Making Twitter Safer: Uncovering Social-Bot on Twitter through User's Content Approach

Nigo Sumaila[1] Shoko Wakamiya[2]
Eiji Aramaki[3]

Several studies have shown that social bots might have impacts on various important areas of our society such as influencing the outcome of elections, the economy, or creating panic in time of crisis. Consequently, there is a growing need to develop and strengthen defense mechanism against these entities. So far, existing methods rely on users' global information, such as profile information, network-related, and from the text content only syntactic information have been used. In this paper, we propose a defense mechanism against social bots on Twitter, a neural network model that incorporates metadata features and semantic information, pairwise tweet similarities and n-gram features from a user's content to differentiate genuine users from social bots. Our model outperforms baseline systems in three publicly available datasets.

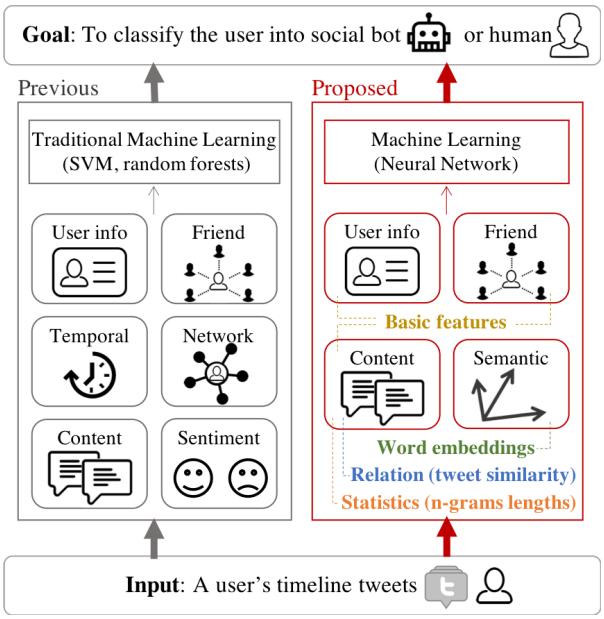


Fig. 1: Overview of the proposed method against previous methods based on many complex features. Some of the features require much time to obtain. Especially network and friend features are inconsistent. In contrast, the proposed method focused mainly on text-oriented features: semantic and content features, which are stable and easy to obtain.

1 Introduction

A recent study [3] suggests that social bots (social media accounts controlled algorithmically by software to generate content and interact with other users, automatically) may have played an essential role in the 2016 U.S presidential election. An analysis of election-related tweets showed that bots produced near one-fifth of the entire conversation. Earlier, during the 2010 U.S midterm election, social bots were used to support some candidates and defame their opponents [30]. These kinds of incidents are not particular only to U.S elections' arena; similar patterns have also been observed in other countries as well [13, 18, 31]. Outside of politics, there are also several reported impacts of social bots. For instance, bots have been associated with causing panic during a time of crisis by disseminating false and unverified information [35]; affecting the stock market by generating a vivid discussion about a company which created an apparent public interest [14].

As a result of the widespread use of social media services, several studies concerned with correlating real-life events with observations made from social media data have been published. From detecting earthquakes [12,32], election outcomes [20,25] to disease outbreaks [2,8,34]. Additionally, [36] estimation places social bot population on Twitter between 9% to 15% of the total active users. Therefore, contents generated by bots might undermine useful predictions made from SNS data. Consequently, there is a growing need to develop and strengthen defences against these entities. However, detecting bot is one of the most daunting tasks in social media analysis [13]. The challenge is understanding what advanced bot can do. Bot have evolved from easily detectable ones which perform mainly one type of activity, such as posting or re-tweeting automatically, to increasingly sophisticated ones. Now the boundary between what constitutes human-like and bot-like behaviour has become fuzzier [14].

Most of the previous work on bot detection fall into two perspectives. First, network-based methods in which the decision is drawn from the users' network analysis, with the assumption that social bots tend to be mostly connected to other bots [6, 39]. However, recent works [4, 29] show that advanced social bots may successfully be connected to humans, making them impossible to be detected solely on network-based assessments. Second, machine learning schemes are trained with language-based, and other features (extracted from users' tweets and metadata) to capture users' behavioural patterns to differentiate real users from social bots. However, language-based features used in previous work were mostly syntactic inherent features, and there is still much to be explored in semantic features.

This paper proposes a defence mechanism against social bots on social network service, Twitter, that leverages nuances in semantic features from a user's timeline tweets combined with tweet similarity, n-gram features and metadata-based features. Fig. 1 illustrates an overview of the proposed method against previous methods based on many sophisticated features. Some of the features require much time to obtain. Especially network and friend features are inconsistent. In contrast, the proposed method focused mainly on text-oriented features: semantic and content features, which are stable and easy to obtain, combined with users' metadata features. The contributions of the paper are summarized as follows:

- We propose a bot detection focused on semantic and content features, which are simple statistics-based features to overhead computational complexity.
- We construct CNN (Convolutional Neural Network) models that incorporate semantic features together with tweet similarity, n-gram features and metadata-based features.
- The experiments conducted with publicly available datasets show that the proposed model outperforms baseline models.

