Uncovering Social Media Bots: a Transparency-focused Approach

Eric F. Santos
Federal University of Rio de Janeiro
Rio de Janeiro, Rio de Janeiro
eric.ferreira@ppgi.ufrj.br

Danilo S. Carvalho
Federal University of Rio de Janeiro
Rio de Janeiro, Rio de Janeiro
dsc@ufrj.br

Livia Ruback
Federal University of Rio de Janeiro
Rio de Janeiro, Rio de Janeiro
liviaruback@gmail.com

Jonice Oliveira
Federal University of Rio de Janeiro
Rio de Janeiro, Rio de Janeiro
jonice@dcc.ufrj.br

ABSTRACT

As the Online Social Networks (OSNs) presence continues to grow as a form of mass communication, tensions regarding their usage and perception by different social groups are reaching a turning point. The number of messages that are exchanged between users in these environments are vast and brought a trust problem, where it is difficult to know if the information is from a real person and if what was said is true. Automated users (bots) are part of this issue, as they may be used to spread false and/or harmful messages through an OSN while pretending to be a person. New attempts to automatically identify bots are in constant development, but so are the mechanisms to elude detection. We believe that teaching the user to identify a bot message is an important step in maintaining the credibility of content on social media. In this study, we developed an analysis tool, based on media literacy considerations, that helps the ordinary user to recognize a bot message using only textual features. Instead of simply classifying a user as a bot or human, this tool presents an interpretable reasoning path that helps to educate the user into recognizing suspicious activity. Experimental evaluation is conducted to test the tool's primary effectiveness (classification) and results are presented. The secondary effectiveness (interpretability) is discussed in qualitative terms.

KEYWORDS

online social network; bot message detection; media literacy

ACM Reference Format:
Eric F. Santos, Danilo S. Carvalho, Livia Ruback, and Jonice Oliveira. 2019. Uncovering Social Media Bots: a Transparency-focused Approach. In *Companion Proceedings of the 2019 World Wide Web Conference (WWW '19 Companion), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3308560.3317599

1 INTRODUCTION

Online Social Networks (OSNs) became a world changing trend on communications in the recent years. This kind of media allows different people, from different countries to share their lives and thoughts. The interactions between people on this environment brought a problem of trust, where it is difficult to know if another user is a real person and if what was said is true. Automated users (*bots*) are part of the issue, as they may be used to spread false and/or harmful messages through an OSN at a fast rate, contaminating real users with misinformation.

Not all the automated users are malignant, for instance, there are bots that spread messages from newspapers and weather applications as an automatic service to publish the same news in various channels. However, OSNs often have been dealing with malignant users and their activities. The most common malicious activity is spamming, wherein an automated user (bot user) disseminates content or malware/viruses to users of the social networks [4]. These bots can be used for several proposes, including: (1) advertising; (2) promoting politically oriented views and opinions; (3) promoting financial trends; (4) generating product reviews; (5) spreading malware, spam, and harmful links; (6) influencing search engine results such that particular links are shown first; (7) generating news feeds; and (8) creating an underground marketplace for purchasing social media followers [1].

The increasingly use of bots proliferating their biased messages can have a negative impact on Society, interfering with the people's democratic, civil and behavioral process. As an example we can mention what occurred in the United States of America presidential election of 2016. Studies suggests that so called "fake news" (i.e., false or misleading statements) might have been decisive to the victory of the current president on the election [5], and that such misinformation was spread by bots [16] on OSNs. Twitter, a widely used OSN where users can share short messages with text, images and videos, admitted to having excluded more than 50,000 bot accounts related to fake news propagation in US election [18].

Another problem that contribute to the misinformation dissemination is that regular (human) users often spread messages without checking whether they are true or not. Such users become new channels for misinformation, filtering and directing it according to their own ideological leanings.

Taking these problems into account, we believe that teaching the user to identify a bot message is an important step towards a healthier environment in social media. We conducted a study to analyze textual features taken from bot and human messages, used in previous related studies. We verify the feasibility for fast bot

WWW '19 Companion, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6675-5/19/05.
https://doi.org/10.1145/3308560.3317599

detection while producing a useful guidance for users to identify a bot message. Such guidance is important from a media literacy standpoint, so that users can keep trust and gain more knowledge from their interactions and OSNs can guarantee credibility in the transmitted contents. We also provide a tool that performs a human/bot classification task and presents an interpretable reasoning path that helps educating users into recognizing suspicious activity.

