, and Ralescu 2013) as fixes to the class imbalance problem. We tried altering the class balance in the training set and changing the class weights assigned to our logistic regression classifier. Neither of these approaches resulted in a significant increase of the classifier performance.

These results help us identify the scenarios in which our model can be applied reliably. For example, applying our model across all the Reddit account pairs would not be ideal due to the extreme class imbalance in such scenarios. Instead, we believe that this model can be applied in settings

Aggregation Method	Less Training Data Smaller Docs	Less Training Data Larger Docs	More Training Data Smaller Docs	More Training Data Larger Docs
Reddit Data:				
Inter-Author mean	0.937	**0.954**	0.943	0.958
Intra-Inter Author Sim	0.945	0.943	0.958	0.960
Intra-Inter Author Std Sim	0.949	0.938	0.960	0.961
Single-pair approach	0.942		0.981	
Single-pair × Intra-Inter Author Sim	**0.951**	0.953	**0.983**	**0.984**
PAN Data:				
Inter-Author mean	0.952	0.951	0.918	0.900
Intra-Inter Author Sim	0.949	0.952	0.924	0.918
Intra-Inter Author Std Sim	**0.955**	**0.953**	0.926	**0.970**
Single-pair approach	0.950		**0.970**	
Single-pair × Intra-Inter Author Sim	0.951	**0.953**	**0.970**	**0.970**

Table 3: Results, measured using the area under the ROC-curve (AUC), from the multi-document and single-pair AV approaches under varying training data and document size settings.

Pos. Frac.	AUC	Avg. Prc.	R@P90	Prc.	Rcl.
0.500	0.984	0.982	0.973	0.947	0.933
0.250	0.984	0.954	0.890	0.857	0.932
0.100	0.984	0.890	0.675	0.667	0.933
0.050	0.984	0.815	0.412	0.487	0.932
0.010	0.984	0.541	0.011	0.155	0.934
0.005	0.984	0.409	0.001	0.083	0.935

Table 4: The AUC, average precision, recall when precision is 90% (R@P90), and precision and recall when classifier operating point is set to 0.5 under varying class imbalance levels

where we have reason to believe account pairs are suspicious. We can use other clues such as network, timing, and communication patterns to narrow down the pool of account pairs. For example, it is unlikely that two account pairs that participate in two completely separate threads to be sock-puppets.

Cross Platform Account Linking

Another use of AV is to link accounts across different platforms. Such verification models could become useful in identifying spread of misinformation that propagate from fringe platforms to more mainstream ones. In this section we will present the approach we took to train a verification model that can verify authorship across Reddit and Twitter. Since there are natural style differences between Twitter and Reddit, we opted to train a new model instead of using our previous model that was only trained with Reddit data.

Approach

To train a new model, we need a list of Reddit and Twitter user accounts that we know are managed by the same person. We collected this data from two sources: from Reddit's TwitterFollowers subreddit where users post their Twitter account to gain more followers, and from a search on Reddit where users mention a Twitter account with the phrase "my Twitter account". Simply having this phrase does not however guarantee that the Twitter account mentioned in the text belongs to the Reddit user. For example, users may mention "this is not my twitter account" or they may talk about their Twitter account but post a link to another account later in the post. Therefore, we used a semi-supervised approach to verify if the account pair belongs to the same person. We discarded any Reddit post or comment that had the phrase "not my twitter" from our list. We also automatically included any account pair that had similar usernames. We considered two usernames to be similar if the ratio between the average length of the two usernames and their Levenshtein edit-distance is less that 0.01. We decided on this threshold by observing this ratio for an annotated set of username pairs and ensured only the highly similar account pairs will be selected. We then manually annotated the rest of the account pairs. We were able to collect 2,218 user account pairs through this process. We then collected all the Tweets and Reddit comments for these account pairs and filtered out pairs that does not have more than 100 Tweets and 100 comments. We were left with 1,015 account pairs after this filtering. Then, as a preprocessing step, we removed all the hashtags, Twitter user mentions, and URLs from the texts. We also removed all the non-English text as identified by FastText's language predictor (Joulin et al. 2016). Then we split the account pairs in to a training set of 680 and a test set of 335 account pairs. Note that this training set is approximately 4% of the size of our Reddit data set.

