Emotional Bots: Content-based Spammer Detection on Social Media

Panagiotis Andriotis
University of the West of England, Computer Science Research Centre, Bristol, BS16 1QY, U.K.
panagiotis.andriotis@uwe.ac.uk

Atsuhiro Takasu
National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, 101-8430, Japan.

Abstract

Recent research indicates that a considerable amount of content on social media is generated by automated accounts. The automata present sophisticated behavior – mimicking humans– aiming at evading traditional detection methods. In this paper, we present a supervised approach to detect automated accounts on Twitter using mainly content-based features. We performed our experiments using four datasets that contain tweets from almost 20K malicious and benign accounts. Our methodology is lightweight and employs users' metadata, content and sentiment features. It performs well on unseen data (0.95 F1-score) reaching 95% precision and recall. This work also demonstrates that sentiment characteristics can add value to social spambot detection algorithms when combined with known features.

1. Introduction

Social media allow people to seamlessly connect with acquaintances across the globe. Microblogging platforms such as Twitter are among the most popular online networks considering their active users (330 million accounts as of April 2018) [34]. However, a large proportion of these users are automata (*bots*), which take advantage of the platform's openness, its flexible policies and its powerful APIs [39].

Prior work on bot detection focuses on exploiting distinctive properties that differentiate human-curated accounts from bots. Research also aims at identifying compromised accounts [12] and Sybil nodes in social media [8]. Furthermore, scholars use supervised and unsupervised learning [43] to recognize fake accounts. Varol *et al*. [41] recently proposed a scheme that utilizes 1,150 features to identify bots and Gilani *et al*. [18] also used the content (i.e. media) uploaded on Twitter from the account into question. Other researchers spot spam messages [37] and accounts that abuse trending topics [51].

In this paper, we use a variety of features derived from Twitter metadata and textual content posted on users' timelines to train a supervised classifier and distinguish humans from bots. We focus on the effectiveness of using features that model sentimental attributes of the accounts. The analysis shows that contemporary bots mimic human behavior, producing sentimental fingerprints similar to those of humans. Moreover, we estimate users' preferences on topics of interest employing a known text modeling method and combine these attributes to strengthen the classifier's accuracy. Finally, we evaluate the persistence of the produced model against time. We show that it maintains its efficiency in identifying humans without the need to be retrained.

Therefore, this work makes the following contributions: a) We conduct a large scale study using four labeled datasets describing accounts as humans or bots. The datasets contain entities with different characteristics, hence they resemble a representative fraction of the Twitter population; b) We show that sentiment analysis can assist in classification tasks where limited numbers of metadata and textual content-based features are utilized; c) We evaluate the effectiveness of our approach against time and we show that the identification of humans is sustainable using a classifier trained on older data.

2. Related Work

The proliferation of automated accounts on social media, also known as "bots", has been early identified by the research community [51]. In the past, researchers estimated that 16% of spammers were bots [19]. Social bots serve numerous causes, either malicious or benign. Ferrara *et al*. [14] illustrate various threats derived from the sophisticated behavior that modern social botnets exhibit. To name a few, they can be used to spread misinformation and disseminate rumors [3], or direct discussions on microblogging services about elections [4].

Bot detection on social networks has been mainly considered as a binary classification problem [2, 35]. Feature extraction relies on observed attributes that characterize

humans' behavior when compared with automated agents. *Network structure* has been used extensively in prior work; the number of followers and friends, the ratio between followers and friends and the number of reciprocal connections are used as features [13, 21, 35]. In our work, we also utilize posts' *textual content*. Although the posted content is not similar among all types of bots, there exist properties that differentiate them from humans. For example, bots are usually trying to tempt users to click on a desired URL, hence they often include URL links in their posts [42, 46]. Other features that have been used in previous research are the *sentiment* [31, 41], which can be extracted from short texts [1, 36], the *length* of the tweet, the *tweet rate* and other *structural attributes* of messages [28, 38, 47], such as the number of *mentions*, *hashtags* and *URLs* [23, 45].

Our approach is based on a probabilistic text modeling algorithm, namely Latent Dirichlet Allocation (LDA) [5]. LDA-based methodologies have been presented lately aiming at detecting fake reviewers or spammers and bots on microblogging services [22]. Yang *et al*. [50] create an active spammer collection approach using LDA on Twitter that enhances passive traditional honeypots. Nilizadeh *et al*. [29] use message propagation dynamics in neighborhoods defined by a collection of LDA-derived topics. Other researchers work with datasets derived from sites semantically similar to Twitter [52] (e.g. *Weibo*). When evaluating their methodology using the standard metric *F-Measure*, they report an accuracy of 90.67%. Liu *et al*. [24] use LDA-based attributes and posts' similarity jointly with network and behavior features to detect spammers on Weibo, but the achieved accuracy is not competitive. Other researchers use a LDA-based feature set that estimates the accounts' level of interest on specific topics, derived from a corpus that incorporated Twitter and translated Weibo texts [23]. Using features proposed by [21] in conjunction with normalized probabilities (stemmed from the LDA model), they achieve F1-score=0.949 with an *Adaboost* classifier.

