12-12-2018

An N-gram-based Approach for Detecting Social Media Spambots

Tianyu Wang
Pace University, tianyu.wang@pace.edu

li-Chiou Chen
Pace University, lchen@pace.edu

Yegin Genc
Pace University, ygenc@pace.edu

An N-gram-based Approach for Detecting Social Media Spambots

Completed Research Paper

Tianyu Wang
Pace University
tianyu.wang@pace.edu

Li-Chiou Chen
Pace University
lchen@pace.edu

Yegin Genc
Pace University
ygenc@pace.edu

Abstract

Ubiquitous nature of social media has transformed the businesses and society. With the rise of AI, artificial accounts, known as bots, have become more pervasive in the society accomplishing complicated tasks. However, they are accompanied with detrimental effects. They can be easily used for spreading fake news or ill-intended information. Detecting such artificial accounts before they cause any harm is an important however complicated task. To detect these spambots at their early stage, we propose a machine learning method that uses content features including n-grams (n many consecutive words) and information entropy. Our method builds up an n-gram dictionary from the content of spam tweets. This dictionary is then used as a benchmark for comparing the similarity of later tweets with the keywords of previous spam tweets. Our proposed n-gram based features have a better performance than the entropy-based feature alone. However, the best performance is achieved when n-gram features and entropy-based features combined. In addition, by using only the first 5% of the data for building n-gram benchmarks, we achieved 85% accuracy in detecting the source authenticity in the remaining data. Our methodology provides insights into the early detection of spambots as well as distinguishing the differences between machine-generated and human-generated information.

Keywords

Nature Language Processing, Classification, bot, Twitter, spam, social media.

Introduction

Social media provides significant improvements on how information is diffused into systems and businesses. Previous research has looked into the transformative impact of social media on organizations and society (Aral et al. 2013). However, the quality of the information shared and the intent for information sharing play a dominant role on whether or not their impact is positive. For example, recent studies reviewed the use of social media for bullying among teenagers, also known as cyberbullying. Similarly, social media can be exploited for the widespread “fake” or low-quality information. Finding episodes of ill-intended or incorrect information is a complex task. Often, it requires the understanding of the content being shared.

The spread of false or low-quality information is typically handled with automated social media accounts, controlled by spambots in platforms such as Twitter. Such accounts have recently raised significant attention due to the potential consequence in political realm. NBC News has published a database of more than 200,000 tweets that Twitter has tied to "malicious activity" from Russia-linked accounts during the 2016 U.S. presidential election (Popken 2018). In addition, there is evidence that Russia might choose the 2018 US midterm elections as a potential target for Russian influence operations (Stukal et al. 2017). Therefore, identifying whether a social media account is authentic (manually managed) or operated by spambots as early as possible has become very important for public information security.

In our research, we categorize twitter accounts based on their authenticity by studying the content they publish. In particular, we study the relationships between the linguistic and information characteristics of social media content and the authenticity of their account. We first deploy a machine learning technique that involves using a sequence of words (n-grams) for detection. Second, drawing from information theory, we study the use of information entropy for the same task. Finally, we propose a novel approach that combines these two techniques. Our experimental results show that the n-gram based approach outperforms the entropy-based model. However, the best performance is achieved when the two approaches are combined as proposed. Next, we study how early we can reach to the desired accuracy rate. That is how much content do we need to train our model in order to achieve acceptable accuracy rates. Our results indicate that as little as 5% of the data is enough to predict the authenticity of the accounts for the rest of the 70% of the tweets around 85% accuracy rate.

This paper is organized as follows. We provide literature review in Section 2. Our proposed n-gram based method for spambot detection is presented in Section 3. We introduce our data collection and experiment design in Section 4. We discuss our results in Section 5. At last, we conclude the paper in Section 6.

Literature Review

The Positive and Negative Impact of Social Media

Social media has become a critical platform that provides many benefits to both organizations and society at large (Aral et al. 2013). Recent studies show that at the organizational level, social media can positively influence the public perception of organizations (Benthaus et al. 2016) and the brand recognition (Xie and Lee 2015) if their use are managed properly. They can help other marketing efforts (e.x. word-of-mouth) and can have significant effect on sales (Chen et al. 2015). From an operational perspective, social media use can bring coherence in a decentralized work setting (Forsgren and Byström 2018) or transform stakeholder relationships in service oriented domains such as healthcare (Spagnoletti et al. 2015).

At the society level, social media has played a pivotal role in social change especially during recent significant events such as disasters and political movements. The literature on the social influence of social media suggests their use empowers communication in communities during crises (Leong et al. 2015; Tim et al. 2017), and foster collective action by enabling collective sense making (Oh et al. 2015).

Another stream of research has studied use of social media platforms as large sensor systems to increase our awareness of society. Primary focus of this stream is the user sentiment and how it can explain different phenomena such as user information sharing behavior (Stieglitz and Dang-Xuan 2013); brand and customer perception (Ghiassi et al. 2016); or stock market (Li et al. 2018). Other studies looked at the relationships between social media use and probability of default among borrowers (Ge et al. 2017); firm equity value (Luo et al. 2013); intensity of customer-firm relationships (Rishika et al. 2013); and cryptocurrency evaluation (Mai et al. 2018).

The positive impact of social media can be accompanied with detrimental effects just like in traditional media (Miranda et al. 2016). For example, heavy use of social media, accompanied with the ability to engage with others anonymously can lead to deviant behavior such as cyberbullying (Lowry et al. 2017; Lowry et al. 2016), potential misuse of data by peers (Ozdemir et al. 2017), or spread of incorrect information such as rumors (Oh et al. 2013) and "fake news" (Vosoughi et al. 2018). While the positive impacts have been heavily studied, their detrimental effects and the ill-intended use of social media is still a growing field for IS research.

