A Case Study of Sockpuppet Detection in Wikipedia

Thamar Solorio and **Ragib Hasan** and **Mainul Mizan**
The University of Alabama at Birmingham
1300 University Blvd.
Birmingham, AL 35294, USA
{solorio,ragib,mainul}@cis.uab.edu

Abstract

This paper presents preliminary results of using authorship attribution methods for the detection of sockpuppeteering in Wikipedia. Sockpuppets are fake accounts created by malicious users to bypass Wikipedia's regulations. Our dataset is composed of the comments made by the editors on the talk pages. To overcome the limitations of the short lengths of these comments, we use an voting scheme to combine predictions made on individual user entries. We show that this approach is promising and that it can be a viable alternative to the current human process that Wikipedia uses to resolve suspected sockpuppet cases.

1 Introduction

Collaborative projects in social media have become very popular in recent years. A very successful example of this is Wikipedia, which has emerged as the world's largest crowd-sourced encyclopaedia. This type of decentralized collaborative processes are extremely vulnerable to vandalism and malicious behavior. Anyone can edit articles in Wikipedia and/or make comments in article discussion pages. Registration is not mandatory, but anyone can register an account in Wikipedia by providing only little information about themselves. This ease of creating an identity has led malicious users to create multiple identities and use them for various purposes, ranging from block evasion, false majority opinion claims, and vote stacking. This is an example of the multi aliasing problem known as "The Sybil Attack" (Douceur, 2002). Unfortunately, Wikipedia does not provide any facility to detect such multiple identities. The current process is carried out by humans, is very time consuming, and final resolution to cases of multiple identities is based on human intuition. A smart sockpuppet can therefore evade detection by using multiple IP addresses, modifying writing style, and changing behavior. Also, a malicious user can create sleeper accounts that perform benign edits from time to time, but are used for sockpuppetry when needed. Identifying such accounts as sockpuppets is not obvious as these accounts may have a long and diverse edit history.

Sockpuppets are a prevalent problem in Wikipedia, there were close to 2,700 unique suspected cases reported in 2012. In this paper, we present a small scale study of automated detection of sockpuppets based on machine learning. We approach this problem from the point of view of authorship attribution (AA), where the task consists of analyzing a written document to predict the true author. If we can successfully model the editors' unique writing style from their comments, then we can use this information to link the sockpuppet accounts to their corresponding puppeteer. We focus on the content from the talk pages since the articles edited on Wikipedia have a fixed and very uniform style. In contrast, we have observed that editors write in a more free-form style during discussions carried out on the talk pages. Our results show that a two-stage process for the task can achieve promising results.

The contributions of this study are as follows:

- We present encouraging preliminary results on using authorship attribution approaches for un-

Proceedings of the Workshop on Language in Social Media (LASM 2013), pages 59–68,
Atlanta, Georgia, June 13 2013.

covering real sockpuppet cases in Wikipedia. To the best of our knowledge, we are the first to tackle this problem.

- We identify novel features that have high discriminative power and are suitable for this task, where the input text is very short. These features can be helpful in other social media settings, as there are many shared characteristics across this genre.

The rest of the paper is organized as follows: in Section 2, we provide a detailed discussion on Wikipedia's editing environment and culture. In Section 3, we talk about authorship attribution and related work. Then in Section 4, we present our detailed approach. In Sections 5, 6, and 7, we discuss the data set, experimental setup, and results, respectively. Finally, we present an overall discussion and future directions in Sections 8 and 9.

2 Background

In Wikipedia, whenever a user acts in bad faith, vandalizes existing articles, or creates spurious articles, that user is banned from editing new content. The ban can last for some hours, to days, and in some cases it can be permanent. Sometimes, a banned user creates a new account to circumvent the ban, or edits Wikipedia without signing in.