An Adaboost classifier is also employed in [28] where the main goal is to strike the balance between precision and recall, increasing the recall in detecting bots. The authors suggest that the best performance of their classifier is achieved with 200 topics. However, the overall performance (F1-score) of this method is limited. Wei *et al*. [44] use LDA-based features (200 topics) to study the impact of malicious accounts on Twitter in terms of spreading ideas that can lead to extremism and terrorism. In [45] they also employ sentiment analysis to pinpoint differences between non-suspended and suspended users. They finally recognize the need to investigate the impact of sentiment analysis on bot detection on Twitter [45].

Dickerson *et al*. [11] use sentiment analysis variables to detect bots on Twitter and conclude they can improve detection accuracy. However, they just use hashtags to define topics of interest. Our work aims to bridge the gap among the aforementioned works [11, 21, 23]. We use features introduced in [21] with the LDA-based features proposed by Liu *et al*. [23] and we study the effectiveness of sentiment analysis as suggested in [11, 45].

3. Datasets

We use three datasets for training in this work (8,973,320 tweets in total). The first (`ICWSM17`) [41] contains Twitter handles of 2,573 accounts, annotated as *humans* (1,747 accounts) or *bots* (826 accounts). The second (`RAID11`) [48, 49] contains usernames and data derived from 1M *malicious spammers* and 10M *normal users* on Twitter. The final dataset (`WWW17`) [7] consists of various accounts (14,398 in total). It contains information for *humans* (namely "genuine"), *social* and *traditional spambots*, and *fake followers*. All these categories except the "genuine" accounts are generally treated as *bots* in this paper. The latter dataset, accumulated by Cresci *et al*. [7] contains information about 3 sets of social spambots and 4 sets of traditional spambots.

3.1. Data pre-processing

Before extracting features, we apply well-known techniques to clear the data. First, we only take into account tweets written in English. To achieve this, we employ *langdetect* [9], a Python language detection framework ported from a Java language detection library [33]. Tweets not written in English (according to the library's output) are ignored. Next, we remove non-ASCII characters from the short texts and we consider accounts which contain no less than 5 tweets in order to be able to extract topic features from their timelines. Additionally, we remove from each tweet: a) web page links (starting with *'http'* or *'https'*), b) mentions to Twitter users (starting with '@'), c) hashtags (starting with '#'), and finally, d) we exclude the prefix *'RT'*, which denotes retweets of other users' content.

3.2. Topic Modeling

The Latent Dirichlet Allocation (LDA) [5] is a Natural Language Processing model that provides an explicit representation of a document using topic probabilities. LDA has been used recently for bot detection on micro-blogging services [23, 28, 29]. We use MALLET [26], a Java-based package for statistical natural language processing and text modeling, which utilizes LDA. MALLET accepts a document (or a set of documents) as an input and *trains* a model describing topic probabilities for this document. This model can be used to *infer* topic probabilities for other documents. The process is done efficiently with MALLET which uses a scalable implementation [26] of the Gibbs sampling method [16].

In our work, following other scholars' practice [29, 44, 45], we merge all tweets of an account in one single doc-

Table 1. Feature vector description

	Features
Metadata	F1: statuses_count F2: followers_count F3: friends_count F4: favourites_count F5: Followers/Friends (F2 + 1/F3 + 1)
Content	F6: RT Ratio F7: # Ratio F8: @ Ratio F9: URL Ratio
Sentiment	F10: Sentiment (vector)
Topics	F11: f_{GOSS}: LDA (vector) F12: f_{LOSS}: LDA (vector)

ument (after performing data pre-processing, as described previously in section 3.1). According to Wei *et al*. [44] topic modeling provides noisy topic distributions for short texts (such as a Twitter feed) [20]. Thus, researchers usually aggregate all tweets produced by a user in a single document.

We train our topic model using 19,851 documents (aggregated tweets from each account) derived from the datasets. Note that when training our LDA model, we also take into account data from Twitter users having less than 5 tweets in their timelines. We extract 200 topics to create the model because this number of topics is commonly used in other recent works [23, 45]. In addition, before training the model, we remove *stop words*, which is a set of english common words, such as *she, is, as*, etc. [29].