Detection in Social Media

Finding useful or eliminating ill-intended information in social media requires retrieving human understanding of the high dimensional data that is in the form of text. This poses significant challenges due to the volume of the information and the presence of a large body of irrelevant personal messages. Therefore, we rely on automated means to extract useful information and ignore others. Text mining is used to reduce this dimensionality for various purposes such as to classify (Martens and Provost 2013) or to provide simpler representation of text collections (Blei 2012; Landauer 2007).

Various adoption of text mining techniques has been proposed to make sense of social media content. For example, Genc et al. (2011) proposed a methodology to classify tweets into their topics. Some other implementation of text analytics on social media data such as sentiment analysis (Ghiassi et al. 2016; Li et al. 2018; Stieglitz and Dang-Xuan 2013) studies collective sense making processes (Oh et al. 2015).

Similar approaches have been extended to detect deceptive use of social media such as the spread of rumors (Vosoughi et al. 2018) and fake news (Shu et al. 2017). Considering these deceptive actions are mostly conducted by unauthentic and often automated accounts (aka bots), finding these bots has been an integral part of detecting deception. To that end, many machine learning approaches were presented to solve this problem. These models used sentiment features (Ferrara et al. 2014) as well as others including cross-correlating activities (Chavoshi et al. 2016), or a combination of linguistic or semantic measurements (Chu et al. 2012; Li et al. 2016). Bot detection has been operationalized as either classification (Stukal et al. 2017) or anomaly detection problem (Miller et al. 2014).

However, the performance of the methods in detecting spambots can be further improved from the previous research. Furthermore, most of the methods in previous research require a large data size to train their models. In our research, we developed a new approach combining Natural Language Processing (NLP) and n-grams to identify twitter spam accounts. Our model is able to not only identify spam accounts with better accuracy rates at the early stage of spreading the spams, but also classify them with lower equal error rates. It would be a new direction to design features for social media content and its spam detection.

Methodology

Detection Methods

The purpose of our research is to detect twitter accounts that utilized by spambots to post spam tweets. Our detection method is based only on the content of the tweets and not rely on meta-data associated with the tweets. We suspect that spam accounts used by the same spam bot always use the same set of keywords to spread and inflate news (Nimmo 2017). By calculating the likelihoods of words occur in the early tweets of known spam accounts, we hope to identify new spam accounts controlled by similar types of spambots.

Twitter tweets are a series of text strings, consisting of alphabet, numbers, and special characters, each tweet contains 140 characters at the maximum and 280 characters in trials since Sep 26, 2017 (Perez 2017) when Twitter increased the maximum tweet size. Because of the size of tweets, the content of each tweet is typically a lot shorter than regular articles, such as novels or essays, and most electronic publications, such as blogs or Facebook posts. The size of the tweets presents a challenge for textual analysis.

To solve this problem, we used a combination of techniques from both natural language processing and machine learning. Although the techniques from both are not new, to our knowledge the way we generate features needed for machine learning classifier is novel. We aggregated tweets by twitter account and selected a certain percentage of tweet corpus to create an n-gram based dictionary. These dictionaries then are used in the calculation of n-grams features that are needed for the classifier in additional to entropy-based features. In this paper, we decide to use Random Forest, a very common supervised classification method, as the classifier for our experiments because of its performance.

Random Forest Classifier

Random forests or random decision forests are an ensemble learning method for classification. This method builds a multitude of decision trees at training time and outputs the class of the individual trees (Ho 1995). The main reason that we choose random forest classifier is because it overcomes the overfitting issue of decision tree by introducing random subset to split a region, which is efficient for the size of data that we have. Other classifiers can be used in the future with our design of features.

A forest is the average of the predictions of its trees:

$$\boldsymbol{F}(\boldsymbol{x}) = \frac{1}{J}\sum_{j=1}^{J} \boldsymbol{f}_{i}(\boldsymbol{x}) \qquad (1)$$

where J is the number of trees in the forest.

For a forest, the prediction is the average of the bias terms plus the average contribution of each feature where K is the number of features. The contribution of our research will focus on the n-gram design of obtaining features from the contents of the sample tweets before the learning process of the classifier.

$$\boldsymbol{F(x) = \frac{1}{J}\sum_{j=1}^{J} c_j + \sum_{k=1}^{K}(\frac{1}{J}\sum_{j=1}^{J} contrib_j(x,k))} \qquad (2)$$

Features

In order to train with classifiers, we will need to calculate features for each tweet. In our n-gram approach, we define two types of features including entropy and n-gram based features.

Entropy Based Feature

In information theory, entropy is the expected value of the information contained in each message. This feature computes the entropy of character distribution and measures the randomness of the twitter tweets (Mitchell 1997). We use Shannon entropy which measures the amount of information in a message. Its formula can explicitly be written as

$$\boldsymbol{H(x) = \sum_{i=1}^{n} P(x_i)I(x_i) = -\sum_{i=1}^{n} P(x_i)log_b P(x_i)} \qquad (3)$$

In (3), *x* is the input message, the content of a tweet in our case. *I(x)* is the information content of x. *P(x)* is the probability of the frequency on the input message. *b* is the base of logarithm, and in our research, we use base 10. Note that the lower the probability is, the higher the uncertainty is in the information.

N-Gram Based Benchmark Features

We noticed that spam bot accounts are mainly created by some scripts (Cresci et al. 2017). Those scripts use top word trend or a dictionary to generate spam content. Thus, we could compare the similarity between all of the tweets in a twitter account and the spam benchmark that have been identified to create previous spam tweets. Our research focuses on identifying this benchmark using a set of sample data containing spam contents. The benchmark might be different depending on the goals of the spambots. For example, some spambots are focusing on marketing, some provide information for coupon, and some are for advertisement. More specifically, if tweets of an account have higher similarity score from the previous spam data set, it is more likely to indicate that the account is spamming.