[1] Non member Nara Institute of Science and Technology
nsumaila(at)gmail.com
[2] Regular member Nara Institute of Science and Technology
wakamiya(at)is.naist.jp
[3] Regular member Nara Institute of Science and Technology
aramaki(at)is.naist.jp

2 Related Work

2.1 Bot Detection

Since the early days of social media, several studies on automatic identification of non-genuine users on social media network have been conducted [11, 39]. Most of the work done in this area fall into two categories: network-based and feature-based decision.

On network-based bot detection, the decision is based on social graphs analysis. For example, Sybil*[1] Guard [39] assumes that non-Sybil region, region bounding human-to-human-established trust relation, is fast mixing and Sybil admission to the region is based on admission control protocol. SybilRank [6] assumes that the connection between users is based upon the trustworthiness among them. Consequently, Sybil accounts show less connectivity to real users and more to other Sybils to appear trustworthy. However, sophisticated bots might gain the trust of real users [4, 29], and to be able to penetrate their communities, making them impossible to be spotted solely by network-based assessments through trust assumption [14]. Therefore, other aspects should be considered.

On feature-based detections, machine learning schemes are trained to capture users' behavioral patterns and metadata in order to differentiate real users from social bots. Wang et al. [37] and Alarifi et al. [1] focus on the detection of spam tweets, which optimizes the amount of data that needs to be gathered by relying only on tweet-inherent features. Varol et al. [36] leverage more than one thousand features distributed in six categories (user-based, friend, network, temporal, content, and sentiment features) obtained from users' metadata, mentions and timeline tweets. Clark et al. [10] shows that natural language text features are effective at separating real users from social bots. However, content and language features extracted from tweets text in the previous works [10, 36, 37] were mostly syntactic inherent features.

2.2 Chat Bot

In Natural Language Processing (NLP) context, the main goal is to realize natural conversation instead of bot detection. Most current bot engines are designed to reply to user utterances based on existing utterance-response pairs [38]. In this framework, to capture the relevance between utterance and response is fundamental. Then, the relevance is calculated by the framework's ability to retrieve the most relevant pairs from the conversation database (retrieval approach) or generate the response to the given utterance (generative approach). Although mostly the target relevance is limited to the short scope, usually, a single utterance-response pair, the proposed method handles another broader significance of the entire tweets (Section 3.2).

3 Model

We formulate the task of automatically identifying non-genuine users on social media as follows: given a user u's posts (timeline tweets) $X_u = \{x_1, x_2, ..., x_n\}$, our goal is to find $Y \rightarrow \{0, 1\} = \{human, bot\}$ for the user such that:

$$Y^* = \arg\max_Y P(Y|X_u)$$

where P is the conditional probability of Y given X_u.

Representation of our model is illustrated in Fig. 2. This model is a slight alteration of a shallow CNN for sentence classification tasks [21, 40] to accommodate our features; basic feature, word embedding, pairwise tweet similarity, and n-gram lengths. Given

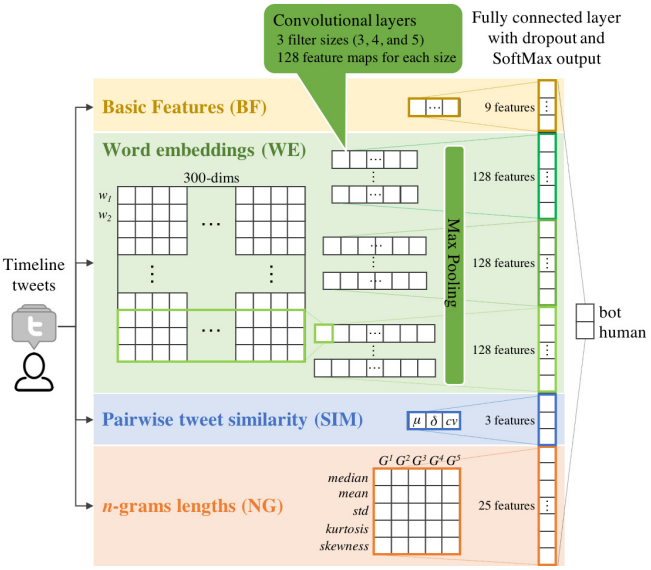


Fig. 2: Visual illustration of the proposed model

a sequence of n tweets X_u from a given user u, we apply it in four different parts. First, basic features are computed (Section 3.1); second, we compute word embedding from the concatenation of all tweets (Section 3.2), then pairwise tweet similarity from all tweets (Section 3.3), and lastly, we compute several statistics about lengths of different classes of n-gram (Section 3.4).