These extra accounts or IP addresses, from which logged out edits are made, are called sockpuppets. The primary (oldest) account is called the sockpuppeteer. Whenever an editor is suspected to be a sockpuppet of another editor, a sockpuppet investigation case is filed against those accounts. Any editor can file a case, but the editor must provide supporting evidence as well. Typical evidence includes information about the editing actions related to those accounts, such as the articles, the topics, vandalism patterns, timing of account creation, timing of edits, and voting pattern in disagreements.

Sometime after the case is filed, an administrator will investigate the case. An administrator is an editor with privileges to make account management decisions, such as banning an editor. If the administrator is convinced that the suspect is a sockpuppet, he declares the verdict as confirmed. He also issues bans to the corresponding accounts and closes the case. If an administrator cannot reach a verdict on a case, he asks for a check user to intervene. Check users are higher privileged editors, who have access to private information regarding editors and edits, such as the IP address from which an editor has logged in. Other interested editors in the case, or the original editor who filed the case can also ask for a check user to intervene. The check user will review the evidence, as well as private information regarding the case, and will try to establish the connection between the sockpuppet and puppeteer. Then the check user will rule on the case. Finally, another administrator will look at the check user report and issue a final verdict. During the process, the accused editors, both the puppeteer and the sockpuppet, can submit evidence in their favor. But this additional evidence is not mandatory.

The current process to resolve suspected cases of sockpuppets has several disadvantages. We have already mentioned the first one. Because it is a manual process, it is time consuming and expensive. Perhaps a more serious weakness is the fact that relaying on IP addresses is not robust, as simple counter measures can fool the check users. An alternative to this process could be an automated framework that relies on the analysis of the comments to link editor accounts, as we propose in this paper.

3 Related Work

Modern approaches to AA typically follow a text classification framework where the classes are the set of candidate authors. Different machine learning algorithms have been used, including memory-based learners (Luyckx and Daelemans, 2008a; Luyckx and Daelemans, 2010), Support Vector Machines (Escalante et al., 2011), and Probabilistic Context Free Grammars (Raghavan et al., 2010).

Similarity-based approaches have also been successfully used for AA. In this setting, the training documents from the same author are concatenated into a single file to generate profiles from author-specific features. Then authorship predictions are based on similarity scores. (Keselj et al., 2003; Stamatatos, 2007; Koppel et al., 2011) are examples of successful examples of this approach.

Previous research has shown that low-level features, such as character n-grams are very powerful

discriminators of writing styles. Although, enriching the models with other types of features can boost accuracy. In particular, stylistic features (punctuation marks, use of emoticons, capitalization information), syntactic information (at the part-of-speech level and features derived from shallow parsing), and even semantic features (bag-of-words) have shown to be useful.

Because of the difficulties in finding data from real cases, most of the published work in AA evaluates the different methods on data collections that were gathered originally for other purposes. Examples of this include the Reuters Corpus (Lewis et al., 2004) that has been used for benchmarking different approaches to AA (Stamatatos, 2008; Plakias and Stamatatos, 2008; Escalante et al., 2011) and the datasets used in the 2011 and 2012 authorship identification competitions from the PAN Workshop series (Argamon and Juola, 2011; Juola, 2012). Other researchers have invested efforts in creating their own AA corpus by eliciting written samples from subjects participating in their studies (Luyckx and Daelemans, 2008b; Goldstein-Stewart et al., 2008), or crawling though online websites (Narayanan et al., 2012).