MALLET outputs 200 topic probabilities for each document. These probabilities describe the topics of interest of each account. The model is saved to allow us to use it for topic inference when there is a need to estimate topic probabilities of similar documents. We employ a Python wrapper from the *gensim* library [32] to train our model and infer topic probabilities. Other parameters are set as follows: *iterations = 50, optimize_interval = 10*.

3.3. Feature Extraction

Metadata Features User metadata features have been used in the past to distinguish humans from bots [27]. We use features widely employed by other researchers [41]. The following features constitute our metadata feature vector: Number of i) *statuses* (including retweets), ii) *followers*, iii) *friends* (a.k.a. followings), and iv) *likes*. Also, we consider the: v) *followers to following* ratio (FtF). Note that we add one unit to the numerator and denominator of FtF (see Table 1) to account for cases where either of the two (or both) is equal to zero.

Content-based Features As discussed in Section 3.2 we accumulate all tweets of an account in a single document. Thus, we estimate the following content-based features: i) *retweets* ratio, i.e. the number of retweets (represented as 'RT' at the beginning of a single tweet) existing in a user's account, divided by the total number of tweets, ii) *hashtag* ratio (number of # divided by the total of tweets), iii) *mention* ratio (number of @ divided by the total of tweets), and iv) *URL* ratio (number of URL links to the total of tweets).

Moreover, the content-based feature vector includes sentiment analysis attributes. We utilize *Sentistrength* [36], a sentiment analysis and opinion mining program that has been successfully used in the past on short texts for various tasks [6, 15, 25]. *Sentistrength* calculates a positive score (from 1, i.e. not positive, to 5, i.e. extremely positive) and a negative score (from -1, i.e. not negative, to -5, i.e. extremely negative) for a short text. Note that when using *Sentistrength*, we input raw, unprocessed tweets, because *Sentistrength* performs its own data processing before extracting sentiment scores.

Therefore, for each Twitter account in our dataset we calculate the positive and negative scores of each tweet with *Sentistrength*. Then, we estimate the *mean value* and the *variance* for positive and negative scores. Thus, the sentiment feature vector for each Twitter account, i.e. F10 (Table 1) is as follows: [mean(positive), mean(negative), variance(positive), variance(negative)].

Finally, we use features derived by LDA topic modeling as suggested by Liu *et al*. [23]. In their recent paper, the authors are using two feature vectors, namely *Global Outlier Standard Score (GOSS)* and *Local Outlier Standard Score (LOSS)*. They combine their $GOSS$ and $LOSS$ vectors with common features proposed by Lee *et al*. [21] and they achieve high accuracy (F1-score = 0.949) on bot detection using two datasets: a) a Social Honeypot dataset that basically consists of traditional spam accounts, and b) a "smart" spammers dataset derived from the Chinese Sina Weibo microblogging network.

We estimate these feature vectors using Liu *et al*.'s [23] method as follows. Each Twitter account i is represented as a vector X_i of K topics; hence $X_i = [x_{i1}, x_{i2}, \ldots, x_{iK}]$, where $x_{ik} \in [0, 1]$ is the estimated (inferred) topic probability extracted by a topic modeling algorithm (such as LDA, e.g. using MALLET) of the i^{th} account for the k^{th} topic ($i, k, n \in \mathbb{N}$). Therefore, if the dataset D consists of n accounts, then D can be seen as: $D = [X_1; X_2; \ldots; X_n]$.

To evaluate the i^{th} account's interest on a certain topic k, compared to the rest of the accounts (n is the total of accounts), Liu *et al*. [23] propose the use of the following equation (equation 1) [23], where $\mu(x_k) = \frac{\sum_{i=1}^{n} x_{ik}}{n}$:

$$GOSS(x_{ik}) = \frac{x_{ik} - \mu(x_k)}{\sqrt{\sum_i (x_{ik} - \mu(x_k))^2}} \quad (1)$$

Thus, the interest of the i^{th} account on the set of K topics derived from our model ($K = 200$), can be represented by the following feature vector: $f^i_{GOSS} = [GOSS(x_{i1}), GOSS(x_{i2}), \ldots, GOSS(x_{iK})]$, according to [23]. This is our feature F11 (seen in Table 1).

Following the same logic, we also compute the $LOSS$ score of a topic, which estimates the interest of an individual on a topic, considering only her own tweets. Hence, the i^{th} account's interest on a topic k can be estimated by the following equation (equation 2) [23], where $\mu(x_i) = \frac{\sum_{k=1}^{K} x_{ik}}{K} = \frac{1}{K}$:

$$LOSS(x_{ik}) = \frac{x_{ik} - \frac{1}{K}}{\sqrt{\sum_k (x_{ik} - \frac{1}{K})^2}} \quad (2)$$

Similarly, according to [23], the i^{th} account can be modeled as a feature vector, derived from the $LOSS$ scores for each topic. Therefore our F12 feature (Table 1) is: $f^i_{LOSS} = [LOSS(x_{i1}), LOSS(x_{i2}), \ldots, LOSS(x_{iK})]$.