Our n-gram benchmark features are the similarity scores between the aggregated tweets in an account and a n-gram benchmark matrix, like a dictionary that includes keywords in tweets and the frequencies of these keywords. Algorithm 1 shows how the n-gram benchmark matrix, $F_{j,k}$, and the similarity score, S, are calculated for any given class or category j.

Algorithm 1: Calculate n-gram Benchmark and Similarity Score

Input: Benchmark tweet vector $T_{m,j}$ refers to the m^{th} tweet in class j; m refers to the number of benchmark tweets that are used to build the benchmark; k refers to the size of the set containing n-grams from the test tweets. j refers to a specific class or category. t refers to the tweet that to be tested.

Output: Similarity score, S, of a tweet

```
for i = 1 to m
    Compute $n - gram\ dictionary\ L_{j,k} += n - grams$ for $tweet\ T_{i,j}$
for i = 1 to k
    Compute $C_{m,k}$ = the frequencies of all $T_{m,j}$ for $L_{j,i}$
Compute $F_{j,k} = \sum_{i=1}^{m} C_{i,k}$
for i = 1 to k
    Compute $P_{j,k}$ = the frequency for $t$ based on n-gram dictionary $L_{j,i}$
Compute $M = P_{j\times k} \times [F_{j\times k}]^T$
Return $S = log_{10}M$
```

Table 1. N-gram frequency matrix for a spam bot

	$L_{j,k}$					
$T_{m,j}$	sales have	have discount	sales make	make money	money buy	buy gold
sales have discount	1	1	0	0	0	0
have discount make money buy gold	0	1	1	1	1	1
$F_{j,n}$	1	2	1	1	1	1

For example, in Table 1, we have two sample tweets: "...it is on sales, have discount..." and "... sales... to make money... buy gold ...". During text pre-processing, we extract only the nouns and verbs of the tweets. Therefore, we cleaned sample tweets and kept only noun and verb for spam bot j, **$T_{m,j}$**: "sales have discount" and "have discount make money buy gold". More details about text cleaning techniques can be found in Text Processing Section. For n=2, we construct the bi-gram dictionary, **$L_{j,k}$**, from the cleaned sample tweets, = *{sales have, have discount, sales make, make money, money buy, buy gold}*. We then calculate the frequencies of each 2-gram in each tweet. For example, the bi-gram frequency vectors for each of the sample tweet would be [1,1,0,0,0,0] and [0,1,1,1,1,1], respectively. 1 refers that the bi-gram exists in the sample tweet and 0 refers that the bi-gram does not exist in the sample tweet. At the end, we calculate the bi-gram benchmark by summing up the frequencies of each sample tweet to obtain benchmark vector $F_{j,n}$ for the n-grams of spam bot j, = [1,2,1,1,1,1].

Table 2. Frequency vector of a test tweet for calculating similarity

	$L_{j,k}$					
t	sales have	have discount	sales make	make money	money buy	buy gold
click this link to have discount and make money	0	1	0	1	0	0

Suppose we have another tweet "…click this link to have discount and … make money…". Its frequency vector would be [0, 1, 1, 1, 0, 0], as in Table 2. Thus, the similarity score, S, measures the distance between the benchmark vector and the frequency vector of this tweet, as below.

$$S = log_{10}(\ [0,1,0,1,0,0] \times\ [1,2,1,1,1,1]^T) = 0.477$$

Experiment Design

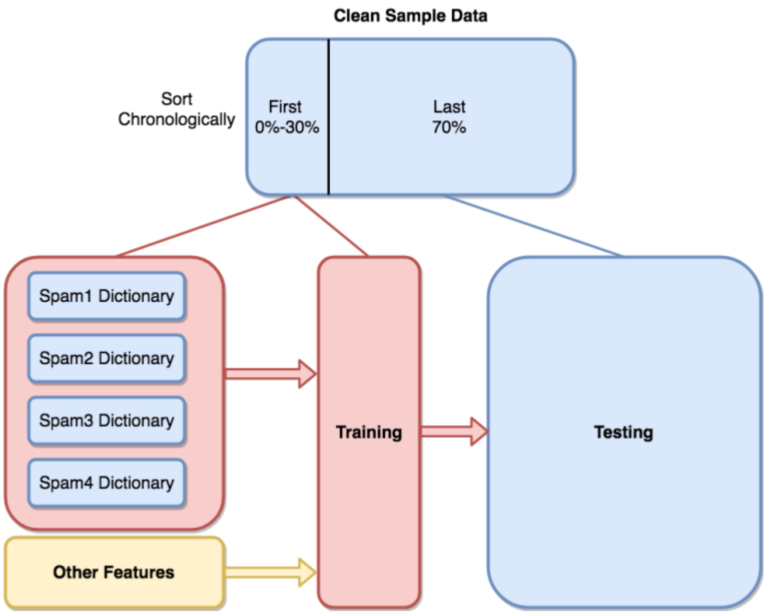

Figure 1. Experiment Flowchart

Figure 1 illustrates the data processing flow in our experiments. First, we pre-processed our tweeter data using NLP techniques. Second, we sorted our data chronologically and split our data in two sections. For each section, we aggregated tweets by accounts. Third, the first section of the data has a dynamic size and was used to build up spam n-gram dictionaries. We calculated n-gram-based features (similarity scores for all groups) and the entropy-based feature and then fit them into Random Forest classifier. Finally, we tested our model using the other section of data. Note that we fixed the portion our testing data as 70% so that we may have a fair comparison of our experiment results when we adjusting the percentage of spam data usages for early detection.

Data Collection

The dataset was collected from MIB Datasets which includes five categories (Cresci et al. 2015). The data set includes tweets from genuine human accounts, tweets posted by traditional spambot, and tweets posted by two types of social spambots as well as tweets from fake followers. The data set is annotated by CrowdFlower (Cresci et al. 2017). Traditional spambot refers to the dataset used by Yang and social spambots refers to spammers of paid apps for mobile devices and products on sale at Amazon.com (Cresci et al. 2017; Yang et al. 2013).