In contrast, in this paper we focus on data from Wikipedia, where there is a real need to identify if the comments submitted by what appear to be different users, belong to a sockpuppeteer. Data from real world scenarios like this make solving the AA problem an even more urgent and practical matter, but also impose additional challenges to what is already a difficult problem. First, the texts analyzed in the Wikipedia setting were generated by people with the actual intention of deceiving the administrators into believing they are indeed coming from different people. With few exceptions (Afroz et al., 2012; Juola and Vescovi, 2010), most of the approaches to AA have been evaluated with data where the authors were not making a conscious effort to deceive or disguise their own identities or writeprint. Since there has been very little research done on deception detection, it is not well understood how AA approaches need to be adapted for these situations, or what kinds of features must be included to cope with deceptive writing. However, we do assume this adds a complicating factor to the task, and previous research has shown considerable decreases in AA accuracy when deception is present (Brennan and Greenstadt, 2009). Second, the length of the documents is usually shorter for the Wikipedia comments than that of other collections used. Document length will clearly affect the prediction performance of AA approaches, as the shorter documents will contain less information to develop author writeprint models and to make an inference on attribution. As we will describe later, this prompted us to reframe our solution in order to circumvent this short document length issue. Lastly, the data available is limited, there is an average of 80 entries per user in the training set from the collection we gathered, and an average of 8 messages in the test set, and this as well limits the amount of evidence available to train author models. Moreover, the test cases have an average of 8 messages. This is a very small amount of texts to make the final prediction.

4 Approach

In our framework, each comment made by a user is considered a "document" and therefore, each comment represents an instance of the classification task. There are two steps in our method. In the first step, we gather predictions from the classifier on each comment. Then in the second step we take the predictions for each comment and combine them in a majority voting schema to assign final decisions to each account.

The two step process we just described helps us deal with the challenging length of the individual comments. It is also an intuitive approach, since what we need to determine is if the account belongs to the sockpuppeteer. The ruling is at the account-level, which is also consistent with the human process. In the case of a positive prediction by our system, we take as a **confidence** measure on the predictions the percentage of comments that were individually predicted as sockpuppet cases.

4.1 Feature Engineering

In this study, we have selected typical features of authorship attribution, as well as new features we collected from inspecting the data by hand. In total, we have 239 features that capture stylistic, grammatical, and formatting preferences of the authors. The features are described below.

Total number of characters: The goal of this feature is to model the author's behavior of writing

long wordy texts, or short comments.

Total number of sentences: We count the total number of sentences in the comments. While this feature is also trying to capture some preferences regarding the productivity of the author's comments, it can tell us more about the author's preference to organize the text in sentences. Some online users tend to write in long sentences and thus end up with a smaller number of sentences. To fragment the comments into sentences, we use the Lingua-EN-Sentence-0.25 from `www.cpan.org` (The Comprehensive Perl Archive Network). This off-the-shelf tool prevents abbreviations to be considered as sentence delimiters.

Total number of tokens: We define a token as any sequence of consecutive characters with no white spaces in between. Tokens can be words, numbers, numbers with letters, or with punctuation, such as *apple, 2345, 15th, and wow!!!*. For this feature we just count how many tokens are in the comment.

Words without vowels: Most English words have one or more vowels. The rate of words without vowels can also be a giveaway marker for some authors. Some words without vowels are *try, cry, fly, myth, gym,* and *hymn*.

Total alphabet count: This feature consists of the count of all the alphabetic characters used by the author in the text.

Total punctuation count: Some users use punctuation marks in very unique ways. For instance, semicolons and hyphens show noticeable differences in their use, some people avoid them completely, while others might use them in excess. Moreover, the use of commas is different in different parts of the world, and that too can help identify the author.

Two/three continuous punctuation count: Sequences of the same punctuation mark are often used to emphasize or to add emotion to the text, such as *wow!!!*, and *really??*. Signaling emotion in written text varies greatly for different authors. Not everyone displays emotions explicitly or feels comfortable expressing them in text. We believe this could also help link users to sockpuppet cases.

Total contraction count: Contractions are used for presenting combined words such as *don't*, *it's*, *I'm*, and *he's*. The contractions, or the spelled-outforms are both correct grammatically. Hence, the use of contraction is somewhat a personal writing style attribute. Although the use of contractions varies across different genres, in social media they are commonly used.

Parenthesis count: This is a typical authorship attribution feature that depicts the rate at which authors use parenthesis in their comments.

