# Bot detection from a single Tweet

**Ivan Anić, Ivan Bilić, Silvije Škudar**

University of Zagreb, Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb, Croatia
{ivan.anic,ivan.bilic2,silvije.skudar}@fer.hr

**Abstract**

With the advent of automatic content generation, it is becoming increasingly hard to tell whether information was crafted by artificial intelligence or real people. Even though it's always been possible to create harmful content, bots potentiate that capability significantly, and so the task of classifying content as generated by bots becomes vitally important, especially on sites used by many people. We focus on detecting bot-generated content on Twitter, a popular social media site. While most of the previous work combines tweets with account-level metadata, we focus on detecting bots solely from a single tweet's content. We experiment with two main approaches: a shallow classification model with hand-crafted features and a deep model that exploits Twitter GloVe embeddings and BERT sentence embeddings.

## 1. Introduction

Social media has rapidly become an integral part of our lives, and a trusted resource of content. People are used to browsing and absorbing information from social networks on a daily basis. However, by trusting them too much, they can easily be influenced without even being aware. The power of social media bots lies in this subtlety of influencing public opinion, which can be used in harmful ways, such as spreading propaganda, fake news (Shu et al., 2018), increasing exposure to inflammatory content (Stella et al., 2018) and falsehood in general, political campaigns, etc.

Academic studies estimate that up to 15% of Twitter users are automated bot accounts (Varol et al., 2017), which is why the task of detecting bots on social networks has gained significant attention over the past decade. What makes a detection task even harder is that bots don't necessarily need to be fully autonomous, as they can also be supervised by a human. Human supervisors can introduce more variety into bot behaviour mechanisms, decreasing the chance for successful detection. Furthermore, bot creators can also analyze the research being done on the topic, together with the code being published, in order to improve their bots and increase their chances to remain "under the radar". It is for this reason that the ability to recognize bot-generated content merely from a single tweet's content is important (Ferrara et al., 2014).

## 2. Related work

Detecting bots has been researched extensively, especially since the recent 2016 United States presidential election. Much of the work used the Twitter API[1] and combined tweet content with various user-level metadata. *BotOrNot* (Davis et al., 2016) used Random Forests with more than 1,000 features derived from the content itself as well as metadata such as account creation time, friend count, user's geographical and language information, all obtained using the Twitter API. Cai et al. (2017) use a deep network and combine text statistics with tweet timestamps and user's tweet history. Similarly, for their deep model, Kudugunta and Ferrara (2018) also use a dozen account-level features like followers, friends and favourites counts. They also check for background images and default user settings. Dickerson et al. (2014) combine user metadata with tweet an in-depth sentiment analysis.

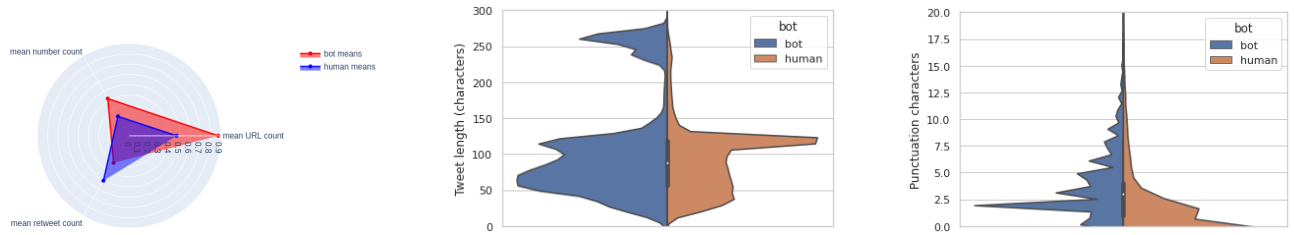
While behavioural data is powerful and has been studied extensively (e.g. Guess et al. (2019)), APIs for obtaining metadata are often inaccessible or limited. Thus, a different, albeit more difficult approach is one where only tweet content is used. Clark et al. (2015) use a classifier and analyze the average URL count per tweet, average pairwise lexical dissimilarity between tweets as well as the word-introduction rate decay over time-ordered tweets. More recently, Oliveira et al. (2019) combine tweet sentiment with tweet characteristic frequencies like URL and mention frequencies.