4. Methodology

This Section describes our methodology for training the classification algorithm (Section 4.1). Subsequently, we show the sequence of actions needed to identify if a Twitter account is a human or not (Section 4.2).

4.1. Model Training

We use the `WWW17`, `RAID11`, and `ICWSM17` datasets. All tweets of a single account are aggregated to form one document, representing the user's generated textual content. After cleaning the data (see Section 3.1) we use all these documents as a single corpus and train a LDA model utilizing MALLET. Then, for each account we extract (i.e. infer) topic probabilities using this LDA model. In addition, we extract metadata and content features as discussed in Section 3.3. Therefore, each account is represented by a 413-dimensional feature vector that comprises features F1 - F12. The accounts are already labeled as *humans* or *spambots*.

Next, we employ *scikit-learn* [30] to train a supervised model that will distinguish accounts as *humans* or *spambots*. *Scikit-learn* is a Python library that contains the implementation of various algorithms for supervised learning. In this work we tested the performance of the following algorithms (note that their abbreviated forms are provided in parentheses): K-Nearest Neighbors (KNN), Decision Tree (CART), Gaussian Naive Bayes (NB), Support Vector Machine (SVM), Random Forest (RF), AdaBoost (AB). We perform 10-fold cross validation using 80% of the data for training. Finally, we use the rest of the dataset (20%) to evaluate the model with "unseen" data as an additional validation experiment. We perform 9 experiments to evaluate the effectiveness of the features. The results of this series of experiments are presented in Section 5.

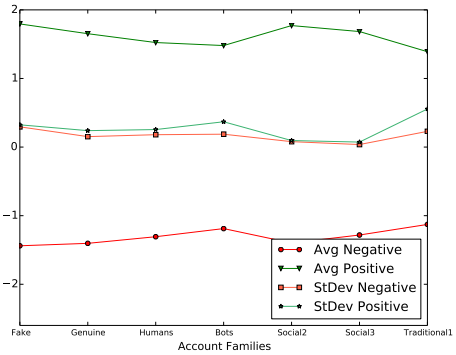

Figure 1. Average positive and negative sentiment scores (and their standard deviations) derived from the account families.

4.2. Account classification

To classify a Twitter account as *human* or *bot*, we perform the following actions. First, we collect tweets from the user's timeline. Currently, Twitter returns up to the most recent 3,200 tweets (and retweets) [40]. We also extract features F1 - F5, which are the metadata features used by the classifier. Then, we extract content (F6 - F9) and sentiment (F10) features from the gathered texts. Finally, we aggregate the account's tweets in a single document, we clean the data (see Section 3.1) and infer topic probabilities using the trained LDA model. These topic probabilities are used to estimate features F11 and F12. F11 and F12 highlight topic preferences of the user based on the community (other Twitter accounts) we used when training our LDA model. Finally, the classifier draws on the feature vector to make a decision (*human* or *bot*).

5. Results

In this section we present the results of our experiments. First, we estimate the mean values and deviations of the positive and sentiment scores retrieved from the `WWW17` and `ICWSM17` datasets, which aggregate a variety of user profiles (humans and bots): a) Humans –*genuine, humans*, b) Bots –*Traditional, Bots, Social, Fake*. We use *sentistrength*, which calculated the positive and negative scores for each tweet in a single account; then we measure their mean values for each account. Figure 1 shows the averages of the mean values of the positive and negative scores (and their standard deviations) for each account family. The figure shows that traditional bots compared to social spambots or fake followers, present lower sentimental (positive and negative) mean values. Also, social spambots and fake followers demonstrate higher positive sentimental values compared to the majority of the human accounts. However, these sentimental features are not strong enough to reliably distinguish humans from bots (as seen below).