In our previous experiments, we were able to minimize the biases caused by content similarity by ensuring we compare authors across different subreddits. We wanted our Reddit-Twitter model to be topic agnostic as well. As a first step, we measured the similarity of content across the accounts using the same process we used before. The topic-similarity distribution was mostly uniform. Our initial attempt to create different-author account pairs by randomly assigning a Twitter account to each Reddit account caused the different-author pairs to have a very low topic similarity. Therefore we decide to match the topic similarity distributions between the same-author and different-author classes. To do this we computed the topic similarity across all the

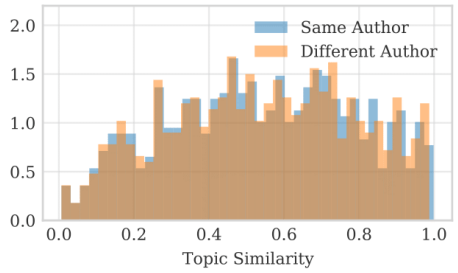


Figure 3: Topic Similarity of same-author accounts and similarity-matched different-author accounts.

	Balanced		Imbalanced	
Aggregation Method	**AUC**	**AP**	**AUC**	**AP**
Inter-Author mean	**0.907**	**0.907**	**0.859**	0.073
Intra-Inter Auth Sim	0.881	0.834	0.855	0.066
Intra-Inter Auth Std	0.876	0.885	0.842	0.085
Single-pair approach	0.837	0.805	0.845	0.084
Single-pair × I-I Sim	0.877	0.900	0.856	**0.129**

Table 5: Cross platform account linking results. AUC = Area under the ROC curve, AP = Average precision

possible different-user account pairs and used a linear sum assignment algorithm to find the optimal assignments. Here we treated the cost as the difference in topic similarity between a same-author and a different-author account pair. Figure 3 shows histograms of the topic similarity values for both same-author and distribution-matched different-author account pairs. Since both our positive and negative training samples follow a similar topic similarity distribution we believe that the final model would not have undesirable biases due to content similarity.

Similar to the previous experiments, we trained two AV models, a single-pair version and a multi-document version in which each account's content split in to multiple chunks. Since the lengths of Tweets and Reddit comments can be different, we performed the chunking by splitting the content by new lines, and grouping them into chunks so that each chunk would roughly have 5,000 characters (1,045 tokens). We trained both models on the topic-similarity matched training dataset. We evaluated the performance of each model and the aggregation strategies that we described before. We also evaluated the models on an imbalanced test dataset in which only 1% of the records were positive (same-author) pairs.

Results and Discussion

The best performance was achieved with the multi-document approach using inter-author classifier mean (Table 5). Unlike in our previous experiments the single-pair approach produced poorer results. We believe this is due to the smaller dataset size. This also shows that in scenarios with less training data, the multi-document approach is more suitable since we create more training records by splitting the content into multiple documents. In these experiments, the classifier for the single-pair approach was trained on 1,360 training records (680 same-author pairs and 680 different-author pairs), whereas the multi-doc classifier was trained on 855,403 (albeit smaller) document pairs. As with our previous experiments, our model performed poorly on the highly imbalanced test set. These results suggests that using this model to make predictions across a large numbers of Twitter and Reddit users might not be reliable. As discussed before, we believe that this model will be useful to link accounts from an already narrowed down list of suspicious accounts where the prior probability of same-author account pairs is relatively high or when other aspects such as timing patterns too are incorporated into the model. Furthermore, these models were trained on a significantly smaller number of users when compared to the previous models. We believe that adding more training data will improve the performance significantly.

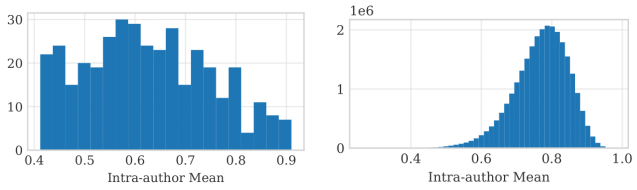
Identifying Reddit Sockpuppets

Another application of AV is to flag potential sockpuppet accounts on online platforms. Sockpuppet accounts are groups of accounts that are controlled by the same person. In this section we will apply our model to two settings where sockpuppet accounts may have been used:

- **Reddit 2017 Suspicious Accounts**: Reddit published a list of 944 accounts that were removed because they were suspected to have Russian Internet Research Agency origin. It is not known if these accounts were managed by the same person, a group of people, or even if the same account was managed by multiple people. However, the Russian IRA is known to have run "Troll Farms" to generate and disseminate content. Our goal was to apply our Reddit model to these accounts and see if the accounts have a consistent writing style and to find pairs or clusters of accounts that share similar writing styles.
- **WallStreetBets**: In January 2021, a short squeeze of the stock of GameStop (a video game retailer) took place. Most of the discussions around this happened on the WallStreetBets subreddit. While there is no strong evidence to suggest sockpuppets were used on this subreddit, sockpuppet accounts have been observed on Reddit (Cox 2018). We applied our model to users who commented on this subreddit during the beginning of the short squeeze to evaluate if any account pairs share similar writing styles.