All caps letter word count: This is a feature where we counted the number of tokens having all upper case letters. They are either abbreviations, or words presented with emphasis. Some examples are *USA*, or "this is *NOT* correct".

Emoticons count: Emoticons are pictorial representations of feelings, especially facial expressions with parenthesis, punctuation marks, and letters. They typically express the author's mood. Some commonly used emoticons are :) or :-) for happy face, :(for sad face, ;) for winking, :D for grinning, <3 for love/heart, :O for being surprised, and :P for being cheeky/tongue sticking out.

Happy emoticons count: As one of the most widely used emoticons, happy face was counted as a specific feature. Both :) and :-) were counted towards this feature.

Sentence count without capital letter at the beginning: Some authors start sentences with numbers or small letters. This feature captures that writing style. An example can be "1953 was the year, ..." or, "big, bald, and brass - all applies to our man".

Quotation count: This is an authorship attribution feature where usage of quotation is counted as a feature. When quoting, not everyone uses the quotation punctuation and hence quotation marks count may help discriminate some writers from others.

Parts of speech (POS) tags frequency: We took a total of 36 parts of speech tags from the Penn Treebank POS (Marcus et al., 1993) tag set into consideration. We ignored all tags related to punctuation marks as we have other features capturing these characters.

Frequency of letters: We compute the frequency of each of the 26 English letters in the alphabet. The count is normalized by the total number of non-white characters in the comment. This contributed 26 features to the feature set.

Function words frequency: It has been widely acknowledged that the rate of function words is a good marker of authorship. We use a list of function words taken from the function words in (Zheng et al., 2006). This list contributed 150 features to the feature set.

All the features described above have been used in previous work on AA. Following are the features that we found by manually inspecting the Wikipedia data set. All the features involving frequency counts are normalized by the length of the comment.

Small "i" frequency: We found the use of small "i" in place of capital "I" to be common for some authors. Interestingly, authors who made this mistake repeated it quite often.

Full stop without white space frequency: Not using white space after full stop was found quite frequently, and authors repeated it regularly.

Question frequency: We found that some authors use question marks more frequently than others. This is an idiosyncratic feature as we found some authors abuse the use of question marks for sentences that do not require question marks, or use multiple question marks where one question mark would suffice.

Sentence with small letter frequency: Some authors do not start a sentence with the first letter capitalized. This behavior seemed to be homogeneous, meaning an author with this habit will do it almost always, and across all of its sockpuppet accounts.

Alpha, digit, uppercase, white space, and tab frequency: We found that the distribution of these special groups of characters varies from author to author. It captures formatting preferences of text such as the use of "one" and "zero" in place of "1" and "0", and uppercase letters for every word.

'A', and an error frequency: Error with usage of "a", and "an" was quite common. Many authors tend to use "a" in place of "an", and vice versa. We used a simple rate of all "a" in front of words starting with vowel, or "an" in front of words starting with consonant.

"he", and "she" frequency: Use of "he", or "she" is preferential to each author. We found that the use of "he", or "she" by any specific author for an indefinite subject is consistent across different comments.

5 Data

We collected our data from cases filed by real users suspecting sockpupeteering in the English Wikipedia. Our collection consists of comments made by the accused sockpuppet and the suspected puppeteer in various talk pages. All the information about sockpuppet cases is freely available, together with information about the verdict from the administrators. For the negative examples, we also collected comments made by other editors in the comment threads of the same talk pages. For each comment, we also collected the time when the comment was posted as an extra feature. We used this time data to investigate if non-authorship features can contribute to the performance of our model, and to compare the performance of stylistic features and external user account information.

Class	Total	Avg. Msg. Train	Avg. Mesg. Test
Sockpuppet	41	88.75	8.5
Non-sockpuppet	36	77.3	7.9

Table 1: Distribution of True/False sockpuppet cases in the experimental data set. We show the average number of messages in train and test partitions for both classes.