Table 2. Comparison of algorithms' accuracy (F1-score) using various features

	Sentiment	Topics	Sentiment &Topics	Metadata	Metadata &Content	ALL excl. Topics	Metadata &Topics	ALL excl. Sentiment	ALL features
KNN	0.661107	0.819944	0.824914	0.910039	0.910039	0.910039	0.910039	0.910039	0.910039
CART	0.688805	0.867108	0.873938	0.908012	0.90552	0.907356	0.933214	0.933874	0.934044
NB	0.664253	0.847038	0.847252	0.671045	0.671041	0.671118	0.671045	0.67108	0.671118
SVM	0.553014	0.814249	0.816872	0.873201	0.876722	0.876797	0.882621	0.883766	0.885067
RF	0.689279	0.819908	0.835809	0.911552	0.915223	0.913716	0.8362	0.82735	0.854269
AB	0.68537	0.894014	0.898837	0.91363	0.918699	0.91841	*0.938365*	0.938441	**0.938844**

Table 3. F1-score achieved by the AB classifier for unseen data using various features

	Sentiment	Topics	Sentiment &Topics	Metadata	Metadata &Content	ALL excl. Topics	Metadata &Topics	ALL excl. Sentiment	ALL features
AB	0.741963	0.922216	0.924799	0.931688	0.93628	0.939724	0.947761	0.948909	**0.950631**

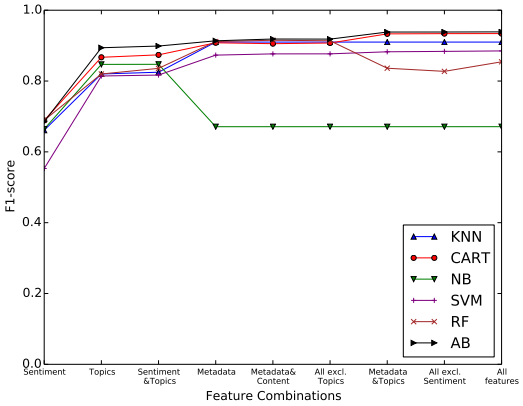

Figure 2. F1-score achieved by various classifiers

Then we train six classifiers (used in similar research works) to identify bots and humans utilizing different sets of features. First, we train the classifiers with sentiment features only (F10), then with topic features only (F11 and F12) and then considering both categories (F10 - F12). Afterwards, we train the classifiers with metadata features only (F1 - F5), then with metadata and content (F1 - F9), then with all features excluding topics and then considering only metadata and topics (F1 - F5 and F11 - F12). Finally, the classifiers are trained with all features excluding sentiment, and then considering all features (F1 - F12). We calculated the F1-scores after performing these experiments and aggregated them in Table 2 and Figure 2. The best performance when only the sentiment features were considered during training is achieved from the Random Forest classifier. The Decision Tree and Adaboost algorithms performed similarly.

As Table 2 shows, when the AB classifier is trained with topic-based features, its accuracy is approximately 0.894. However, Liu *et al*. [23] report higher accuracy (0.934) when they train the Adaboost classifier using the topic-based features on another dataset (Weibo dataset). This might occured by the fact that in their work they use a limited amount of accounts. Additionally, their criteria for choosing "smart" spammers (as they call them) are not well-defined. Moreover, they actually translate Chinese short texts in English to perform their classification task, which might introduced biases. We consider a variety of datasets and only tweets written in English. The results also demonstrate that the classifier's performance does not drastically improve when we combine sentiment and topic features.

The combination of metadata and topics however improves the classification accuracy (F1-score = 0.938365). On unseen data we report an accuracy of approximately 0.948 (Table 3). If the Adaboost classifier is trained with all features excluding sentiment features, the difference in performance is not critical (0.9384). The addition of the sentiment features slightly improves the overall classification ability of the Adaboost classifier. Our proposed feature set reaches an accuracy (average F1-score) of approximately 0.9389. The classifier accomplished an accuracy of 0.9506 on unseen data using our feature set (Table 3).

In general, Adaboost outperforms the other classifiers in our experiments and the Decision Tree classifier achieves similar results as seen in Figure 2. The classifier identifies humans more efficiently (on unseen data): its precision is 0.95, the recall is 0.97 and the F1-score is 0.96 on humans. For bots, the precision is 0.95, the recall is 0.93 and the F1-score is 0.94. It's weighted average accuracy (F1-score) is approximately 0.95 (Table 3). Thus, our classifier is built on concepts presented in similar works [23, 45] but it also improves its accuracy using a small feature set (F1 - F12) with

the addition of sentiment attributes (F10). Finally, the accuracy of our classifier is higher compared to results presented recently by other works [18, 29, 47] (0.8644, 0.91, 0.89, respectively, using different datasets). However, this comparison is based only on the *nominal values* of the achieved accuracy.

6. Evaluation and Discussion

The proposed model for bot detection on microblogging services uses a supervised content-based approach. Although the classifier will probably benefit from frequent retraining (given that it is also based on the level of interest of an account on specific topics), we conduct a final experiment to measure its resilience. We test the accuracy of our classifier on a dataset that presents different characteristics compared to the datasets we used for training. Additionally, the experiment occurs 7 months after the training phase.