We re-labeled the dataset into 2 groups, as in Table 3. Our goal is to train the classifier to determine if a specific twitter account is controlled by a spambot based on the contents of the tweets posted by the account. This research problem is a binary classification problem in which each input entry, a Twitter account, would be classified as either human or spam.

Table 3. Dataset Source

Source	Number of Accounts	Number of Tweets	Year Collected	Label
genuine accounts	3474	8377522	2011	human
traditional spambots #1	1000	145094	2009	spam
social spambots #2	3457	428542	2014	spam
social spambots #3	464	1418626	2011	spam
fake followers	3351	196027	2012	spam

Tweet Sampling & Aggregation

Since our goal is to determine if an account is controlled by a spambot, we need to observe the aggregated behavior exhibited by all of the tweets posted by the same account. Therefore, in order to reduce computational complexity and re-balance the data, we randomly re-sampled the database by account, aggregated our dataset by accounts, and combined all of the tweets in each account as one data sample. Table 4 shows the volume of data in our experiments in each of the five sources with 1,000 different accounts in total. Pseudo code can be found in Appendix A.

Table 4. The volume of the dataset in each category

category	Number of unique accounts	Number of unique tweets	Label
human	400	209,794	human
spam1	150	20,179	spam
spam2	150	6,565	spam
spam3	150	43,580	spam
spam4	150	9,954	spam
Total	1000	290,072	

Text Processing

Before we calculate features for our classification, we need to pre-process the tweets. Textual data such as tweets is different from numerical data. Such data is represented in human language and is not easy to directly convert it into quantitative format. In addition, processing raw text directly could be very noisy due to some of the text content may not contain useful information. We used natural language processing toolkit (NLTK) to process our data (Loper and Bird 2002). Detail processing steps are shown in Figure 2. The main texts of tweets were first extracted from raw data. Numbers and punctuations were then removed. The texts were then stemmed and lemmatized. Finally, the verbs and nouns were extracted and the stop words are removed.

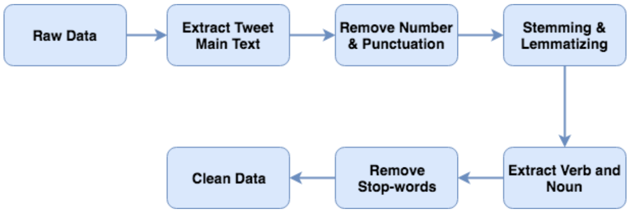

Figure 2. Text Processing

Tweet's Main Text

A Twitter tweet contains five different entities (shown in Figure 3) including prefix RT, @username, text content, short-URL, and #hash-tag. Prefix RT indicates whether the tweet is re-broadcasted from another account. A short-URL is a dynamically generated URL of a website. @Username shows the interactive relation between each account. A hash-tag reflects the assigned topic. All of them are optional in a tweet except for the text content. In our research, we only focused on the analysis of text content because we would like to know how accurate we can detect the spambot only based on the text without other meta-data, such as how the tweets are posted and the interaction among the accounts. We removed all of other entities during text processing.

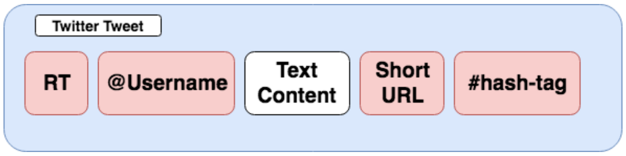

Figure 3. Twitter's Tweet Structure

Remove numerical and punctuation characters

In a main content of a tweet, it could contains a series of numerical characters (0-9) or punctuation characters (,.~!@#$%^&*()[]<>…). These characters would be treated as noisy signal in the analysis of text. Thus, we cleaned up those characters and kept only the alphabetic characters.

Stemming and Lemmatizing

The goal of both stemming and lemmatization is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form. Stemming algorithms work by cutting off the end or the beginning of the word, considering a list of common prefixes and suffixes that can be found in an inflected word. Lemmatization, on the other hand, takes into consideration the morphological analysis of the words. Figure 4 shows two examples.

Figure 4. Examples of Stemming and Lemmatizing

Stemming: cat, cats, cat's, cats' => **cat**

Lemmatizing: am, are, is => **be**

In our text processing steps, we use both stemming and lemmatizing techniques to decrease the derivation of texts so that our textual analysis will work on the sole meaning of text content. More specifically, we use NLTK's Snowball Stemmer and WordNet Lemmatizer to achieve our goal (Porter 2001).

Extract Verbs and Nouns

In human language, especially English, nouns and verbs are the main entities of communications. In a sentence, the majority of the useful meaning comes from verbs and nouns. Therefore, in our experiments, we extract only verbs and nouns for later textual analysis.

Remove Stop-words and Single Letter Words

A stop word is a commonly used word (such as "the", "a", "an", "in") that usually does not contribute too much in the content of text. Another example of stop-words is "www" in URL, "RT" in a tweet, and preposition in English. Therefore, removing those commonly used words would help increase with our textual analysis.

Table 5. Samples of Clean Data

account identifier	category	text
482517693	human	today collection daakuday
1002202471	spam	introduction tolkien review
100219528	spam	lost pounds ketones everyone

Table 5 shows five samples of clean data after text processing. The clean data is then used to calculate features for classification experiments.

Feature Calculations

We first calculated the entropy of characters for all of the tweets in each account. We then used bi-gram (N=2) to calculate the n-gram benchmark matrix. We used 30% of spam accounts as our baseline in each source to construct the N-gram benchmark matrix. We will vary this percentage later and analyze its impact in the discussion section. When training and testing the classifier, we calculated similarity scores of an account between its tweets and each of the four n-gram benchmark matrixes. The higher the score is, the more likely this account is to come from that spam source. Table 6 shows samples of features for tweets of sample accounts.