Our dataset has two types of cases: confirmed sockpuppet, and rejected sockpuppet. The confirmed cases are those where the administrators have made final decisions, and their verdict confirmed the case as a true sockpuppet case. Alternatively, for the rejected sockpuppet cases, the administrator's verdict exonerates the suspect of all accusations. The distribution of different cases is given in Table 1.

Of the cases we have collected, one of the notable puppeteers is "-Inanna-". This editor was active in Wikipedia for a considerable amount of time, from December 2005 to April 2006. He also has a number of sockpuppet investigation cases against him. Table 2 shows excerpts from comments made by this editor on the accounts confirmed as sockpuppet. We highlight in boldface the features that are more noticeable as similar patterns between the different user accounts.

An important aspect of our current evaluation framework is the preprocessing of the data. We "cleansed" the data by removing content that was not written by the editor. The challenge we face is that Wikipedia does not have a defined structure for comments. We can get the difference of each modification in the history of a comment thread. However, not all modifications are comments. Some can be reverts (changing content back to an old version), or updates. Additionally, if an editor replies to more than one part of a thread in response to multiple com-

Comment from the sockpuppeteer: -Inanna- Mine was original and i have worked on it more than 4 hours**.I** have changed it many times by opinions**.L**ast one was accepted by all the users(except for khokhoi)**.I** have never used sockpuppets**.P**lease dont care Khokhoi,Tombseye and Latinus.They are changing all the articles about Turks.The most important and famous people are on my picture.
Comment from the sockpuppet: Altau Hello**.I** am trying to correct uncited numbers in Battle of Sarikamis and Crimean War by resources but khoikhoi and tombseye always try to revert them**.C**ould you explain them there is no place for hatred and propagandas, please?
Comment from the others: Khoikhoi Actually, my version WAS the original image. Ask any other user. Inanna's image was uploaded later, and was snuck into the page by Inanna's sockpuppet before the page got protected. The image has been talked about, and people have rejected Inanna's image (see above).

Table 2: Sample excerpt from a single sockpuppet case. We show in boldface some of the stylistic features shared between the sockpuppeter and the sockpuppet.

System	P	R	F	A (%)
B-1	0.53	1	0.69	53.24
B-2	0.53	0.51	0.52	50.64
Our System	**0.68**	**0.75**	**0.72**	**68.83**

Table 3: Prediction performance for sockpuppet detection. Measures reported are Precision (P), Recall (R), F-measure (F), and Accuracy (A). B-1 is a simple baseline of the majority class and B-2 is a random baseline.

ments, or edits someone else's comments for any reason, there is no fixed structure to distinguish each action. Hence, though our initial data collector tool gathered a large volume of data, we could not use all of it as the preprocessing step was highly involved and required some manual intervention.

6 Experimental Setting

We used Weka (Witten and Frank, 2005) – a widely recognized free and open source data-mining tool, to perform the classification. For the purpose of this study, we chose Weka's implementation of Support Vector Machine (SVM) with default parameters.

To evaluate in a scenario similar to the real setting in Wikipedia, we process each sockpuppet case separately, we measure prediction performance, and then aggregate the results of each case. For example, we take data from a confirmed sockpuppet case and generate the training and test instances. The training data comes from the comments made by the suspected sockpuppeteer, while the test data comes from the comments contributed by the sockpuppet account(s). We include negative samples for these cases by collecting comments made on the same talk pages by editors not reported or suspected of sockpuppeteering. Similarly, to measure the false positive ratio of our approach, we performed experiments with confirmed non-sockpuppet editors that were also filed as potential sockpuppets in Wikipedia.

7 Results

The results of our experiments are shown in Table 3. For comparison purposes we show results of two simple baseline systems. **B-1** is the trivial classifier that predicts every case as sockpuppet (majority). **B-2** is the random baseline (coin toss). However as seen in the table, both baseline systems are outperformed by our system that reached an accuracy of 68%. B-1 reached an accuracy of 53% and B-2 of 50%.