The word embedding layer is followed by a convolution layer comprised of several filters of different sizes (3, 4 and 5), but with the same width as the dimension of the embedding vectors. Every filter performs convolution over the embedding matrix, producing a feature map. Next, we run the feature map through an element-wise non-linear transformation, using Rectified Linear Unit (ReLU) [23]. Then, applying 1-max pooling [5], the maximum value is extracted from each feature map. All extracted values from feature map are concatenated together with features from basic, pairwise tweet similarity, and n-grams lengths. We then feed all features to a full connected SoftMax layer for final output. Dropout [22] is used as a mean of regularization.

This study aims to show the contribution of semantic features in spotting bots when combined with basic features, tweet similarity, and n-gram features obtained from users' tweets.

3.1 Basic Features

Basic features mostly comprise those that are extracted from users' metadata and some from tweets. For a given user u, we extract 9 basic features as follow: number of followers (users that follow u); number of following (users that u follows); ratio between the number of followers and the sum of the number of followers and following; number of tweets authored by u; age of u's account in years; length of u's profile description; length of u's screen name; average number of URL and average number of hashtags (both relatively to the number of analyzed tweets). Fig. 3 shows the distribution of the number of followings relative to URL average (a) and relative to the length of profile description (b) from honeypot dataset (see Section 4.1). We can observe that most genuine accounts, those belonging to real people, followed fewer accounts and have URL average below 2, in contrast, bots have a higher number of followings. Features extracted from users' metadata have been successfully applied to classify users on Twitter [9, 36, 37]. However, advanced bots can camouflage human-like metadata information which decreases the

*[1] Social bots are also known as Sybils

*[1] We interchangeably use human and genuine to refer real users as opposed to social bots.

effectiveness of these features. Thus, we combine basic features with much more sophisticated features to counter bots' deceptive tactics.

3.2 Word Embedding

Word embedding models enable us to represent a text in a dense vector of real numbers, one per word in the vocabulary. On the contrary to vector space models [33], word embedding is built on the notion of distributional hypothesis [16], which assumes that words that are used and occur in the same context are semantically related to each other. Thus, word embedding entails efficiently encoding semantic information of a text or document. Therefore, we take $X'_u = x_1 + x_2 + ...x_n$ as a concatenation of n tweets of a given user u and produce word embedding of 300 dimension that capture semantic features from the user's content.

3.3 Pairwise Tweet Similarity

A recent study [10] has shown that content from pure genuine users tends to be very dissimilar on average compared to content produced by automatons. Since we assume that bots generate tweets with similar structure and minor modifications from one tweet to another, we design the feature to capture the tweet similarity based on that assumption.

We introduce three features that enable us to quantify the degree of similarity of a particular set of pairs of tweets from a given user u: mean of pairwise tweet similarity (μ_u), standard deviation of pairwise tweet similarity (δ_u), and coefficient of variation (cv_u) of pairwise tweet similarity. Furthermore, the Jaro Similarity metric [19] computes the edit distance to quantify how two sentences are dissimilar by counting the minimum number of transposition needed to transform one sentence into the other. Therefore, we apply the metric to compute the similarity between a pair of tweets, $Sim(x_1, x_2)$, as follow:

$$Sim(x_1, x_2) = \begin{cases} 0 & if\ m = 0 \\ \frac{1}{3}\left(\frac{m}{|x_1|} + \frac{m}{|x_2|} + \frac{m - t_r}{m}\right) & otherwise \end{cases}$$

Being $|x|$ the number of characters in a tweet x, m the number of matching characters between two tweets (x_1, x_2) and t_r the number of transpositions needed to change one tweet to another.

The mean μ_u of all pairwise tweet similarities on a sample of n tweets of a given user u is calculated by:

$$\mu_u = \frac{2}{n(n-1)} \sum_{x_i, x_j \in X_u} Sim(x_i, x_j)$$

Being δ_u the standard deviation of pairwise tweet similarity on a sample of n tweets of a given user u, where:

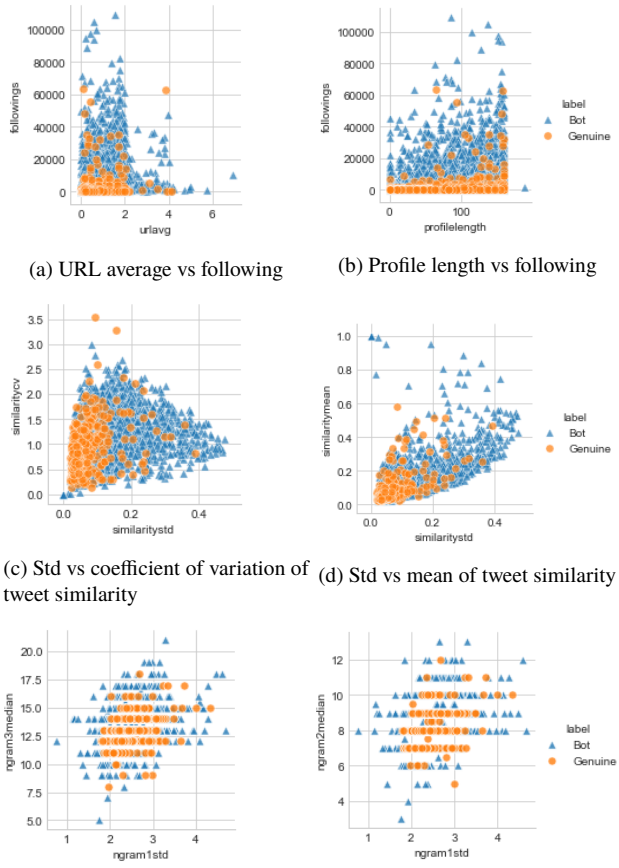
$$\delta_u = \sqrt{\frac{2 \sum_{x_i, x_j \in X_u} (Sim(x_i, x_j) - \mu)^2}{n(n-1)}}$$

Then, the coefficient of variation of pairwise tweet similarity cv_u of a given user u is calculated as $cv_u = \delta_u / \mu_u$.

Fig. 3(c) and (d) illustrate the distribution of the three similarity features. In general bot accounts have very similar content compared to human accounts, but with also high standard deviation. However, similar to Fig. 3(a) and (b), Fig. 3(c) and (d) show that some bots are clustered together or close to genuine accounts. Therefore, pairwise tweet similarity or basic features alone are not enough to separate bot accounts from genuine accounts in some instances.

3.4 n-gram Lengths

n-gram (G^n) is a contiguous sequence of n items from a given sample of text. n-gram models are used over a broad range of tasks in NLP such as text categorization [7], machine translation [15, 26], and speech recognition [17]. In social bot detection task, Wang et al. [37] used a combination of term frequency (tf) and term frequency times inverse document frequency (tf-idf) of unigram (1-gram), bi-gram (2-gram), and tri-gram (3-gram). They achieved their best results by combining user features (e.g, length of profile name, length of profile description, etc) with n-gram features which showed to be very effective. However, computing tf and tf-idf can be computationally expensive on a large dataset. Therefore, we take a different approach by computing several statistics from the number of characters in n-grams (length of n-gram).



(a) URL average vs following
(b) Profile length vs following
(c) Std vs coefficient of variation of tweet similarity
(d) Std vs mean of tweet similarity
(e) Std of 1-gram length vs median of 3-gram length
(f) Std of 1-gram length vs median of 2-gram length

Fig. 3: Distribution of some basic, pairwise tweet similarity, and n-gram features from a sample of $DATA_{HP}$

We use word-based n-gram, e.g., for the given sentence "<user> it should be a good time !", we compute the following: 1-gram (G^1) ={ *"<user>", "it", "should", "be", "a", "good", "time", "!"*}; 2-gram (G^2) = {*"<user> it", "it should", "should be", "be a", "a good", "good time", "time !"*}. Being $|G^n| = \{len(g_1), len(g_2), ..., len(g_k)\}$ a sequence of lengths of n-gram of class n (we compute n-gram of $n = 1$ up to $n = 5$) from a given user's tweets. As for 1-gram (G^1) of the above example, it would be $|G^1| = \{6, 2, 6, 2, 1, 4, 4, 1\}$. From $|G^n|$ the following five statistics are determined: $median(|G^n|)$, $mean(|G^n|)$, $std(|G^n|)$, $kurtosis(|G^n|)$, and $skewness(|G^n|)$, totaling 25 features (five statistics for $|G^1|$ to $|G^5|$). Fig. 3(e) and (f) show the distribution of some n-gram features. Human accounts are clustered in the middle while bot accounts are spread.

4 Experiments

We empirically investigated the proposed models by using two standard datasets from previous works and also tested with a mixed dataset. This chapter initially describes the datasets used to train and evaluate our models. We then give implementation details of our models and baseline models that we compared against our models. At last, we show the results followed by feature importance and error analysis.

4.1 Dataset

We evaluated our models with two publicly available datasets ($DATA_{HP}$ and $DATA_{VAROL}$) and the combination of them $DATA_{MIX}$, respectively.

- $DATA_{HP}$: This dataset is from the social honeypot experiment [24]. It consists of 22000 content polluters (bots) with over 2 million tweets and 19K legitimate users with over 3 million tweets. For our study, we randomly selected 7000 bots and 3000 genuine users, and 200 tweets for each user.
- $DATA_{VAROL}$: This is a manually labeled dataset as in [36]. It consists of about 2500 binary labeled twitter accounts (0 for genuine or human-controlled accounts and 1 for bot-controlled accounts). We ran our crawler in January 2018 and collected 200 tweets from each user's timeline to reduce the crawling time. Note that 200 tweets is the maximum number of tweets per request under the standard Twitter API*[2] as of January 2018. Since some accounts were deleted or had changed their privacy settings from public to private as of time of crawling, the number of successfully crawled accounts reduced to about 2000 (about 600 bots and 1400 humans).
- $DATA_{MIX}$: We combined together the two datasets ($DATA_{HP}$ and $DATA_{VAROL}$), resulting in the dataset with about 12000 users.