Table 6. Samples of Tweets and its Associated Similarity Scores

Label	text	spam1	spam2	spam3	spam4
spam	confucius say	34.38	46.31	1409.29	16.44
human	shep help place	130.84	106.47	166.02	123.66
spam	koop huis...	25.53	3.86	13.85	7.31

Results and Discussions

Classification Results

Using supervised machine learning method, we trained our data using Random Forest classifier. We used Python, Skit-learn and Matplotlib packages to implement the experiments (Rossum 1995) (Pedregosa et al. 2011) (Hunter 2007). Note that we used L2-Normalization in our classification.

Table 7. Testing Results[1]-30% Data for Benchmark Matrixes

	spam
TPR	0.967
TNR	0.954
FPR	0.046
FNR	0.033
FAR	0.020
FRR	0.076
EER	0.048
ACC	0.959

Table 7 shows the classification results using Random Forest classifier. True Positive Rates (TPR) for spam is 96.7%. True Negative Rates (TNR) is 95.4%. Equal Error Rates (EER) is 4.8%. Overall accuracy rates (ACC) is 95.9%. It shows how accurate our predictions are for the testing cases in our experiment. Note that our TPR is higher than TNR. It indicates our model can detect spam account better than non-bot account.

Discussion

In the first experiment, we used 30% of data to calculate four n-gram benchmark matrixes. In order to find an optimal percentage for constructing the matrixes and investigate how this percentage impact our results, we run several additional experiments.

Early Detection of Spambots

Our detection method will be helpful for social media platform if it can detect spam tweets at the early stage when a specific type of spambot is spreading spam tweets. To understand how effective our method is for early detection, we sorted our dataset chronologically and extracted only the first 5%-30% of tweets to calculate the n-gram benchmark matrixes. By doing so, this method can build up its spam keyword dictionary using the early tweets spread by spambot.

Table 8 shows the classification results using a range from the first 5% to the first 30% of data for benchmark matrixes.

[1] TPR: True Positive Rate; TNR: True Negative Rate; FPR: False Positive Rate; FNR: False Negative Rate; EER: Equal Error Rate; ACC: Accuracy Rate.

Table 8. Model Comparisons for ACC

Spam data Usages	Entropy	Ngram	Entropy + Ngram
5%	0.615	0.903	0.866
10%		0.882	0.927
15%		0.917	0.940
20%		0.914	0.940
25%		**0.919**	0.934
30%		0.917	**0.959**

We are also interested in understanding how effective our proposed n-gram features are without other types of features. We conducted the same experiments but varied the features in the classifier: one with only the entropy feature, the other with only the n-gram based features. Note that for entropy model, the changing of spam usages for dictionary will not affect the results.

In Table 8, Entropy column shows the results using only one feature – entropy. The classification results are not accurate (61.5%). Results shows that using only the entropy feature is not able to achieve decent detection rates. The results using only our proposed n-gram based features are much better than those using entropy alone. The accuracy rates (ACCs) are arriving 88.2% when using first 10% of spam data. Overall accuracy rates are between 88.2%-91.9% when using Ngram features only and between 86.6%-95.9% when using both Ngram and entropy features. Our n-gram method alone seems to work very well for both human and spam groups. The n-gram based features contributed more than the entropy feature and the combination of entropy and Ngram. The reason is mainly because n-gram-based frequency is able to capture more variation in patterns of the texts than the entropy-based feature. Furthermore, our results indicate that the content of spambot tends to repeat or has similar patterns.

Grid-search Experiment

Since using both N-gram based and entropy features performs the best, we further varied the percentage of data in constructing the n-gram benchmark matrixes between 0% and 30%. These experiments would provide us information on how much data will be needed for an effective early detection.

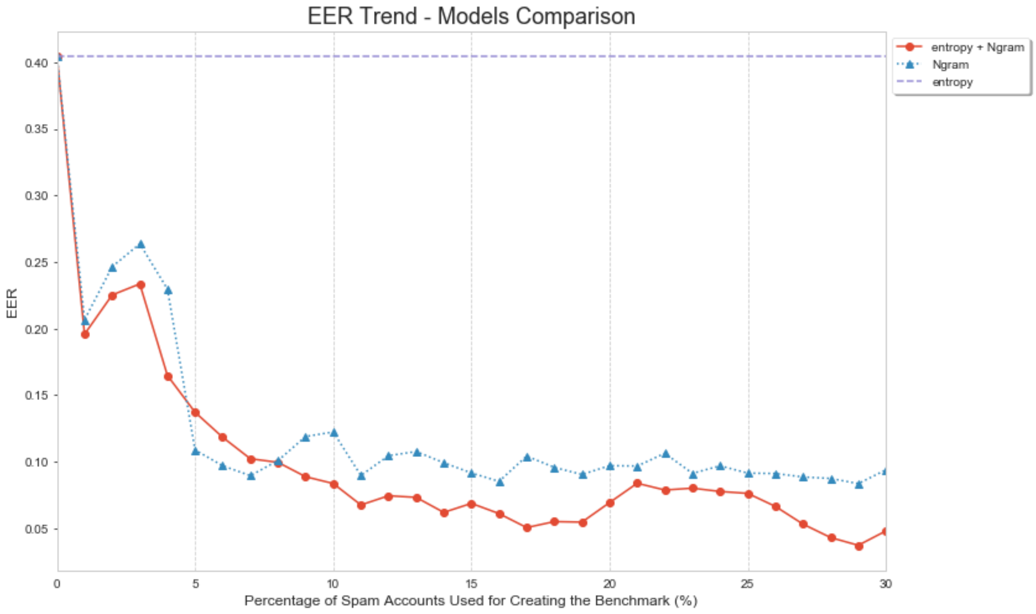

Figure 5. EER Trend

Figure 5 shows a trend of EER in spam detection with three models. Vertical axis is EER and horizontal axis stands for the percentage of spam accounts used for creating the dictionary benchmark. The model using only entropy features illustrates only a flat trend, which indicates the changes of spam features will not affect the results of entropy model. Both the model with spam features alone and model with spam and entropy features show a decreasing trend in EER. The model using both entropy and Ngram features performs the best. Specifically, the EER declines below 15% when using 5% of spam data. EER becomes relatively flat when they reach about 30% of data for n-gram benchmarks.