Works approaching bots detection on OSNs typically collect many features, such as message content, network, profile, and others to characterize an automated user. However, combining such features may be resource intensive and also OSN specific. We focus on the text because it is the simplest way for a user to be aware that there may be something wrong with the message. The combination of many characteristics, such as number of followers, number of friends, among others, would make more difficult the literacy process, i.e., for a human to check all possibilities and judge if it was generated automatically.

Our solution is based on Twitter messages. Twitter is microblogging, a service that allows the user to share messages containing up to 140 characters (recently changed to 280), images, and videos. The user can play two roles: follower and followed. As a follower, the user can select another person which have an affinity and track their activities. As a followed, it shares your activities to other users that desire to follow your profile [11]. Recent works indicate a massive use of Twitter around the world. Aslan [3] has shown that there are about 500 million messages sent per day and 100 million daily active users. The message exchanges on Twitter achieved the mark of 500 million in the last year (about 5787 messages every second) [6], and 326 million people use this OSN every day. These numbers point to the platform's profile being focused for fast message creation and dissemination.

Experiments were conducted to evaluate the fitness of the proposed approach to the purposes of *(a)* identifying bots and *(b)* educating the user. The former was tested for classification performance (quantitative) and the latter for interpretability (qualitative).

The remainder of this paper is organized as follows: Section 2 introduces basic concepts and presents related works. Section 3 presents the approach to classify bots and humans and to assist users to distinguish between them. Section 4 details the experiments and discusses the evaluation results. Section 5 presents the guide tool, and Section 6 brings conclusions and final remarks.

2 BASIC CONCEPTS AND RELATED STUDIES

In this section, we summarize the main subjects covered in this work and the related studies in their respective areas: bot detection and media literacy. Bot detection research here focuses mainly into message features, and media literacy studies introduce the importance of this area.

2.1 Bot Detection

Detection of bots in OSNs as a task consists in classifying any given OSN user into bot or non-bot (human). As new approaches for detection are developed, so are bot countermeasures to avoid detection. Therefore, this task has no permanent solution.

Many of the works on bot detection collect many features, such as content, network, profile, and others to characterize an automated user. However, Martinez-Romo and Araujo [13] state that most of these features, such as the number of followers and friends, account creation dates, and others, can not only be easily manipulated by bots but that collecting all this data is a resource-intensive task. The research works presented in this section focus on the bot identification through the message, which is the common resource from most OSNs.

Martinez-Romo and Araujo [13] proposed an approach not to identify bot users, but spam messages. It applied statistical analysis of language to detect a spam message in Twitter trending topics. A statistical language model (SLM) is a probability distribution $P(s)$ over strings s that attempts to reflect how frequently a string s occurs in a sentence. The authors introduced an architecture that collects trending topics from the Twitter API [1], labels the messages, extracts the features, trains a classifier and detects a spam message. The collected trending topics date from 30 April 2012 to 8 May 2012, from English speakers. In order to label spam, messages were selected if the text contained a link. These links were classified using services that provide blacklisted sites. If a link were found in any blacklist, then the message was labeled as spam.

Igawa et. al. [10] studied a wavelet-based approach for account classification using only text messages generated by users in an online social network. This approach worked in conjunction with a new weighting scheme, called Lexicon Based Coefficient Attenuation (LBCA), that serves as input to a classifier algorithm. This research evidenced the low computational costs in identifying which kind of account will be analyzed since only the text is being taken into account. They use information retrieval techniques to analyze text content and conducted to conduct two experiments with Twitter datasets: matching the accounts with humans and bots, and identifying an account as human or bot. They used wavelets to decompose the signals brought by the weighting process and these signals are part of the detection process.

The dataset was collected from Twitter and only messages related to FIFA World Cup 2014 were retrieved, particularly the query "BRASIL" (Brazil, in Portuguese), "COPA" (World Cup, in Portuguese), "COPA2014", manually labeled. It used random forests and multilayer perceptron as classification algorithms and used other weighting schemes. However, the proposed LBCA had the best precision in both experiments using random forest classifier. In a later exploration, it proposed to analyze the behavior of the content produced by bots for improving spam detection activities in online social media.

Other works use textual features to improve their classifier, as those developed by Alsaleh et. al. [2] and Alarifi et. al. [1] that analyzed bot accounts to find the best set of features to be used in the classification step. They include: *i)* number of hashtags per tweet, *ii)* number of times a hashtag has been used, *iii)* number of links, *iv)* whether the profile picture contains a face, or it is the basic Twitter profile picture, *v)* mentions of different users with the same text, and *vi)* number of lists in which the user is listed on, are examples of selected features. With the features set, the authors used four machine learning algorithms for the classification step:

[1] Application Programming Interface

Decision Tree, Random Forest, Support Vector Machines (SVM), and Multilayer Neural Network.