We ran a series of tweet anonymization process where we substituted all mentions and links with tokens <user> and <link> from all tweets, respectively. Using an available library*[3], all non-English tweets were removed and only accounts that remained with more than 20 messages were considered for further analysis. For additional data sanitization, the cleaning method from [21] was applied to the datasets.

4.2 Baseline

To evaluate the proposed models, we compared our results with three baseline systems as follows.

- BL_{VAROL} (Varol et al., 2017) [36]: A content-based framework that leverages more than a thousand features distributed in 6 categories: I. user-based features - extracted from user metadata (e.g, screen name length, number of digits in screen name, accounts age, etc); II. friend-based features - obtained from language use, local time, popularity, etc; III. network-based features - extracted from user's network structure; IV. temporal features - derived from user activity, such as average rates of tweet production over various periods and distribution of time intervals between tweets; V. content and language features - extracted from tweet text, mostly syntactic features such as POS tagging usage, statistics about the number of words in a tweet, etc. and VI. sentiment features - obtained from measurement of mood and emotions conveyed in the tweets. They trained a set of machine learning schemes with a subset of 100 features and reported that random forest yielded the best result.

*[2] https://developer.twitter.com/en/docs.html
*[3] https://github.com/saffsd/langid.py

Table 1: Performance of our models on different datasets

Model	$DATA_{VAROL}$	$DATA_{HP}$	$DATA_{MIX}$
BL_{VAROL} [36]	0.890	0.960	0.940
BL_{CLARK} [10]	-	0.960	-
BL_{WANG} [37]	-	0.940 prec	-
	-	0.940 rec	-
BF+WE	0.873	0.984	0.954
BF+WE+SIM	0.889	0.986	0.956
BF+WE+NG	0.884	0.985	0.954
BF+WE+SIM+NG	**0.914**	**0.988**	**0.960**

- values are not available.
Values in AUC except when prec and rec used.
prec and rec are precision and recall, respectively.

Table 2: McNemar chi-squared test with Yates correction of 1.0 of the BF + WE model against: BF + WE + SIM model (a); BF + WE + NG model (b); BA + WE + SIM + NG model (c). On about 3.5K users assembled from the original honeypot dataset. Both models were trained with $DATA_{HP}$.

(a) BF + WE + SIM model

		BF + WE	
		Correct	Incorrect
BF + WE + SIM	Correct	3125	68
	Incorrect	40	194

$p = 0.0090$

(b) BF + WE + NG model

		BF + WE	
		Correct	Incorrect
BF + WE + NG	Correct	3124	68
	Incorrect	41	194

$p = 0.0124$

(c) BF + WE + SIM + NG model

		BF + WE	
		Correct	Incorrect
BF + WE + SIM + NG	Correct	3126	68
	Incorrect	39	194

$p = 0.0065$

- BL_{CLARK} (Clark et al., 2016) [10]: A machine learning approach based on natural language text features to provide the base for identifying non-genuine accounts. They used three content-based features: I. average pairwise tweet dissimilarity; II. word introduction rate decay parameter and III. average number of URLs per tweets;
- BL_{WANG} (Wang et al., 2015) [37]: A machine learning approach that focuses on optimizing the amount of data that needs to be gathered by relying only on tweet-inherent features. They applied three feature categories: I. user features - similar of those used in BL_{VAROL}; II. content features including n-gram based features such as term frequency (tf) and term frequency times inverse document frequency (tf-idf) of unigram (1-gram), bi-gram (2-gram), and tri-gram (3-gram), and III. sentiment features such as automatically and manually created sentiment lexicons from the tweet text.

Table 3: Excerpt of users' tweets with their respective predicted and gold standard (denoted Gold) labels. The first three users are from $DATA_{HP}$ and the latter three users are from $DATA_{VAROL}$.

User ID	Tweet ID	Tweets	Predicted	Gold
u_1	t_{11}	we do not plant knowledge when young, it will give us no shade	human	bot
	t_{12}	he was posting him up user user the cavs didn't have anybody that could help lebron		
u_2	t_{21}	why can't you just be straightforward ? <USER> find joy in the ordinary	bot	bot
	t_{22}	i have so many thoughts <URL> firstly, <USER> is great wish he could do all of		
u_3	t_{31}	hello <USER> we have arrived ! ! ! ! <USER> friday fun fact of the day a ford truck is sold every 41 seconds ford150 <USER> the new chrysler	human	human
	t_{32}	right amp center last spring concert so bittersweet qingatw if you can do high school		
u_4	t_{41}	RT <USER>: #NowPlaying on #Spotify <USER> "One Night" <URL>...	bot	human
	t_{42}	RT <USER>: #NowPlaying on #Spotify Ayron Michael "One Night" <URL> <URL>...		
u_5	t_{51}	<USER> why is dengue spray used in presence of students in school	bot	bot
	t_{52}	afer attock students at jehlum fainted due to effect of den <USER>		
u_6	t_{61}	i finally get a saturday morning to sleep in and i'm awake at 8 am	human	human
	t_{62}	<USER> mike i can't believe it sully oh mike mike i'm on a t shirt		