Approach

In each of these cases we identified a set of users with enough content to apply our model. For the reddit suspicious accounts, we selected the 28 accounts where the total number of comments and submissions (Reddit posts) that the account is more than 100 for our analysis. For WallStreetBets we selected users who participated in the subreddit during the time period of November 1st, 2020 to January, 20th 2021, that have more posted more than 100 comments on the subreddit. We filtered out highly repetitive and bot accounts based on the compression ratio of the text similar to



(a) Reddit Suspicious Accounts (b) WallStreetBets Accounts

Figure 4: Intra-author classifier score means

the initial filtering we did on our Reddit dataset. After this filtering we were left with 7,318 accounts. We also collected comments from user accounts from Reddit's r/funny and r/gaming, two subreddits that usually receive a large number of comments, to use as baselines when comparing results. Similar to WallStreetBets, we removed repetitive and bot accounts and selected users that had more than 100 comments. We identified 1,558 users from r/funny and 1,110 users from r/gaming subreddits.

For predictions, we used the combination of single-pair and the 20-comment multi-pair models trained on the full Reddit dataset which resulted in the best performance in our experiments. Like before, we ran our models on each user pair and computed the inter-author and intra-author classifier scores and the classifier score of the single-pair approach. We used the Single-pair $\times$ Intra-Inter Author Similarity score as the aggregation method to arrive at a final score. We also use the mean intra-author classifier score as a measure of the consistency of writing style within a user account. The next step is to determine the threshold at which a given user pair is considered to be potential sockpuppets. Based on our results from the previous experiments, we decided to flag user pairs with an aggregated score greater than 0.995. Setting the threshold at such extreme value makes sure that we only flag account pairs that our model predicts with a very high degree of certainty. This also means that we might fail to identifying a large number of sockpuppet accounts. Since majority of the misclassifications due to content-similarity we discovered were false negatives, we believe that content-similarity-related biases in these predictions would be minimal.

Results

Figure 4 shows the intra-author means for the Reddit suspicious users and users from WallStreetBets subreddit. Unlike the Reddit accounts that we trained and tested our model on, the intra-author means of the Reddit suspicious accounts vary significantly, and in-fact have a lower value when compared to the users in our original dataset. This shows that the writing style across the same account is not consistent and suggests that the content on each account may be written by multiple individuals. Our results did show that four account pairs have an aggregate score higher that 0.995. However, only one of accounts had a high intra-author mean score which makes it difficult to conclusively predict if even the four account pairs are examples of sockpuppet accounts.

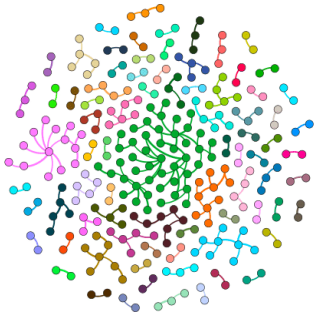


Figure 5: Clusters of accounts with similar writing styles from r/wallstreetbets. Nodes represent user accounts and an edges represent a highly similar writing style

On the other hand, the intra-author classifier score means from the users from WallStreetBets subreddit followed a similar pattern that of our original dataset. A vast majority of the user account comparisons did not show a shared writing style. We observed 205 account pairs having an aggregated score greater than 0.995 belonging to 283 user accounts (3.8% of the accounts we selected for analysis). In comparison, 9 account pairs belonging to 17 (1%) accounts from the r/funny subreddit had an aggregated score greater than 0.995 and none of the account pairs from r/gaming exceeded this threshold. Figure 5 shows a diagram of these accounts where the user accounts are represented by nodes and a shared writing style is represented by an edge between two nodes. Connected components of this graph are colored in different shades. We also used the Reddit API to find out if the 7,318 accounts we analyzed were deleted or suspended. 6.1% of the accounts were deleted and 6.4% of the accounts were suspended. However, we did not observe any significant difference between the suspension or deletion rate between the 283 accounts that we identified as having shared writing styles and the rest.