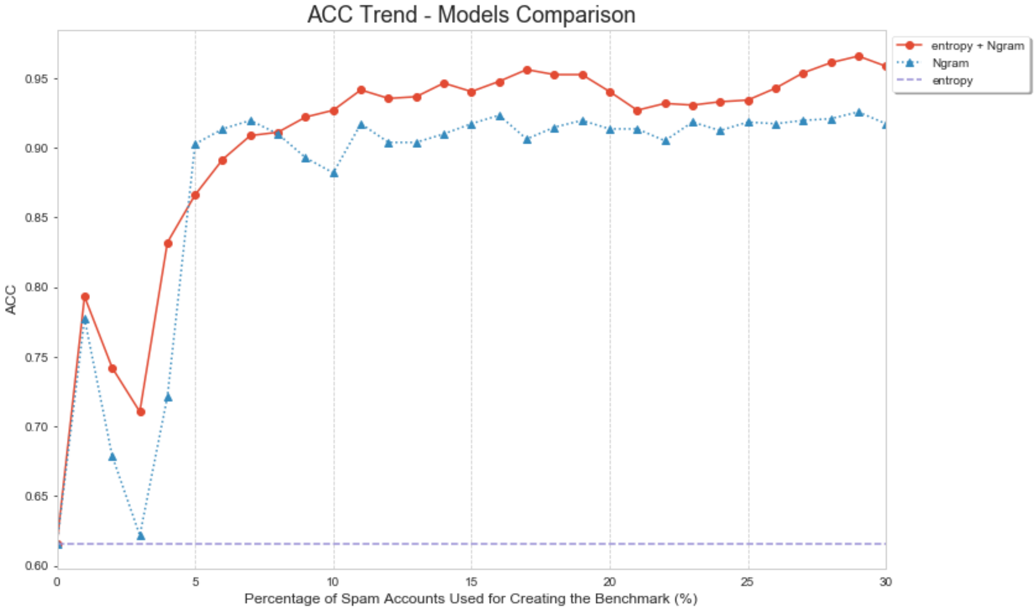

Figure 6. ACC Trend

While EER decreases, ACC has an increasing trend, as in Figure 6. The model using only entropy feature illustrates a flat trend. Both the model with spam features and model with spam and entropy features show an increasing trend. After using more than 8% of spam data for our benchmark, the model using both entropy and Ngram features performs the best. In particular, ACC becomes flat when the benchmark data reaches around 20%. We notice that the ACC arrives above 85% when we only use 5% of spam data to build up the benchmark.

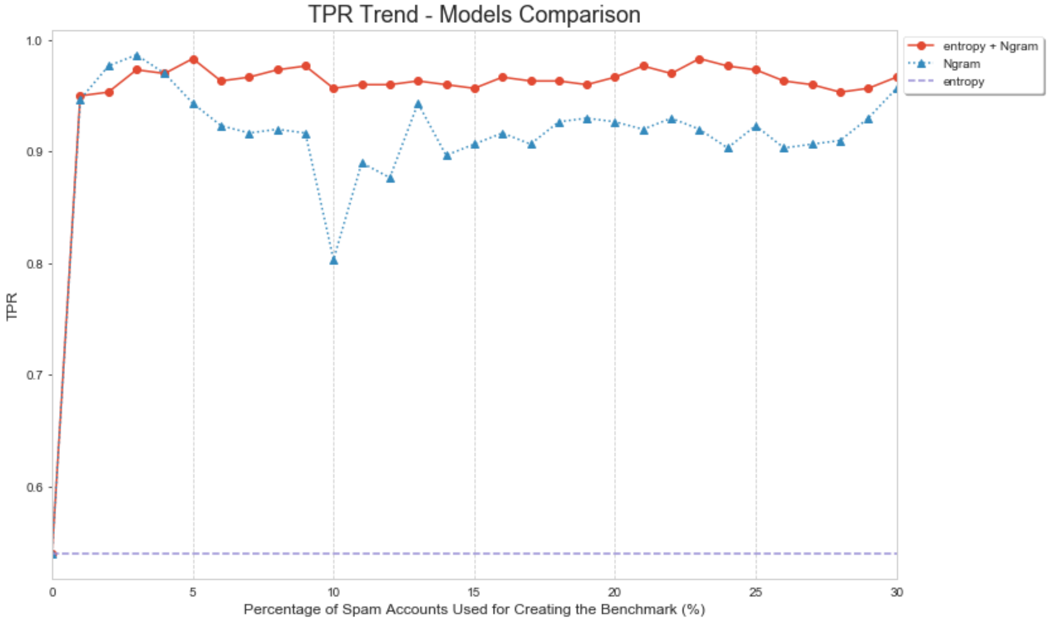

Figure 7. TPR Trend

Since our research is mainly focusing on detection of spambots, the trend of TPR is more important than those in TNR. In Figure 7, it shows a trend of TPR in spam detection with three models. Both the model using Ngram and the model using entropy and Ngram have an increasing trend. The model using both Ngram and entropy features performs the best. In particular, ACC are all above 90% and becomes relatively flat when we use 5% of spam data. This indicates that the combination of using Ngram and entropy features will improve spam true positive rate.