Dickerson et. al. [7] proposed linguistic, network and application of variables to distinguish humans from bots using sentiment features. They presented SentiBot, a sentiment-aware architecture for the Twitter platform. On this architecture, they combined a set of features, such as sentiment features, neighborhood metrics, syntactic metrics, semantic linguistic models, graph-theoretic metrics, among others, collected from Twitter, related to the Indian elections in 2014. The users were manually labeled as bots or humans.

In these works, the main objective is to generate computational tool and methods to detect a bot user or a bot message leaving aside more information. However, they are not focused on assisting users in the task of detecting bots.

2.2 Media Literacy

Media literacy is an area that discusses the ability to access, analyze, evaluate and create messages in various contexts [12]. This area covers how the media is accessed (e.g., by television or internet), how the message is analyzed according to the reader's previous knowledge, how it is evaluated, and how a new message is sent forward.

This area has studies on how a media message need to be sent and how the receptor understands it. The user should be able to analyze and evaluate the media content, pondering the message relevance and confidence.

Fleming [8] conducted a case study at the Journalism School at Stony Brook University to implement a new form to present media literacy in the course to help students to assess the news quality. Through interview analysis, the news literacy form teaches students how to access, evaluate, analyze and appreciate journalism.

In [15], it is presented the main advantages of media literacy: 1) it promotes critical thinking skills to make independence choice, as which media select and how to interpret the information; 2) impacts individuals and society; 3) how to analyze and discuss a media message among others. Based on these advantages, we believe that media literacy is very important in the context of OSNs since this environment allows fast spreading of information. Such information is generated and consumed by its users, therefore they have responsibilities in how they handle with this information.

3 BUILDING THE CLASSIFICATION MODEL

In this section we present the dataset used in our experiment, the steps to prepare the data, and detail the classification process used to identify bots, as well as the method for obtaining the classification decision explanations.

3.1 Dataset

We chosen Twitter as OSN since it has a set of characteristics that favors this kind of research. An example is the hashtag (#), used as a subject mark in each message, so users can direct their messages to a specific theme. Another example is the mention (@), that allows to reply a user message or quote them.

One of the most important features that makes Twitter an useful OSN for study is its powerful API[2], which is well documented and allows a data sample for free download.

For our analysis we select a dataset labeled by Morstatter et. al. [14], that was collected from Twitter in the Arab Spring between 2011 and 2013, using the keywords: #libya, #gaddafi, #benghazi, #brega, #misrata, #nalut, #nafusa, #rhaibat, as well as a geographic bounding box around Libya. Using the Twitter API in 2015 they got account status from each user. Whoever was different from active was considered a bot. They provide a user list which includes the class (bot or human).

3.2 Preprocessing

We use the list of users labeled as users or humans to collect from Twitter all English messages from each user. An obstacle encountered was that some Twitter accounts no longer exist, which decreased the number of bot messages. This caused the dataset to be severely unbalanced towards human users.

To overcome this problem, we decided to select for each bot a group of human users for which messages most closely resemble the bots textual subjects. In this way, the comparison would be ideally done under textual cues that are less related to the topic, since the topic would be the same for both classes. The idea behind such alignment of topics is that by isolating the "topic feature" – a meta feature for the distribution of words in a message – the remaining textual features would be more easily captured by a Machine Learning classifier. On the other hand, such restriction of the word distribution also limits the ability to capture other possible textual features that are also unrelated to the topic. An alternative solution to this problem would be to separate the messages by topic and compare the bot messages with the human ones in all topics but the ones they share. Such solution was, however, not feasible under the time constraints for this study.

In summary, only users posting messages about subjects matching at least one bot were selected. For each bot, all human users were ranked by a relative word frequency score using the following formula:

$$score_{b_i,h_j} = \sum_{i,j} \sum_{k} F_{b_i}(w_k) * F_{h_j}(w_k) \quad | \; w_k \in W = W_{b_i} \cap W_{h_j} \tag{1}$$

where b_i is a bot user, h_j is a human user, $F_u(w)$ is the relative frequency of the word w in user's u messages, given by the quotient of the word count $\#w$ by the most frequent word count $\#w^*$. W_u is the word set of user u. Such score increases with the amount and frequency of shared vocabulary between two users.