For the miss-classified instances of confirmed sockpuppet cases, we went back to the original comment thread and the investigation pages to find out the sources of erroneous predictions for our system. We found investigation remarks for 4 cases. Of these 4 cases, 2 cases were tied on the predictions for the individual comments. We flip a coin in our system to break ties. From the other 2 cases, one has the neutral comment from administrators: "Possible", which indicates some level of uncertainty. The last one has comments that indicate a meat puppet. A meat puppet case involves two different real people

where one is acting under the influence of the other. A reasonable way of taking advantage of the current system is to use the confidence measure to make predictions of the cases where our system has the highest confidence, or higher than some threshold, and let the administrators handle those cases that are more difficult for an automated approach.

We have also conducted an experiment to rank our feature set with the goal of identifying informative features. We used information gain as the ranking metric. A snapshot of the top 30 contributing features according to information gain is given in Table 4. We can see from the ranking that some of the top-contributing features are idiosyncratic features. Such features are white space frequency, beginning of the sentence without capital letter, and no white space between sentences. We can also infer from Table 4 that function word features (My, me, its, that, the, I, some, be, have, and since), and part of speech tags (VBG-Verb:gerund or present participle, CD-Cardinal number, VBP-Verb:non-3rd person singular present, NNP-Singular proper noun, MD-Modal, and RB-Adverb) are among the most highly ranked features. Function words have been identified as highly discriminative features since the earliest work on authorship attribution.

Finally, we conducted experiments with two edit timing features for 49 cases. These two features are edit time of the day in a 24 hour clock, and edit day of the week. We were interested in exploring if adding these non-stylistic features could contribute to classification performance. To compare performance of these non-authorship attribution features, we conducted the same experiments without these features. The results are shown in Table 5. We can see that average confidence of the classification, as well as F-measure goes up with the timing features. These timing features are easy to extract automatically, therefore they should be included in an automated approach like the one we propose here.

Features
Whitespace frequency
Punctuation count
Alphabet count
Contraction count
Uppercase letter frequency
Total characters
Number of tokens
me
my
its
that
Beginning of the sentence without capital letter †
VBG-Verb:gerund or present participle
No white space between sentences †
the
Frequency of L
I
CD-Cardinal number
Frequency of F
VBP-Verb:non-3rd person singular present
Sentence start with small letter †
some
NNP-Singular proper noun
be
Total Sentences
MD-Modal
? mark frequency
have
since
RB-Adverb

Table 4: Ranking of the top 30 contributing features for the experimental data using information gain. Novel features from our experiment are denoted by †.

Features used	**Confidence**	**F-measure**
All + timing features	84.04%	0.72
All - timing features	78.78%	0.69

Table 5: Experimental result showing performance of the method with and without timing features for the problem of detecting sockpuppet cases. These results are on a subset of 49 cases.

8 Discussion

The experiments presented in the previous section are encouraging. They show that with a relatively small set of automatically generated features, a machine learning algorithm can identify, with a reasonable performance, the true cases of sockpuppets in Wikipedia.