In a similar vein, our work focuses on classifying individual tweets with no additional metadata. We investigate the impact of features obtained from raw tweet text on performance of a shallow SVM model. Additionally, we tried generating entire tweet embeddings using a BERT (Reimers and Gurevych, 2019) based model, and after noticing significant differences between human and bot embeddings, we passed it through a fully connected layer to see whether this information could yield improvements, described in detail in Section 4.1.

## 3. Dataset

Our dataset consists of 6760 accounts with 100 tweets per account written in English, which totals to 676000 tweets, where 264000 tweets are used for testing, and the remaining 412000 tweets for training. Additionally, of those 412000 tweets, 82400 tweets are used for validation when training a deep model. All the datasets have balanced class distributions. The dataset we used was published for PAN @ CLEF 2019 - Bots and Gender Profiling competition task (Rangel and Rosso, 2019). There is also a Spanish subset of the dataset and all the accounts in the dataset contain gender labels, but this paper is focused only on the bot de-

[1] https://developer.twitter.com/en




(a) Comparison of mean URL, retweet and number counts,

(b) Character count distributions,

(c) Punctuation count distributions.

Figure 1: Comparison of feature values that significantly differed between bot and human content.

tection task using only tweets written in English.

### 3.1. Preprocessing

Special tokens such as URLs and numbers are replaced with special values. Stopwords, non-words and single characters are removed, HTML tags are stripped, text is lowercased and so forth. Other than that, the way tweets are preprocessed differs across models. When using GloVe Twitter embeddings (Pennington et al., 2014), tweets are preprocessed using a slightly modified Python version of the Ruby script given on the GloVe official website, and tokenized using NLTK TweetTokenizer. For BERT embeddings, SpaCy's rule-based sentence segmentizer is used to extract the sentences from tweets.

## 4. Model/our approach

We experiment with different model approaches. First, we use a baseline SVM model that takes hand-crafted features as input. Hand-crafted features are chosen based on observations described in Section 4.1. Next, we employ two different deep models. The first one takes GloVe word vectors pretrained specifically on a Twitter corpus as input and it's composed of LSTM and fully connected output layers. How we use GloVe embeddings is pointed out in Section 4.2. The second model is a sentence transformer that is used to produce and feed tweet embeddings into the fully connected layers. Sentence transformer is not fine-tuned. Section 4.3. explains tweet embeddings.

### 4.1. Hand-crafted features

To gain a better understanding of the differences between bot and human behaviour, we crafted new features from tweet content. For each tweet, 16 statistical features are created. These can be roughly grouped in four categories: word features (WF, e.g. word count, character count), punctuation features (PF, e.g. punctuation count), unique word features (UF, e.g. number of unique words) and special token features (SF, number of retweets, URLs and numbers). Tweets were separated in two groups,s one containing bot tweets, and the other containing human ones. This lets us compare the mean and standard deviation of each feature's values per tweet.

We analyzed each feature separately, and while some differed significantly, many were redundant. Interestingly, Figure 1a shows that bots posted URLs and numbers almost twice more than human users on average, while people retweeted more. Since the dataset is of political nature, higher URL count could be explained by suspecting these posts linked to external political or propaganda sites. Additionally, it seems a number of bots pushed the tweet length near its 280 character maximum, while human users usually remained below 150 characters per tweet, as shown on Figure 1b. From Figure 1c it's visible that there was also more variance in punctuation use in bot content.

Keeping this insight in mind, we scaled the new features so they could later be used as inputs to the SVM model. Even tough we use all 16 features as inputs to the model, we do note that some of them might be redundant, as they do not differ significantly between the bot and human classes.