Conclusion

To detect social media spambots at their early stage, we proposed a method that combines a sequence of word frequencies and information entropy of the content to generate features for machine learning algorithms. Specifically, frequencies of bi-grams (two consecutive words) extracted from spam tweets allowed us to determine source authenticity with an average of 96.7% true positive accuracy and 95.9% overall accuracy. The accuracy of True Positive Rate was improved when an entropy-based feature is added to the algorithm. In addition, we demonstrated that our method can be considered as an early detector of twitter spambots. By only using the first 5% of data for building bi-gram benchmarks, we achieved 85% accuracy in detecting the source authenticity for the remaining data.

Since bots are heavily used in spreading ill-intended information, our methodology has important practical implications. By detecting such accounts early, we can stop the spread of false information in a timely manner. Our findings have also some theoretical implications. Our method implicitly shows that human generated and machine generated content differ in the way how the words are related to each other (frequencies of bi-grams show different patterns) and the amount of information they contain (information entropy is different). Acknowledging such differences can help developing information systems theories in the context of artificial intelligence and human computer interaction.

Appendix A: Pseudo Code in Python

Algorithm: Sampling and Aggregation on Tweet Data

Input: Tweet data frame, **data** with attributes {**user_id, text, timestamp, category**}, target sample size: **sample_size**

Output: aggregated data frame **data_agg**

```
for each category
    user = random.sample(list(data.user_id.unique()), sample_size)
    data_sample = data.loc[data['user_id'].isin(user).sort('timestamp')]
    data_agg = data_sample.groupby(['user_id','category']))['text'].apply(' '.join)
return data_agg
```

Python Method Note:

random.sample(): Return a k length list of unique elements chosen from the population sequence.

groupby(): Group series using mapper (dict or key function, apply given function to group, return result as series) or by a series of columns.

apply(): Apply a function along an axis of the DataFrame

join(): Join columns with other DataFrame either on index or on a key column

References

Aral, S., Dellarocas, C., and Godes, D. 2013. "Introduction to the Special Issue—Social Media and Business Transformation: A Framework for Research," *Information Systems Research* (24:1), pp. 3-13.

Benthaus, J., Risius, M., and Beck, R. 2016. "Social Media Management Strategies for Organizational Impression Management and Their Effect on Public Perception," *The Journal of Strategic Information Systems* (25:2), pp. 127-139.

Blei, D. M. 2012. "Probabilistic Topic Models," *Communications of the ACM* (55:4), pp. 77-84.

Chavoshi, N., Hamooni, H., and Mueen, A. 2016. "Identifying Correlated Bots in Twitter," *International Conference on Social Informatics*: Springer, pp. 14-21.

Chen, H., De, P., and Hu, Y. J. 2015. "It-Enabled Broadcasting in Social Media: An Empirical Study of Artists' Activities and Music Sales," *Information Systems Research* (26:3), pp. 513-531.

Chu, Z., Gianvecchio, S., Wang, H., and Jajodia, S. 2012. "Detecting Automation of Twitter Accounts: Are You a Human, Bot, or Cyborg?," *IEEE Transactions on Dependable and Secure Computing* (9:6), pp. 811-824.

Cresci, S., Di Pietro, R., Petrocchi, M., Spognardi, A., and Tesconi, M. 2015. "Fame for Sale: Efficient Detection of Fake Twitter Followers," *Decision Support Systems* (80), pp. 56-71.

Cresci, S., Di Pietro, R., Petrocchi, M., Spognardi, A., and Tesconi, M. 2017. "The Paradigm-Shift of Social Spambots: Evidence, Theories, and Tools for the Arms Race," *Proceedings of the 26th International Conference on World Wide Web Companion*: International World Wide Web Conferences Steering Committee, pp. 963-972.

Ferrara, E., Varol, O., Davis, C., Menczer, F., and Flammini, A. 2014. "The Rise of Social Bots," (59).

Forsgren, E., and Byström, K. 2018. "Multiple Social Media in the Workplace: Contradictions and Congruencies," *Information Systems Journal* (28:3), pp. 442-464.

Ge, R., Feng, J., Gu, B., and Zhang, P. 2017. "Predicting and Deterring Default with Social Media Information in Peer-to-Peer Lending," *Journal of Management Information Systems* (34:2), pp. 401-424.

Genc, Y., Sakamoto, Y., and Nickerson, J. 2011. "Discovering Context: Classifying Tweets through a Semantic Transform Based on Wikipedia," *Foundations of augmented cognition. Directing the future of adaptive systems*), pp. 484-492.

Ghiassi, M., Zimbra, D., and Lee, S. 2016. "Targeted Twitter Sentiment Analysis for Brands Using Supervised Feature Engineering and the Dynamic Architecture for Artificial Neural Networks," *Journal of Management Information Systems* (33:4), pp. 1034-1058.

Ho, T. K. 1995. "Random Decision Forests," *Document analysis and recognition, 1995., proceedings of the third international conference on*: IEEE, pp. 278-282.

Hunter, J. D. 2007. "Matplotlib: A 2d Graphics Environment," *Computing in Science and Engg.* (9:3), pp. 90-95.

Landauer, T. K. 2007. "Lsa as a Theory of Meaning," *Handbook of latent semantic analysis* (6), pp. 3-34.

Li, J. S., Chen, L.-C., Monaco, J. V., Singh, P., and Tappert, C. C. 2016. "A Comparison of Classifiers and Features for Authorship Authentication of Social Networking Messages," Concurrency and Computation Practice Experience, 29 (14).

Leong, C. M. L., Pan, S. L., Ractham, P., and Kaewkitipong, L. 2015. "Ict-Enabled Community Empowerment in Crisis Response: Social Media in Thailand Flooding 2011," *Journal of the Association for Information Systems* (16:3), p. 174.