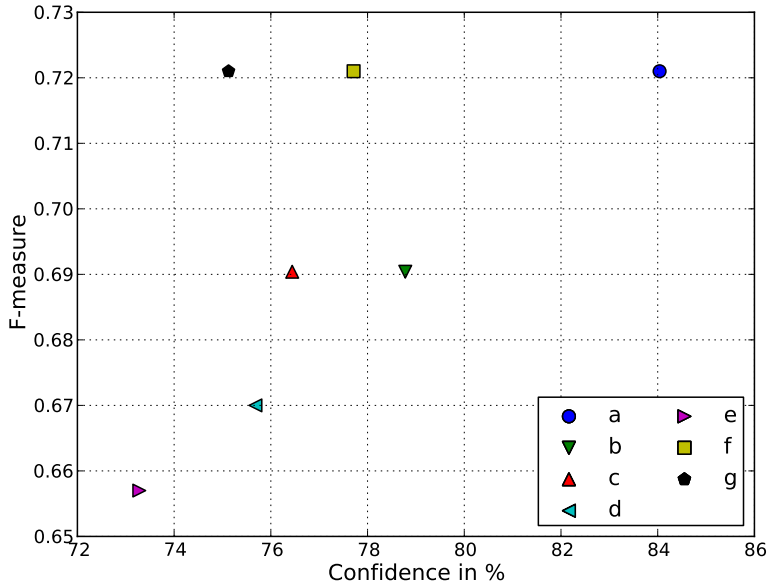

Figure 1: A plot of confidence in % for successful cases vs. F-measure for the system where we remove one feature group at a time. Here marker a) represents performance of the system with all the features. Markers b) timing features, c) part of speech tags, d) idiosyncratic features, e) function words, f) character frequencies, and g) AA features, represent performance of the system when the specified feature group is removed.

Since falsely accusing someone of using a sockpuppet could lead to serious credibility loss for users, we believe a system like ours could be used as a first pass in resolving the suspected sockpuppet cases, and bring into the loop the administrators for those cases where the certainty is not high.

To further investigate the contribution of different groups of features in our feature set, we conducted additional experiments where we remove one feature group at a time. Our goal is to see which feature group causes larger decreases in prediction performance when it is not used in the classification. We split our feature set into six groups, namely timing features, parts of speech tags, idiosyncratic features, function words, character frequencies, and authorship attribution features. In Figure 1, we show the result of the experiments. From the figure, we observe that function words are the most influential features as both confidence, and F-measure showed the largest drop when this group was excluded. The idiosyncratic features that we have included in the feature set showed the second largest decrease in prediction performance. Timing features, and part of speech tags have similar drops in F-measure but they showed a different degradation pattern on the confidence: part of speech tags caused the confidence to decrease by a larger margin than the timing features. Finally, character frequencies, and authorship attribution features did not affect F-measure much, but the confidence from the predictions did decrease considerably with AA features showing the second largest drop in confidence overall.

9 Conclusion and Future Directions

In this paper, we present a first attempt to develop an automated detection method of sockpuppets based solely on the publicly available comments from the suspected users. Sockpuppets have been a bane for Wikipedia as they are widely used by malicious users to subvert Wikipedia's editorial process and consensus. Our tool was inspired by recent work on the popular field of authorship attribution. It requires no additional administrative rights (e.g., the ability to view user IP addresses) and therefore can be used by regular users or administrators without check user rights. Our experimental evaluation with real sock-

puppet cases from the English Wikipedia shows that our tool is a promising solution to the problem.

We are currently working on extending this study and improving our results. Specific aspects we would like to improve include a more robust confidence measure and a completely automated implementation. We are aiming to test our system on all the cases filed in the history of the English Wikipedia. Later on, it would be ideal to have a system like this running in the background and pro-actively scanning all active editors in Wikipedia, instead of running in a user triggered mode. Another useful extension would be to include other languages, as English is only one of the many languages currently represented in Wikipedia.

Acknowledgements

This research was supported in part by ONR grant N00014-12-1-0217. The authors would like to thank the anonymous reviewers for their comments on a previous version of this paper.

References

S. Afroz, M. Brennan, and R. Greenstadt. 2012. Detecting hoaxes, frauds, and deception in writing style online. In *Proceedings of the 2012 IEEE Symposium on Security and Privacy (S&P)*, pages 461–475. IEEE, May.

Shlomo Argamon and Patrick Juola. 2011. Overview of the international authorship identification competition at PAN-2011. In *Proceedings of the PAN 2011 Lab Uncovering Plagiarism, Authorship, and Social Software Misuse, held in conjunction with the CLEF 2011 Conference on Multilingual and Multimodal Information Access Evaluation*, Amsterdam.

M. Brennan and R. Greenstadt. 2009. Practical attacks against authorship recognition techniques. In *Proceedings of the Twenty-First Innovative Applications of Artificial Intelligence Conference*.