4.3 Implementation Setup

Our model is implemented in tensorflow on top of publicly available*[4] implementation of CNN for sentence classification as presented by [21].

4.3.1 Models' Settings

To initialize our word embedding of dimension 300, word2vec pre-trained embeddings [28] were used. For convolutions, we set the number of filters to 128 for each filter-size of 3, 4, and 5. We applied 'Dropout' to the input to the penultimate layer with a probability of 0.5. Optimization is performed using a stochastic gradient (SGD) with an initial learning rate of 0.005 and 0.0001 for early stopping.

4.3.2 Models' Variants

Four different variations of the proposed models, starting from one layer CNN with word embeddings [21] combined with basic features (**BF + WE model**) were implemented. To the **BF + WE model** either/both pairwise tweet similarity (**SIM**) (explained in Section 3.3) or/and n-gram lengths (**NG**) (described in Section 3.4), creating **BF + WE + SIM model, BF + WE + NG model**, and **BF + WE + SIM + NG model**, respectively. It took up to 3 days to finish a 5-fold cross validation.

4.4 Results

Table 1 illustrates the results of our models' assessments through a 5-fold cross validation.

$DATA_{HP}$ and $DATA_{MIX}$: The BF + WE model, which is just one layer CNN with pre-trained word embeddings from users' timeline tweets plus basic features, performs well on $DATA_{HP}$ and $DATA_{MIX}$ compared to the baseline systems. As we expected, adding more features (SIM, NG, and SIM + NG) to the BF + WE model improves performance significantly over the three datasets. This suggests that these features have comparatively low correlation among them. We achieved our best results on all datasets when combining all features, which produces BF + WE + SIM + NG model (see Fig. 2).

$DATA_{VAROL}$: Despite not achieving almost-ceiling result like on $DATA_{HP}$, our models performed reasonably well on $DATA_{VAROL}$ too, yielding the state-of-the-art result as well. It is important to state that this dataset is more recent compared to $DATA_{HP}$, and possibly it contains more advanced bots. Furthermore, it has only 2000 users and of those 600 are bots which might not have been enough data to train our models to understand underlying differences between bots and human users. In summary, all our models performed well in all datasets, and the model that combines all features outperformed the three baselines.

4.5 Feature Contribution

To understand better the improvement of performance over the BF + WE model when adding more features (see Section 4.4), we employed McNemar test for paired nominal data [27]. This test is appropriate for binary classification tasks. Since we compare the results of the algorithms when applied to the same datasets.

We assembled new data of about 3500 users from the original honeypot dataset completely exclusive with $DATA_{HP}$. We next tested all models with this dataset and compared the models' outcomes to those of the BF + WE model. Using McNemar chi-square test with one degree of freedom under the null hypothesis ($p = 0.05$) that the models have a negligible decrease of error rate, i.e., it would be determined that there is no significant performance improvement from the BF + WE model if the p value from the test is equal or greater than that of the null hypothesis.

Table 2(a) shows the McNemar chi-square test with Yates correction of 1.0 of the BF + WE model against the BF + WE + SIM model. The p value of the test is equal to 0.009 (less than the p value associated with the null hypothesis), which proves that adding pairwise tweet similarity features; indeed, we gain evident performance improvement. Table 2(b) presents the p value of the test against the BF + WE + NG model is equal to 0.0124, suggesting the significance to consider pairwise n-grams lengths features with the BF + WE model. Similarly, we can interpret the result of the BF + WE + SIM + NG model, as shown in Table 2(c).

Analogous to what we have observed in Section 4.4, Table 2(c) shows that the most significant performance improvement from the BF + WE model is gained when applying the BF + WE + SIM + NG model which combined all the features, and produced the lowest p value among all the three McNemar tests.

4.6 Error Analysis

We conducted some empirical analyses to gain more insights into our model outputs. As stated earlier, in recent years, social bots have become sophisticated enough to generate human-like content [14]. Thus, discriminating bots from human users is not a straightforward task. However, from the analysis of our models' inputs-outputs, we observed that in general, our models performed well even in the presence of non-obvious bots. Table 3 shows a sample of our model output on the two datasets, $DATA_{VAROL}$ and $DATA_{HP}$.