Li, T., van Dalen, J., and van Rees, P. J. 2018. "More Than Just Noise? Examining the Information Content of Stock Microblogs on Financial Markets," *Journal of Information Technology* (33:1), pp. 50-69.

Loper, E., and Bird, S. 2002. "Nltk: The Natural Language Toolkit," in: *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics - Volume 1*. Philadelphia, Pennsylvania: Association for Computational Linguistics, pp. 63-70.

Lowry, P. B., Moody, G. D., and Chatterjee, S. 2017. "Using It Design to Prevent Cyberbullying," *Journal of Management Information Systems* (34:3), pp. 863-901.

Lowry, P. B., Zhang, J., Wang, C., and Siponen, M. 2016. "Why Do Adults Engage in Cyberbullying on Social Media? An Integration of Online Disinhibition and Deindividuation Effects with the Social Structure and Social Learning Model," *Information Systems Research* (27:4), pp. 962-986.

Luo, X., Zhang, J., and Duan, W. 2013. "Social Media and Firm Equity Value," *Information Systems Research* (24:1), pp. 146-163.

Mai, F., Shan, Z., Bai, Q., Wang, X., and Chiang, R. H. 2018. "How Does Social Media Impact Bitcoin Value? A Test of the Silent Majority Hypothesis," *Journal of Management Information Systems* (35:1), pp. 19-52.

Martens, D., and Provost, F. 2013. "Explaining Data-Driven Document Classifications," *Information Systems Research*).

Miller, Z., Dickinson, B., Deitrick, W., Hu, W., and Hai Wang, A. 2014. "Twitter Spammer Detection Using Data Stream Clustering," (260), pp. 64–73.

Miranda, S. M., Young, A., and Yetgin, E. 2016. "Are Social Media Emancipatory or Hegemonic? Societal Effects of Mass Media Digitization," *MIS Quarterly* (40:2), pp. 303-329.

Mitchell, T. 1997. *Machine Learning. Macgraw-Hill Companies*.

Nimmo, B. 2017. "#Botspot: Twelve Ways to Spot a Bot." from https://medium.com/dfrlab/botspot-twelve-ways-to-spot-a-bot-aedc7d9c110c

Oh, O., Agrawal, M., and Rao, H. R. 2013. "Community Intelligence and Social Media Services: A Rumor Theoretic Analysis of Tweets During Social Crises," *MIS Quarterly* (37:2).

Oh, O., Eom, C., and Rao, H. R. 2015. "Research Note—Role of Social Media in Social Change: An Analysis of Collective Sense Making During the 2011 Egypt Revolution," *Information Systems Research* (26:1), pp. 210-223.

Ozdemir, Z. D., Jeff Smith, H., and Benamati, J. H. 2017. "Antecedents and Outcomes of Information Privacy Concerns in a Peer Context: An Exploratory Study," *European Journal of Information Systems* (26:6), pp. 642-660.

Pedregosa, F., Ga, #235, Varoquaux, l., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., #201, and Duchesnay, d. 2011. "Scikit-Learn: Machine Learning in Python," *J. Mach. Learn. Res.* (12), pp. 2825-2830.

Perez, S. 2017. "Twitter Trials Expanding Tweets from 140 Characters to 280."   Retrieved 09/26/2017, 2017, from https://beta.techcrunch.com/2017/09/26/twitter-trials-an-expansion-beyond-140-characters/

Popken, B. 2018. "Twitter Deleted 200,000 Russian Troll Tweets. Read Them Here." 2018, from https://www.nbcnews.com/tech/social-media/now-available-more-200-000-deleted-russian-troll-tweets-n844731

Porter, M. F. 2001. "Snowball: A Language for Stemming Algorithms."

Rishika, R., Kumar, A., Janakiraman, R., and Bezawada, R. 2013. "The Effect of Customers' Social Media Participation on Customer Visit Frequency and Profitability: An Empirical Investigation," *Information Systems Research* (24:1), pp. 108-127.

Rossum, G. 1995. "Python Reference Manual," CWI (Centre for Mathematics and Computer Science).

Shu, K., Sliva, A., Wang, S., Tang, J., and Liu, H. 2017. "Fake News Detection on Social Media: A Data Mining Perspective," *ACM SIGKDD Explorations Newsletter* (19:1), pp. 22-36.

Spagnoletti, P., Resca, A., and Sæbø, Ø. 2015. "Design for Social Media Engagement: Insights from Elderly Care Assistance," *The Journal of Strategic Information Systems* (24:2), pp. 128-145.

Stieglitz, S., and Dang-Xuan, L. 2013. "Emotions and Information Diffusion in Social Media—Sentiment of Microblogs and Sharing Behavior," *Journal of management information systems* (29:4), pp. 217-248.

Stukal, D., Sanovich, S., Bonneau, R., and Tucker, J. A. 2017. "Detecting Bots on Russian Political Twitter," *Big data* (5:4), pp. 310-324.

Tim, Y., Pan, S. L., Ractham, P., and Kaewkitipong, L. 2017. "Digitally Enabled Disaster Response: The Emergence of Social Media as Boundary Objects in a Flooding Disaster," *Information Systems Journal* (27:2), pp. 197-232.

Vosoughi, S., Roy, D., and Aral, S. 2018. "The Spread of True and False News Online," *Science* (359:6380), pp. 1146-1151.

Xie, K., and Lee, Y.-J. 2015. "Social Media and Brand Purchase: Quantifying the Effects of Exposures to Earned and Owned Social Media Activities in a Two-Stage Decision Making Model," *Journal of Management Information Systems* (32:2), pp. 204-238.

Yang, C., Harkreader, R., and Gu, G. 2013. "Empirical Evaluation and New Design for Fighting Evolving Twitter Spammers," *IEEE Transactions on Information Forensics and Security* (8:8), pp. 1280-1293.