The intuition behind the scoring formula is as follows:

- By calculating the relative frequency $F_u(w)$ of word w in user's u messages, a frequency vector $\hat{f}_u$ can be obtained, where each position refers to a single word in the shared vocabulary $W = W_{b_i} \cap W_{h_j}$.
- A measure of alignment between the user's word distributions can then be obtained by taking the dot product of

[2] Application Programming Interface

vectors $\hat{f}_u$, as it grows linearly with the relative frequency product $\hat{f}_{b_i} * \hat{f}_{h_i}$ of each word in the vocabulary.

- $score_{b_i,h_j}$ express the dot product $\langle \hat{f}_{b_i}, \hat{f}_{h_j} \rangle$ between frequency vectors for the bot user b_i and human user h_j, where $b_i \in B$, the set of bot users, and $h_j \in H$, the set of human users.

For each bot, the top n ranked human users were selected so that the number of messages were as close as possible. The selected users (bots and humans) messages constituted a balanced dataset, which was used for this work.

Before this step we had 9348030 messages in total, being 9036790 human messages and 311240 bot messages. The number of bot users is 106, while the number of humans before the preprocessing was 18963. After the preprocessing, we had 295307 human messages and 506 human users, almost 5 human users per bot and the number of messages between the classes balanced.

3.3 Analysis

To proceed with the analysis, we selected textual features to train a decision tree model classifier. Since one of our contributions is a user guide on how to detect a bot message, this approach was selected due to the ease of interpreting the resulting model. The decision tree algorithm implementation is *CART*[3]. The use of only textual features is a limiting factor regarding model accuracy, but provides way of classification that is reproducible by a human user with no further tools, given enough explanation about the path taken in the tree to reach a decision.

The selected textual features are shown in Table 1. We selected some features according the OSN characteristics, such as hashtag and mentions, that are common on Twitter. Part of Speech (POS) features were used to explain the message composition allowing superficial syntactic and semantic analysis. The selected POS were *noun, pronoun, verb, adverb, adjective, preposition, conjunction, numeral and interjection.*

Following the works of Alsaleh et. al. [2] and Alarifi et. al. [1], we used sentiment classification as a feature. The python[4] library *VADER* [9] was used to generate the values: positive, neutral and negative sentiment as 1, 0, -1, respectively, to our classifier.

4 EXPERIMENTS AND THE EXPLANATION METHOD

For evaluating model performance regarding the bot classification task, we implemented a 10-fold cross-validation, to train and test our classifier. To generate an intelligible tree and avoid overtraining, we set the tree maximum depth as 4 and the tree maximum leaf nodes as 7. Table 2 presents the mean and standard deviation from the training and test.

[3]https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTreeClassifier
[4]https://www.python.org/

Table 2: Model Results (10-fold CV)

Index	Mean	Std. Deviation
Accuracy	72%	0.002
Precision	68%	0.002
Recall	80%	0.003
F1	73%	0.002
Area under curve (AUC)	72%	0.002

With the decision tree simplification, some features showed to be more important than others. We found out that the POS-tag features, such as the number of interjection, conjunction, preposition, adverb, numeral and adjective present on the text does not contribute with the classification.
The same occurred for the number of hashtags, punctuation, and mentions. The sentiment classification is not an important feature of this dataset. The features used in the final classifier were the number of uppercase characters, the number of pronouns, the number of verbs and the number of links present in the message.

In the resulting tree, we observed that bot messages do not have much uppercase text (less than one), and the number of words categorized as noun and verb are less than five and one, respectively. As a recent update on Twitter [17], the number of allowed characters in the message was doubled (from 140 to 280), we could observe that messages with more than 140 characters have a considerable probability of being a bot message. Messages with less than 140 characters, but with less than one pronoun and one link in the text may also indicate a bot.

The next step was then describing the decision path in a human-interpretable way for a given message. For this, we traversed the decision path "translating" the attribute names to more understandable descriptions, and used a simple node separation syntax, as the following examples show:

Decision path (single): $qtd\_upper < 1$ is **true** $\Rightarrow$ *bot*
Explanation: See Figure 1

Decision path (all human): $qtd\_upper < 1$ is **false** $\rightarrow$
$qtd\_text \leq 140$ is **true** $\rightarrow$
$pron\_count \leq 1$ is **false** $\Rightarrow$ *human*
Explanation: See Figure 2

Figure 3 illustrates the full decision tree.