*[4] https://github.com/dennybritz/cnn-text-classification-tf

4.6.1 False Positive

Our models failed to correctly classify users labelled as human but exhibited automated behaviour, bot-like behaviour. Some of these accounts belonged to human users but most of their content was generated by connected applications such as Spotify or Youtube. We also observed cases of miss labelling on the dataset, e.g., user u_4 on Table 3. Accounts labelled as human/bot, although a double check of their content and profile revealed that they are more likely to belong to the different label.

4.6.2 False Negative

Similar to false positive, our models also triggered false negative for users labelled as a bot. Yet, checking on their tweets, metadata and overall activities showed that these accounts might be human accounts (e.g. user u_1 on Table 3).

5 Conclusion

This paper proposed an approach to classify Twitter users into social bots and human users with a CNN model considering features obtained from their texts and metadata. Given a Twitter user's tweets, the model captures the semantic features through a pre-trained word embedding, finds the content similarity by pairwise tweets comparison and statistics about lengths of various class of n-gram, and extracted features from the user's metadata. Our results showed that pairwise tweet similarity and n-gram features, when combined with semantic features, improve performance over the basic + word embedding model (BF + WE model). Our model outperformed the baseline systems, yielding state-of-the-art results on the three datasets.

In future work, we plan to consider more users information such as network, the temporal patterns of content generation, emotions or mood conveyed by the user's content. We also plan to create a vast new dataset containing more new bots.

Acknowledgments

This study was supported in part by JSPS KAKENHI Grant Numbers JP19K20279 and JP19H04221 and Health and Labor Sciences Research Grant Number H30-shinkougyousei-shitei-004.

References

[1] Abdulrahman Alarifi, Mansour Alsaleh, and AbdulMalik Al-Salman. Twitter turing test: Identifying social machines. *Information Sciences*, 372(Supplement C):332–346, 2016.

[2] Eiji Aramaki, Sachiko Maskawa, and Mizuki Morita. Twitter Catches the Flu: Detecting Influenza Epidemics Using Twitter. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1568–1576, 2011.

[3] Alessandro Bessi and Emilio Ferrara. Social Bots Distort the 2016 US Presidential Election Online Discussion. SSRN Scholarly Paper ID 2982233, Social Science Research Network, 2016.

[4] Yazan Boshmaf, Ildar Muslukhov, Konstantin Beznosov, and Matei Ripeanu. Design and analysis of a social botnet. *Computer Networks*, 57(2):556–578, 2013.

[5] Y-Lan Boureau, Jean Ponce, and Yann LeCun. A theoretical analysis of feature pooling in visual recognition. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 111–118, 2010.

[6] Qiang Cao, Michael Sirivianos, Xiaowei Yang, and Tiago Pregueiro. Aiding the detection of fake accounts in large scale social online services. In *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, page 15. USENIX Association, 2012.

[7] William B. Cavnar and John M. Trenkle. N-Gram-Based Text Categorization. *Ann arbor mi*, 48113(2):161–175, 1994.

[8] Lauren E. Charles-Smith, Tera L. Reynolds, Mark A. Cameron, Mike Conway, Eric H. Y. Lau, Jennifer M. Olsen, Julie A. Pavlin, Mika Shigematsu, Laura C. Streichert, Katie J. Suda, and Courtney D. Corley. Using Social Media for Actionable Disease Surveillance and Outbreak Management: A Systematic Literature Review. *PLOS ONE*, 10(10):e0139701, 2015.

[9] Despoina Chatzakou, Nicolas Kourtellis, Jeremy Blackburn, Emiliano De Cristofaro, Gianluca Stringhini, and Athena Vakali. Mean Birds: Detecting Aggression and Bullying on Twitter. In *Proceedings of the 2017 ACM on Web Science Conference*, WebSci '17, pages 13–22, 2017.

[10] Eric M. Clark, Jake Ryland Williams, Chris A. Jones, Richard A. Galbraith, Christopher M. Danforth, and Peter Sheridan Dodds. Sifting robotic from organic text: A natural language approach for detecting automation on Twitter. *Journal of Computational Science*, 16(Supplement C):1–7, 2016.

[11] George Danezis and Prateek Mittal. SybilInfer: Detecting Sybil Nodes using Social Networks. In *The Network and Distributed System Security Symposium*, pages 1–15, 2009.

[12] Paul S. Earle, Daniel C. Bowden, and Michelle Guy. Twitter earthquake detection: Earthquake monitoring in a social world. *Annals of Geophysics*, 54(6), 2012.

[13] Emilio Ferrara. Disinformation and Social Bot Operations in the Run Up to the 2017 French Presidential Election. SSRN Scholarly Paper ID 2995809, Social Science Research Network, Rochester, NY, 2017.

[14] Emilio Ferrara, Onur Varol, Clayton A. Davis, Filippo Menczer, and Alessandro Flammini. The rise of social bots. *Commun. ACM*, 59(7), 2016.