In this experiment we utilize `ICWSM17` and a new dataset, namely `ASONAM17` [17, 18]. The latter dataset contains metadata for a variety of users. The accounts have been annotated by humans as being *bots* –i.e. automated agents– or *humans*. Gilani *et al*. [17] state that the unanimous agreement between human annotators for this annotation task was very high (average 89%) compared to the average agreement given by `BotorNot` [10] (approx. 47.9%). `BotorNot`, now renamed as `botometer` [41], uses a large feature vector (1,150 features) that models users' behavior. The `ASONAM17` dataset's accounts are divided into popularity bands starting from accounts having more than 9M followers, and including accounts with approximately 1M followers and 100K followers, respectively. Hence, these accounts present structural differences compared to those we used to train our model. Moreover, we consider in this evaluation only those accounts from the `ICWSM17` and `ASONAM17` datasets that are still "alive". In other words, we exclude public accounts labeled as *"User not found"* or *"User has been suspended"* from Twitter [39]. We crawled Twitter on March 2018 and the number of accounts that were still alive (accessible) can be seen in Table 4. The same Table shows the "survival rate" of these accounts.

Furthermore, we removed from the `ASONAM17` dataset accessible accounts that were originally classified [17] both as humans and bots. After cleaning the data and removing non-English tweets, we only considered accounts that contained more than 5 tweets. In total, for this evaluation we use as a basis 4,567 Twitter accounts. We should note here that sometimes when we tried to estimate the topic-based features (LOSS) proposed by Liu *et al*. [23] we encountered `ZeroDivisionErrors`. This happens when all topic probabilities for an account are estimated to be equal. However, Liu *et al*. [23] do not report this limitation in their work. We encountered some (limited) cases of this kind during our experiments.

Table 4. Accessible Twitter accounts (late March 2018)

Datasets	Accessible Bots	Accessible Humans
`ICWSM17`	728/826 **88.14%**	1,480/1,747 **84.72%**
`ASONAM17`	1,230/1,492 **82.44%**	1,594/1,939 **82.21%**

For each of the aforementioned accounts we follow the methodology described in Section 4.2. We use the model we trained seven months earlier to see its resilience over time. The goal is to assess if it can classify bots and humans without the need to be retrained. The classifier's accuracy (F1-score) when all accounts are considered drops to 0.67. We believe that this decline occurs because the characteristics of the automated accounts existing in the `ASONAM17` dataset are structurally different compared to the other datasets. However, our model maintains its ability to efficiently identify human accounts, especially those which can be seen as *influential* or *popular* (`ASONAM17`). It's accuracy at predicting humans in the `ASONAM17` dataset is 0.9465 and in the `ICWSM17` dataset is 0.8426. The accuracy on human accounts in general is 0.8881. On the other hand, the model fails to maintain its accuracy on predicting bot accounts (0.3028). This outcome can be explained if we consider that automated accounts are usually "interested" in a wide range of topics [23] which eventually can change more frequently compared to humans' interests. Therefore, ephemerality seems to be a limitation of our approach.

7. Conclusions

Social media gradually gain space in our digital lives and substitute traditional means of communication. The availability, anonymity, and the flexible terms that govern online platforms offer a fruitful ground to malicious actors to misuse them. Our work uses a content-based approach to identify automated accounts which demonstrate malicious activity. We showed that compared to traditional bots, social spambots and fake followers are more inclined to post messages with a positive sentimental fingerprint. When we combined metadata and content-based features (including sentimental analysis attributes), we achieved high classification accuracy (0.9389 F1-score). This nominal value is higher compared to the accuracy reported on other state-of-the-art systems and suggests that sentiment analysis features can assist in bot identification. Finally, we showed that after a considerable period of time, our model is able to reliably identify humans without retraining.

Acknowledgements

This work was initiated by Dr. Andriotis while working as an overseas researcher under a Postdoctoral Fellowship of the Japan Society for the Promotion of Science (JSPS).

References

[1] P. Andriotis, A. Takasu, and T. Tryfonas. Smartphone message sentiment analysis. In G. Peterson and S. Shenoi, editors, *Advances in Digital Forensics X*, pages 253–265. Springer Berlin Heidelberg, 2014. 2

[2] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida. Detecting spammers on twitter. In *Collaboration, electronic messaging, anti-abuse and spam conference (CEAS)*, volume 6, page 12, 2010. 1

[3] A. Bessi, M. Coletto, G. A. Davidescu, A. Scala, G. Caldarelli, and W. Quattrociocchi. Science vs conspiracy: Collective narratives in the age of misinformation. *PloS one*, 10(2):e0118093, 2015. 1