John Douceur. 2002. The Sybil attack. In Peter Druschel, Frans Kaashoek, and Antony Rowstron, editors, *Peer-to-Peer Systems*, volume 2429 of *Lecture Notes in Computer Science*, pages 251–260. Springer Berlin / Heidelberg.

H. J. Escalante, T. Solorio, and M. Montes-y Gómez. 2011. Local histograms of character n-grams for authorship attribution. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 288–298, Portland, Oregon, USA, June. Association for Computational Linguistics.

Jade Goldstein-Stewart, Kerri A. Goodwin, Roberta Evans Sabin, and Ransom K. Winder. 2008. Creating and using a correlated corpus to glean communicative commonalities. In *Proceedings of LREC-2008, the Sixth International Language Resources and Evaluation Conference*.

P. Juola and D. Vescovi. 2010. Empirical evaluation of authorship obfuscation using JGAAP. In *Proceedings of the 3rd ACM workshop on Artificial Intelligence and Security*, pages 14–18. ACM.

Patrick Juola. 2012. An overview of the traditional authorship attribution subtask. In *PAN 2012 Lab, Uncovering Plagiarism, Authorship and Social Software Misuse, held in conjunction with CLEF 2012*.

V. Keselj, F. Peng, N. Cercone, and C. Thomas. 2003. N-gram based author profiles for authorship attribution. In *Proceedings of the Pacific Association for Computational Linguistics*, pages 255–264.

Moshe Koppel, Jonathan Schler, and Shlomo Argamon. 2011. Authorship attribution in the wild. *Language Resources and Evaluation*, 45:83–94.

David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. 2004. Rcv1: A new benchmark collection for text categorization research. *J. Mach. Learn. Res.*, 5:361–397, December.

Kim Luyckx and Walter Daelemans. 2008a. Authorship attribution and verification with many authors and limited data. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 513–520, Manchester, UK, August.

Kim Luyckx and Walter Daelemans. 2008b. Personae: a corpus for author and personality prediction from text. In *Proceedings of LREC-2008, the Sixth International Language Resources and Evaluation Conference*.

Kim Luyckx and Walter Daelemans. 2010. The effect of author set size and data size in authorship attribution. *Literary and Linguistic Computing*, pages 1–21, August.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: the Penn treebank. *Comput. Linguist.*, 19(2):313–330, June.

A. Narayanan, H. Paskov, N.Z. Gong, J. Bethencourt, E. Stefanov, E.C.R. Shin, and D. Song. 2012. On the feasibility of internet-scale author identification. In *Proceedings of the 33rd conference on IEEE Symposium on Security and Privacy*, pages 300–314. IEEE.

S. Plakias and E. Stamatatos. 2008. Tensor space models for authorship attribution. In *Proceedings of the 5th Hellenic Conference on Artificial Intelligence: Theories, Models and Applications*, volume 5138 of *LNCS*, pages 239–249, Syros, Greece.

Sindhu Raghavan, Adriana Kovashka, and Raymond Mooney. 2010. Authorship attribution using probabilistic context-free grammars. In *Proceedings of the 48th Annual Meeting of the ACL 2010*, pages 38–42, Uppsala, Sweden, July. Association for Computational Linguistics.

E. Stamatatos. 2007. Author identification using imbalanced and limited training texts. In *Proceedings of the 18th International Workshop on Database and Expert Systems Applications, DEXA '07*, pages 237–241, Sept.

E. Stamatatos. 2008. Author identification: Using text sampling to handle the class imbalance problem. *Information Processing and Managemement*, 44:790–799.

I. H. Witten and E. Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kauffmann, 2nd edition.

Rong Zheng, Jiexun Li, Hsinchun Chen, and Zan Huang. 2006. A framework for authorship identification of online messages: Writing-style features and classification techniques. *Journal of the American Society for Information Science and Technology*, 57(3):378–393.