[15] Nizar Habash. The use of a atructural n-gram language model in generation-heavy hybrid machine translation. In *Natural Language Generation*, pages 61–69. Springer, 2004.

[16] Zellig S. Harris. Distributional Structure. *WORD*, 10(2-3):146–162, 1954.

[17] Teemu Hirsimäki, Janne Pylkkönen, and Mikko Kurimo. Importance of High-Order N-Gram Models in Morph-Based Speech Recognition. *IEEE Trans. Audio, Speech & Language Processing*, 17(4):724–732, 2009.

[18] Philip N. Howard and Bence Kollanyi. Bots, #strongerin, and #brexit: Computational propaganda during the UK-EU referendum. SSRN Scholarly Paper ID 2798311, Social Science Research Network, 2016.

[19] Matthew A Jaro. Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. *Journal of the American Statistical Association*, 84(406):414–420, 1989.

[20] V. Kagan, A. Stevens, and V. Subrahmanian. Using Twitter Sentiment to Forecast the 2013 Pakistani Election and the 2014 Indian Election. *IEEE Intelligent Systems*, 30(1):2–5, 2015.

[21] Yoon Kim. Convolutional Neural Networks for Sentence Classification. *arXiv:1408.5882 [cs]*, August 2014. arXiv: 1408.5882.

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, May 2017.

[24] Kyumin Lee, Brian David Eoff, and James Caverlee. Seven months with the devils: A long-term study of content polluters on twitter. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, pages 185–192, 2011.

[25] T. Mahmood, T. Iqbal, F. Amin, W. Lohanna, and A. Mustafa. Mining Twitter big data to predict 2013 Pakistan election winner. In *IEEE International Multi Topic Conference*, pages 49–54, 2013.

[26] José B. Marino, Rafael E. Banchs, Josep M. Crego, Adrià de Gispert, Patrik Lambert, José AR Fonollosa, and Marta R. Costa-Jussà. N-Gram-Based Machine Translation. *Computational linguistics*, 32(4):527–549, 2006.

[27] Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.

[28] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and aheir compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, pages 3111–3119, 2013.

[29] A. Paradise, R. Puzis, and A. Shabtai. Anti-Reconnaissance Tools: Detecting Targeted Socialbots. *IEEE Internet Computing*, 18(5):11–19, 2014.

[30] Jacob Ratkiewicz, Michael Conover, Mark R. Meiss, Bruno

Gonçalves, Alessandro Flammini, and Filippo Menczer. Detecting and tracking political abuse in social media. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, pages 297–304, 2011.

[31] Marco Aurelio Ruediger, Amaro Grassi, Ana Freitas, Andressa Contarato, Carolina Taboada, Danilo Carvalho, Humberto Ferreira, Lucas Roberto da Silva, Pedro Lenhard, Rachel Bastos, and Thomas Traumann. Robôs, redes sociais e política no Brasil: Estudo sobre interferências ilegítimas no debate público na web, riscos à democracia e processo eleitoral de 2018. Report, FGV DAPP, 2017.

[32] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 851–860, 2010.

[33] Gerard Salton, Anita Wong, and Chung-Shu Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[34] Ramanathan Sugumaran and Jonathan Voss. Real-time Spatio-temporal Analysis of West Nile Virus Using Twitter Data. In *Proceedings of the 3rd International Conference on Computing for Geospatial Research and Applications*, COM.Geo '12, pages 39:1–39:2, 2012.

[35] Daniel Tobert, Arvind von Keudell, and Edward K. Rodriguez. Lessons From the Boston Marathon Bombing: An Orthopaedic Perspective on Preparing for High-Volume Trauma in an Urban Academic Center. *Journal of Orthopaedic Trauma*, 29:S7, 2015.

[36] Onur Varol, Emilio Ferrara, Clayton A. Davis, Filippo Menczer, and Alessandro Flammini. Online Human-Bot Interactions: Detection, Estimation, and Characterization. *arXiv:1703.03107 [cs]*, 2017. arXiv: 1703.03107.

[37] Bo Wang, Arkaitz Zubiaga, Maria Liakata, and Rob Procter. Making the Most of Tweet-Inherent Features for Social Spam Detection on Twitter. *arXiv:1503.07405 [cs]*, 2015. arXiv: 1503.07405.

[38] Zhao Yan, Nan Duan, Junwei Bao, Peng Chen, Ming Zhou, Zhoujun Li, and Jianshe Zhou. Docchat: An information retrieval approach for chatbot engines using unstructured documents. In *Proceedings of the 54th Annual Meeting of the Association For Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 516–525, 2016.

[39] Haifeng Yu, Michael Kaminsky, Phillip B. Gibbons, and Abraham Flaxman. SybilGuard: Defending Against Sybil Attacks via Social Networks. In *Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '06, pages 267–278, 2006.

[40] Ye Zhang and Byron Wallace. A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. *arXiv:1510.03820 [cs]*, October 2015. arXiv: 1510.03820.