Here we applied our model to relatively active accounts (with at least 100 comments). However, an average Reddit user will usually have a smaller number of comments on a single subreddit. We opted to set this threshold because it gives our models, especially the our multi-document model, enough data to make very reliable predictions. However, our model can be applied to user accounts with fewer comments with a slight reduction in performance. Our experiments show that the single-pair model is able to achieve an AUC of 0.92 with 20 comments (Figure 1).

Limitations

In this work we tried to evaluate our models in settings that are as close to the real world as possible. We also assessed different biases of our models by applying them on varying document sizes, by ensuring topic-similarity-related biases are small, and by varying the class balance in our test sets. However, there are some scenarios that we were unable to approximate.

Throughout this paper, we assumed that the users do not try to change their writing style to evade detection. This as-

sumption might be true for simple sockpuppet account creators and abusers. If bad actors know that stylometry is used as a detection mechanism, a more sophisticated bad actor could change their writing style between accounts. While we did not test our approach in an adversarial setting, previous studies have shown that adversarial attacks are successful against authorship attribution (Brennan and Greenstadt 2009). However, such attempts to mask one's writing style can also be identified (Afroz, Brennan, and Greenstadt 2012).

Another deviation from real-world scenarios is that our same-author Reddit models were trained only on data from separate subreddits. In real sockpuppet and ban evasion settings the users would likely participate in the same subreddit. We did not include same-author same-subreddit documents in our training set to avoid topic-similarity biases. As seen in our content similarity experiments, for most users, even if we picked two different subreddits, we see that they have similar content. Therefore we believe that our models would be able to make accurate predictions in same-author same-subreddit settings.

Making reliable predictions with shorter documents is still a challenge in stylometry research. We tested our approach on varying document sizes, and showed that our models can achieve performances of 0.9-0.95 AUC with Reddit accounts with 20-40 comments. This level of performance might be adequate for analysis of flagged or suspicious accounts where the prior probability of a same-author account pair is high. However, for scenarios with high class imbalance, even higher performance levels are required to avoid false positives. These scenarios would require a larger amount of user content to make reasonably accurate predictions. While our approach works well for scenarios with moderate class imbalance, it performs poorly in high class imbalance settings. Therefore, applying this approach on a large user base will not be reliable. Our approach can be used in conjunction with other network and engagement features, as part of an abuse detection model, or be used in scenarios where a larger number of user accounts pairs share similar writing styles. Examples for such usecases could be analyzing already flagged accounts and digital forensics.

Conclusion

The spread of misinformation, disinformation, and trolling has contributed to the division in our current society and has caused real harm. Combating this requires research and help from a diverse set of disciplines. We believe that stylometry and authorship verification can play a role in this, both in identifying bad actors and aiding in investigations. We evaluated our previous AV approach to show that it adapts well to social media settings and expanded it to support scenarios when there are multiple documents per author. We also showed that the expanded method performs better than the single-pair approach especially in scenarios where training data is limited. We then showed that AV can be used to link social media accounts that share similar writing style, within the same platform and even across different platforms and showed that it can be useful in investigating suspicious accounts to gain insight about the account operates. Even applied in a conservative way, these methods can provide valuable insights about important cases of social media manipulation, suggesting that members of IRA troll farms shared accounts and that coordinated short squeezes may have involved sockpuppet accounts.

Acknowledgements

We thank the reviewers for their helpful comments and feedback. This work was supported by the U.S. NSF grant 1931005 and the McNulty Foundation.

References

Abbasi, A.; and Chen, H. 2008. Writeprints: A Stylometric Approach to Identity-level Identification and Similarity Detection in Cyberspace. *ACM Transactions on Information Systems*, 26.

Afroz, S.; Brennan, M.; and Greenstadt, R. 2012. Detecting Hoaxes, Frauds, and Deception in Writing Style Online. In *2012 IEEE Symposium on Security and Privacy*.

Ali, A.; Shamsuddin, S. M.; and Ralescu, A. L. 2013. Classification with class imbalance problem. *International Journal of Advances in Soft Computing and its Applications*, 5(3).

Almishari, M.; Kaafar, D.; Oguz, E.; and Tsudik, G. 2014. Stylometric Linkability of Tweets. In *Proceedings of the 13th Workshop on Privacy in the Electronic Society*, WPES '14. Association for Computing Machinery.