### 4.2. Word embeddings

We train our deep model on pretrained GloVe Twitter word vectors (Pennington et al., 2014). The tweet preprocessing done in this approach is briefly explained in Section 3.1.. For each tweet, all the tokens are replaced with their corresponding GloVe Twitter embedding. Tokens that have no corresponding embedding are replaced with a random embedding sampled from gaussian distribution with zero mean and unit variance.

### 4.3. Tweet embeddings

Parameter estimation on our dataset gives results that motivate the usage of tweet embeddings. Each user is represented with a particular multivariate gaussian distribution and each tweet embedding is then viewed as a vector sampled from its user's distribution. We can estimate user's variance vector $\boldsymbol{\sigma}$, calculated as $diag(\boldsymbol{\Sigma})$ where $\boldsymbol{\Sigma}$ is a covariance matrix describing the user's distribution. Calculating the mean variance vector for humans and bots separately, and comparing the two vectors yields that humans have higher variance in 742 out of 768 dimensions, meaning that tweet embeddings capture the fact that human behaviour is less predictable. However, the only use of that statistical analysis is to encourage, or at least not discourage the use of tweet-level embeddings, meaning we do not embed the variance information into the models in any way. Regarding statistical significance of those results, t-test results show that for 730 out of 768 embedding dimensions, human tweet embeddings differ from bot generated ones at a significance level of 5%, counteracting the potential ocurrence of a false positive by implementing Bonferroni correction (Bonferroni, 1935).

In this approach, we generate tweet-level embeddings using a pretrained sentence transformer based on BERT-base

Table 1: Classification accuracy and macro F1-score for the hand-crafted feature experiment, GloVe + LSTM experiment and BERT experiment.

| Model | Accuracy | Macro F1-score |
|---|---|---|
| TFIDF + SVM | 0.755 | 0.752 |
| TFIDF + SVM + WF | 0.760 | 0.757 |
| TFIDF + SVM + PF | 0.757 | 0.754 |
| TFIDF + SVM + UF | 0.763 | 0.760 |
| TFIDF + SVM + SF | 0.754 | 0.751 |
| TFIDF + SVM + ALL | **0.770** | **0.768** |
| GloVe + LSTM | 0.806 | 0.788 |
| BERT | **0.812** | **0.792** |

model [2]. Given a sentence, sentence transformer generates a 768-dimensional sentence embedding. Tweet embedding is then constructed by averaging all sentence embeddings that belong to the same tweet. The sentence transformer is not fine-tuned on our data. In contrast to GloVe, tweet embeddings are generated for each tweet in our dataset so in this case there's no need to use the random gaussian embedding.

## 5. Experiments and Results

Having acquired an idea about the difference between content generated by bots and people, we conduct an experiment to empirically show whether extracting statistical information from raw tweet content improves bot detection. To this end, we separate hand-crafted features into four groups as described in Section 4.1.. Our baseline consists of sklearn's unigram TFIDF Vectorizer to convert preprocessed tweets into 10,000 dimensional vectors as inputs for the SVM. We then experiment with concatenating each of the four feature groups to obtained TFIDF vectors separately, as well as combining all of them. SVM is then trained on the entire train dataset and evaluated on the test dataset. The results of these experiments are shown in the upper section of Table 1. Surprisingly, special token features (SF) do not improve the scores despite their frequencies being significantly different, as shown in Figure 1a. However, the combination of all hand-crafted features does improve the scores somewhat.

In GloVe + LSTM experiment, first the text is preprocessed in order to be coherent with the GloVe Twitter pretrained word vectors. Those embeddings are then fed into LSTM whose output is fed to fully connected layers. More about the preprocessing steps and word embeddings can be found in Section 3.1. and Section 4.2. respectively. We use LSTM with four layers. Furthermore, in all experiments except the baseline, we use two fully connected layers. The first one with ReLu and the second, output layer with the Sigmoid activation function which outputs the probability of an input tweet being written by a bot. We observe that for training with different GloVe embedding sizes, changes in performance are negligible.