[4] A. Bessi and E. Ferrara. Social bots distort the 2016 u.s. presidential election online discussion. *First Monday*, 21(11), 2016. 1

[5] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003. 2

[6] F. Calefato, F. Lanubile, and N. Novielli. How to ask for technical help? evidence-based guidelines for writing questions on stack overflow. *Information and Software Technology*, 94:186 – 207, 2018. 3

[7] S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi. The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 963–972, 2017. 2

[8] G. Danezis and P. Mittal. Sybilinfer: Detecting sybil nodes using social networks. In *NDSS*, pages 1–15. San Diego, CA, 2009. 1

[9] M. M. Danilak. langdetect 1.0.7, 2016. 2

[10] C. A. Davis, O. Varol, E. Ferrara, A. Flammini, and F. Menczer. Botornot: A system to evaluate social bots. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 273–274. International World Wide Web Conferences Steering Committee, 2016. 6

[11] J. P. Dickerson, V. Kagan, and V. S. Subrahmanian. Using sentiment to detect bots on twitter: Are humans more opinionated than bots? In *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)*, pages 620–627, Aug 2014. 2

[12] M. Egele, G. Stringhini, C. Kruegel, and G. Vigna. Compa: Detecting compromised accounts on social networks. In *NDSS*, 2013. 1

[13] M. Egele, G. Stringhini, C. Kruegel, and G. Vigna. Towards detecting compromised accounts on social networks. *IEEE Transactions on Dependable and Secure Computing*, 14(4):447–460, July 2017. 2

[14] E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini. The rise of social bots. *Commun. ACM*, 59(7):96–104, June 2016. 1

[15] A. Garas, D. Garcia, M. Skowron, and F. Schweitzer. Emotional persistence in online chatting communities. *Scientific Reports*, 2:402, 2012. 3

[16] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741, Nov 1984. 2

[17] Z. Gilani, R. Farahbakhsh, G. Tyson, L. Wang, and J. Crowcroft. Of bots and humans (on twitter). In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, ASONAM '17, pages 349–354, New York, NY, USA, 2017. ACM. 6

[18] Z. Gilani, E. Kochmar, and J. Crowcroft. Classification of twitter accounts into automated agents and human users. In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, ASONAM '17, pages 489–496, New York, NY, USA, 2017. ACM. 1, 6

[19] C. Grier, K. Thomas, V. Paxson, and M. Zhang. @spam: The underground on 140 characters or less. In *Proceedings of the 17th ACM Conference on Computer and Communications Security*, CCS '10, pages 27–37, New York, NY, USA, 2010. ACM. 1

[20] L. Hong and B. D. Davison. Empirical study of topic modeling in twitter. In *Proceedings of the First Workshop on Social Media Analytics*, SOMA '10, pages 80–88, New York, NY, USA, 2010. ACM. 3

[21] K. Lee, B. Eoff, and J. Caverlee. Seven months with the devils: A long-term study of content polluters on twitter, 2011. 2, 3

[22] J. Li, C. Cardie, and S. Li. Topicspam: a topic-model based approach for spam detection. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 217–221, 2013. 2

[23] L. Liu, Y. Lu, Y. Luo, R. Zhang, L. Itti, and J. Lu. Detecting "smart" spammers on social network: A topic model approach. In *Proceedings of the NAACL Student Research Workshop*, pages 45–50, San Diego, California, June 2016. Association for Computational Linguistics. 2, 3, 4, 5, 6

[24] Y. Liu, B. Wu, B. Wang, and G. Li. Sdhm: A hybrid model for spammer detection in weibo. In *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)*, pages 942–947, Aug 2014. 2

[25] C. Livas, K. Delli, and N. Pandis. "my invisalign experience": content, metrics and comment sentiment analysis of the most popular patient testimonials on youtube. *Progress in Orthodontics*, 19(1):3, Jan 2018. 3

[26] A. K. McCallum. Mallet: A machine learning for language toolkit. 2002. 2

[27] A. Mislove, S. Lehmann, Y.-Y. Ahn, J.-P. Onnela, and J. Rosenquist. Understanding the demographics of twitter users, 2011. 3

[28] F. Morstatter, L. Wu, T. H. Nazer, K. M. Carley, and H. Liu. A new approach to bot detection: Striking the balance between precision and recall. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 533–540, Aug 2016. 2

[29] S. Nilizadeh, F. Labrèche, A. Sedighian, A. Zand, J. Fernandez, C. Kruegel, G. Stringhini, and G. Vigna. Poised: Spotting twitter spam off the beaten paths. In *Proceedings of the*

2017 ACM SIGSAC Conference on Computer and Communications Security, CCS '17, pages 1159–1174, New York, NY, USA, 2017. ACM. 2, 3, 6