Bergsma, S.; Post, M.; and Yarowsky, D. 2012. Stylometric analysis of scientific articles. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Bevendorff, J.; Ghanem, B.; Giachanou, A.; Kestemont, M.; Manjavacas, E.; Markov, I.; Mayerl, M.; Potthast, M.; Rangel, F.; Rosso, P.; Specht, G.; Stamatatos, E.; Stein, B.; Wiegmann, M.; and Zangerle, E. 2020. Overview of PAN 2020: Authorship Verification, Celebrity Profiling, Profiling Fake News Spreaders on Twitter, and Style Change Detection. In *11th International Conference of the CLEF Association (CLEF 2020)*. Springer.

Bird, S.; Klein, E.; and Loper, E. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media, Inc.

Bischoff, S.; Deckers, N.; Schliebs, M.; Thies, B.; Hagen, M.; Stamatatos, E.; Stein, B.; and Potthast, M. 2020. The Importance of Suppressing Domain Style in Authorship Analysis. *CoRR*.

Blackwell, L.; Chen, T.; Schoenebeck, S. Y.; and Lampe, C. 2018. When Online Harassment Is Perceived as Justified. In *ICWSM*.

Boenninghoff, B.; Nickel, R.; and Kolossa, D. 2021. O2D2: Out-Of-Distribution Detector to Capture Undecidable Trials in Authorship Verification. In *CLEF 2021 Labs and Workshops, Notebook Papers*. CEUR-WS.org.

Brennan, M.; and Greenstadt, R. 2009. Practical Attacks Against Authorship Recognition Techniques. *Proceedings of the 21st Innovative Applications of Artificial Intelligence Conference, IAAI-09*.

Bu, Z.; Xia, Z.; and Wang, J. 2013. A sock puppet detection algorithm on virtual spaces. *Knowledge-Based Systems*, 37.

Burrows, J. 2002. 'Delta': a Measure of Stylistic Difference and a Guide to Likely Authorship. *Literary and Linguistic Computing*.

Cox, J. 2018. Inside a Reddit Sockpuppet Operation. https://www.vice.com/en/article/438v7j/inside-a-reddit-sockpuppet-operation-spam-upvote-shadowbanned. Accessed: 2021-09-10.

Daelemans, W.; Kestemont, M.; Manjavacas, E.; Potthast, M.; Rangel, F.; Rosso, P.; Specht, G.; Stamatatos, E.; Stein, B.; Tschuggnall, M.; Wiegmann, M.; and Zangerle, E. 2019. Overview of PAN 2019: Author Profiling, Celebrity Profiling, Cross-domain Authorship Attribution and Style Change Detection. In *10th International Conference of the CLEF Association (CLEF 2019)*.

El Manar El Bouanani, S.; and Kassou, I. 2013. Authorship Analysis Studies: A Survey. *International Journal of Computer Applications*, 86.

Halvani, O.; Graner, L.; and Regev, R. 2020a. Cross-Domain Authorship Verification Based on Topic Agnostic Features. In *CLEF (Working Notes)*.

Halvani, O.; Graner, L.; and Regev, R. 2020b. A Step Towards Interpretable Authorship Verification. *arXiv preprint arXiv:2006.12418*.

Halvani, O.; Winter, C.; and Graner, L. 2019. Assessing the Applicability of Authorship Verification Methods. In *Proceedings of the 14th International Conference on Availability, Reliability and Security*, ARES '19. Association for Computing Machinery.

Herrman, J. 2021. Everything's a Joke Until It's Not. https://www.nytimes.com/2021/01/29/style/gamestop-wallstreetbets-reddit.html. Accessed: 2021-09-10.

Hirst, G.; and Feiguina, O. 2007. Bigrams of syntactic labels for authorship discrimination of short texts. *Literary and Linguistic Computing*, 22(4).

Hoover, D. L. 2004. Delta prime? *Literary and Linguistic Computing*, 19(4).

Jiang, S.; Metzger, M. J.; Flanagin, A. J.; and Wilson, C. 2020. Modeling and Measuring Expressed (Dis)belief in (Mis)information. In *ICWSM*.

Johansson, F.; Kaati, L.; and Shrestha, A. 2013. Detecting multiple aliases in social media. In *2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013)*.

Joulin, A.; Grave, E.; Bojanowski, P.; and Mikolov, T. 2016. Bag of Tricks for Efficient Text Classification. *arXiv preprint arXiv:1607.01759*.

Kestemont, M.; Luyckx, K.; Daelemans, W.; and Crombez, T. 2012. Cross-genre authorship verification using unmasking. *English Studies*, 93(3).