[2]https://github.com/UKPLab/sentence-transformers

BERT experiment is essentially employing a BERT-based sentence transformer that outputs 768-dimensional sentence embeddings from which tweet embeddings are then constructed. Tweet embeddings are then fed into fully connected layers that are the same as in GloVe + LSTM experiment. See Section 4.3. about tweet embeddings. The results in Table 1 show that BERT embeddings are dominant in bot detection task as well. GloVe embeddings need to be processed by LSTM in order to come close to BERTs performance, whose embeddings are directly fed into the fully connected layers, with no additional processing in between.

## 6. Conclusion

In this paper we presented an overview of the task of detecting social media bots, and proposed a few approaches to classify bots on Twitter based solely on the tweet contents. Although the information to make conclusions upon is in this way very limited, we have successfully shown that text generated by bots is still different enough than that of humans to successfully distinguish bots from humans.

There are several promising directions for future improvements on proposed methodologies, such as experimenting with different network architectures, hyperparameters, as well as combining hand-crafted features with word embeddings. Regarding BERT, word embeddings could be used instead of sentence embeddings, and BERT model fine-tuning could also be carried out during training.

## References


Carlo E Bonferroni. 1935. Il calcolo delle assicurazioni su gruppi di teste. *Studi in onore del professore salvatore ortu carboni*, pages 13–60.

C. Cai, L. Li, and D. Zengi. 2017. Behavior enhanced deep bot detection in social media. In *2017 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pages 128–130.

Eric M. Clark, Jake Ryland Williams, Richard A. Galbraith, Chris A. Jones, Christopher M. Danforth, and Peter Sheridan Dodds. 2015. Sifting robotic from organic text: A natural language approach for detecting automation on twitter. *CoRR*, abs/1505.04342.

Clayton Allen Davis, Onur Varol, Emilio Ferrara, Alessandro Flammini, and Filippo Menczer. 2016. Botornot: A system to evaluate social bots. In *Proceedings of the 25th International Conference Companion on World Wide Web*, WWW '16 Companion, page 273–274, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

John P. Dickerson, Vadim Kagan, and V. S. Subrahmanian. 2014. Using sentiment to detect bots on twitter: Are humans more opinionated than bots? In *Proceedings of the 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '14, page 620–627. IEEE Press.

Emilio Ferrara, Onur Varol, Clayton A. Davis, Filippo Menczer, and Alessandro Flammini. 2014. The rise of social bots. *CoRR*, abs/1407.5225.

Andrew Guess, Jonathan Nagler, and Joshua Tucker. 2019. Less than you think: Prevalence and predictors of fake

news dissemination on facebook. *Science Advances*, 5(1).

Sneha Kudugunta and Emilio Ferrara. 2018. Deep neural networks for bot detection. *CoRR*, abs/1802.04289.

Rodrigo Ribeiro Oliveira, Cláudio Moisés Valiense de Andrade, José Solenir Lima, Conceição Silva, and Almir Moreira da Silva Neto. 2019. Bot and gender identification: Textual analysis of tweets.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Francisco Rangel and Paolo Rosso. 2019. Pan19 author profiling: Bots and gender profiling, February.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11.

Kai Shu, Suhang Wang, and Huan Liu. 2018. Understanding user profiles on social media for fake news detection. In *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 430–435. IEEE.

Massimo Stella, Emilio Ferrara, and Manlio De Domenico. 2018. Bots increase exposure to negative and inflammatory content in online social systems. *Proceedings of the National Academy of Sciences*, 115(49):12435–12440.

Onur Varol, Emilio Ferrara, Clayton A. Davis, Filippo Menczer, and Alessandro Flammini. 2017. Online human-bot interactions: Detection, estimation, and characterization. *CoRR*, abs/1703.03107.