[30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 4

[31] J. Ratkiewicz, M. Conover, M. Meiss, B. Gonçalves, A. Flammini, and F. Menczer. Detecting and tracking political abuse in social media, 2011. 2

[32] R. Řehůřek and P. Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. http://is.muni.cz/publication/884893/en. 3

[33] N. Shuyo. Language detection library for java, 2010. 2

[34] Statista. Most popular social networks worldwide as of April 2018, ranked by number of active users (in millions). https://bit.ly/2ddRJHi, 2018. Accessed: 2018-06-26. 1

[35] G. Stringhini, C. Kruegel, and G. Vigna. Detecting spammers on social networks. In *Proceedings of the 26th Annual Computer Security Applications Conference*, ACSAC '10, pages 1–9, New York, NY, USA, 2010. ACM. 1, 2

[36] M. Thelwall, K. Buckley, and G. Paltoglou. Sentiment strength detection for the social web. *Journal of the American Society for Information Science and Technology*, 63(1):163–173, 2012. 2, 3

[37] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song. Design and evaluation of a real-time url spam filtering service. In *2011 IEEE Symposium on Security and Privacy*, pages 447–462, May 2011. 1

[38] O. Thonnard and M. Dacier. A strategic analysis of spam botnets operations. In *Proceedings of the 8th Annual Collaboration, Electronic Messaging, Anti-Abuse and Spam Conference*, CEAS '11, pages 162–171, New York, NY, USA, 2011. ACM. 2

[39] Twitter Developer. Docs. https://developer.twitter.com/en/docs, 2018. Accessed: 2018-06-26. 1, 6

[40] Twitter Developer Platform. GET statuses/user_timeline. https://developer.twitter.com/en/docs/tweets/timelines/api-reference/get-statuses-user_timeline, 2018. Accessed: 2018-06-26. 4

[41] O. Varol, E. Ferrara, C. Davis, F. Menczer, and A. Flammini. Online human-bot interactions: Detection, estimation, and characterization, 2017. 1, 2, 3, 6

[42] A. H. Wang. Detecting spam bots in online social networking sites: A machine learning approach. In S. Foresti and S. Jajodia, editors, *Data and Applications Security and Privacy XXIV*, pages 335–342. Springer Berlin Heidelberg, 2010. 2

[43] G. Wang, T. Konolige, C. Wilson, X. Wang, H. Zheng, and B. Y. Zhao. You are how you click: Clickstream analysis for sybil detection. In *USENIX Security Symposium*, volume 9, pages 1–008, 2013. 1

[44] W. Wei, K. Joseph, H. Liu, and K. M. Carley. The fragility of twitter social networks against suspended users. In *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 9–16, Aug 2015. 2, 3

[45] W. Wei, K. Joseph, H. Liu, and K. M. Carley. Exploring characteristics of suspended users and network stability on twitter. *Social Network Analysis and Mining*, 6(1):51, Jul 2016. 2, 3, 5

[46] Y. Xie, F. Yu, K. Achan, R. Panigrahy, G. Hulten, and I. Osipkov. Spamming botnets: Signatures and characteristics. *SIGCOMM Comput. Commun. Rev.*, 38(4):171–182, Aug. 2008. 2

[47] C. Yang, R. Harkreader, and G. Gu. Empirical evaluation and new design for fighting evolving twitter spammers. *IEEE Transactions on Information Forensics and Security*, 8(8):1280–1293, Aug 2013. 2, 6

[48] C. Yang, R. Harkreader, J. Zhang, S. Shin, and G. Gu. Analyzing spammers' social networks for fun and profit: A case study of cyber criminal ecosystem on twitter. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 71–80, New York, NY, USA, 2012. ACM. 2

[49] C. Yang, R. C. Harkreader, and G. Gu. Die free or live hard? empirical evaluation and new design for fighting evolving twitter spammers. In R. Sommer, D. Balzarotti, and G. Maier, editors, *Recent Advances in Intrusion Detection*, pages 318–337, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. 2

[50] C. Yang, J. Zhang, and G. Gu. A taste of tweets: Reverse engineering twitter spammers. In *Proceedings of the 30th Annual Computer Security Applications Conference*, ACSAC '14, pages 86–95, New York, NY, USA, 2014. ACM. 2

[51] S. Yardi, D. Romero, G. Schoenebeck, and D. Boyd. Detecting spam in a twitter network. *First Monday*, 15(1), 2009. 1

[52] X. Zheng, J. Wang, F. Jie, and L. Li. Two phase based spammer detection in weibo. In *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, pages 932–939, Nov 2015. 2