Kestemont, M.; Manjavacas, E.; Markov, I.; Bevendorff, J.; Wiegmann, M.; Stamatatos, E.; Potthast, M.; and Stein, B. 2020. Overview of the Cross-Domain Authorship Verification Task at PAN 2020. In Cappellato, L.; Eickhoff, C.; Ferro, N.; and Névéol, A., eds., *CLEF 2020 Labs and Workshops, Notebook Papers*. CEUR-WS.org.

Kestemont, M.; Stamatatos, E.; Manjavacas, E.; Bevendorff, J.; Potthast, M.; and Stein, B. 2021. Overview of the Authorship Verification Task at PAN 2021. In *CLEF 2021 Labs and Workshops, Notebook Papers*. CEUR-WS.org.

Koppel, M.; and Schler, J. 2004. Authorship Verification as a One-Class Classification Problem. In *Proceedings of the Twenty-First International Conference on Machine Learning*, ICML '04. Association for Computing Machinery.

Koppel, M.; Schler, J.; and Argamon, S. 2009. Computational methods in authorship attribution. *Journal of the American Society for Information Science and Technology*, 60(1).

Kumar, S.; Cheng, J.; Leskovec, J.; and Subrahmanian, V. 2017. An Army of Me: Sockpuppets in Online Discussion Communities. In *Proceedings of the 26th International Conference on World Wide Web*.

Leevy, J. L.; Khoshgoftaar, T. M.; Bauder, R. A.; and Seliya, N. 2018. A survey on addressing high-class imbalance in big data. *Journal of Big Data*, 5(1).

Lundberg, S. M.; and Lee, S.-I. 2017. A Unified Approach to Interpreting Model Predictions. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc.

Luyckx, K.; and Daelemans, W. 2005. Shallow text analysis and machine learning for authorship attribtion. *LOT Occasional Series*.

Neal, T.; Sundararajan, K.; Fatima, A.; Yan, Y.; Xiang, Y.; and Woodard, D. 2017. Surveying Stylometry Techniques and Applications. *ACM Comput. Surv.*, 50(6).

Nirkhi, S.; Dharaskar, R.; and Thakare, V. 2016. Authorship Verification of Online Messages for Forensic Investigation. *Procedia Computer Science*, 78. 1st International Conference on Information Security and Privacy 2015.

Overdorf, R.; and Greenstadt, R. 2016. Blogs, Twitter Feeds, and Reddit Comments: Cross-domain Authorship Attribution. *Proceedings on Privacy Enhancing Technologies*, 2016(3).

Potha, N.; and Stamatatos, E. 2014. A profile-based method for authorship verification. In *Hellenic Conference on Artificial Intelligence*. Springer.

Reddit. 2017. Suspicious Accounts Wiki - reddit.com. https://www.reddit.com/wiki/suspiciousaccounts. Accessed: 2021-09-06.

Redmiles, E. M.; Bodford, J. E.; and Blackwell, L. 2019. "I Just Want to Feel Safe": A Diary Study of Safety Perceptions on Social Media. In *ICWSM*.

Solorio, T.; Hasan, R.; and Mizan, M. 2013. A case study of sockpuppet detection in wikipedia. In *Proceedings of the Workshop on Language Analysis in Social Media*.

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

Stamatatos, E. 2016. Authorship Verification: A Review of Recent Advances. *Research on computing science*, 123.

Stamatatos, E. 2017. Authorship attribution using text distortion. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*.

Starbird, K. 2017. Examining the Alternative Media Ecosystem Through the Production of Alternative Narratives of Mass Shooting Events on Twitter. In *ICWSM*.

Veenman, C. J.; and Li, Z. 2013. Authorship Verification with Compression Features. In *CLEF (working notes)*.

Vosoughi, S.; Zhou, H.; and Roy, D. 2015. Digital Stylometry: Linking Profiles Across Social Networks. In Liu, T.-Y.; Scollon, C. N.; and Zhu, W., eds., *Social Informatics*. Springer.

Weerasinghe, J.; and Greenstadt, R. 2020. Feature Vector Difference based Neural Network and Logistic Regression Models for Authorship Verification. In *CLEF 2020 Labs and Workshops, Notebook Papers*. CEUR-WS.org.

Weerasinghe, J.; Singh, R.; and Greenstadt, R. 2021. Feature vector difference based authorship verification for open world settings. In *CLEF 2021 Labs and Workshops, Notebook Papers*. CEUR-WS.org.

Weld, G. C.; Glenski, M.; and Althoff, T. 2021. Political Bias and Factualness in News Sharing Across more then 100, 000 Online Communities. In *ICWSM*.