A simple qualitative evaluation was designed so that a decision path explanation would be able to be understood by at least 2 people: A function called $interpretability(expl(m))$ were $expl(m)$ is the explanation given by the tool on the decision path taken for the input message m. The function is $true$ for the case 2 people declared to understand the explanation, and $false$ otherwise. Due to the small size of the decision tree, we could cover all decision paths so that $interpretability(expl(m))$ was true for any path.

Table 1: Textual Features Selected

Feature	Description
Number of links	Number of URLs in the message
Number of mentions	Number of Mentions (@) in the message
Number of hashtags	Number of Hashtags (#) in the message
Number of punctuation	Number of punctuation in the message
Number of uppercase	Number of letters in uppercase in the message
Number of Nouns	Number of as nouns in the message
Number of Pronoun	Number of pronouns in the message
Number of Verbs	Number of verbs in the message
Number of Adverbs	Number of adverbs in the message
Number of Adjectives	Number of adjectives in the message
Number of Prepositions	Number of prepositions in the message
Number of Conjunctions	Number of conjunctions in the message
Number of Numeral	Number of numeral words in the message
Number of Interjections	Number of interjections in the message
Text length	Number of words in the message
Sentiment	Sentiment message classification

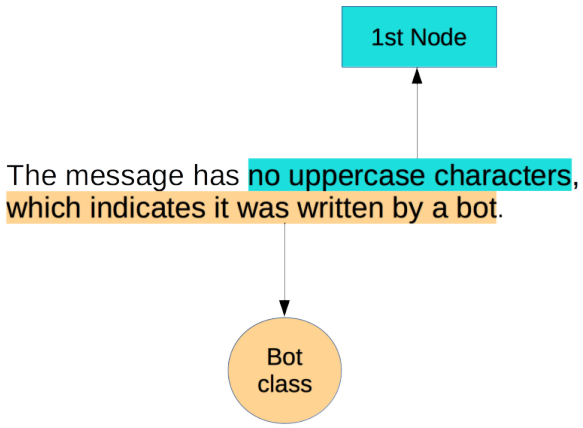

Figure 1: Explanation for the classification decision with decison path: *qtd_upper* < 1 is true ⇒ *bot*

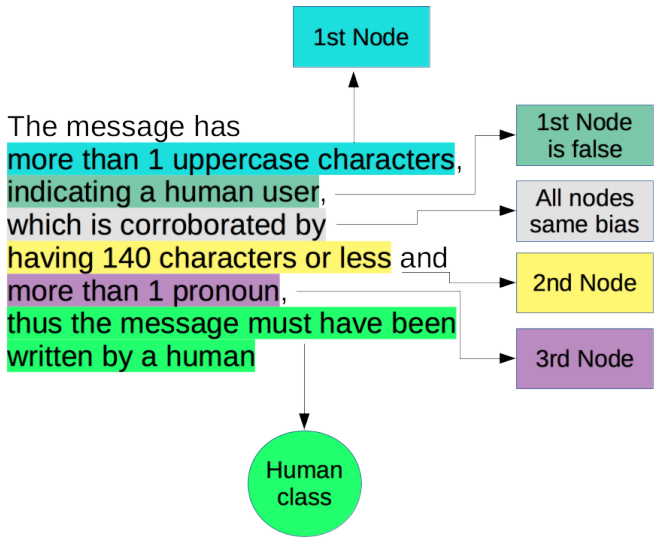

Figure 2: Explanation for the classification decision with decison path: *qtd_upper* < 1 is false → *qtd_text* ≤ 140 is true → *pron_count* ≤ 1 is false ⇒ *human*

The decision to keep the decision tree small was also made knowing that the bots are a moving target in terms of classification. As soon as the provided explanations become commonplace among the OSN users, bot creators will also update their systems to elude the classification criteria. However, by keeping the tree simple and so also the provided explanations, the users would focus on the most relevant cues for bot messages, making it easier for users to keep up to date with the changing criteria. This cat-and-mouse game presumes the frequent inclusion and exclusion of features on the classifier, which can capture the correct textual tracks left by the bot systems. Such features would inevitably go through Semantic Analysis, as bot systems become more sophisticated. The challenge then becomes keeping the explanation of such features (i.e., syntactic, semantic) simple, so the users can identify their instances to properly evaluate them.

5 USER INTERFACE

In this section, we present the user interface to the method developed in work. In order to facilitate use of the tool, we proposed a web browser plugin that allows the user to select the Twitter message and check if it may be a bot message and the explanation

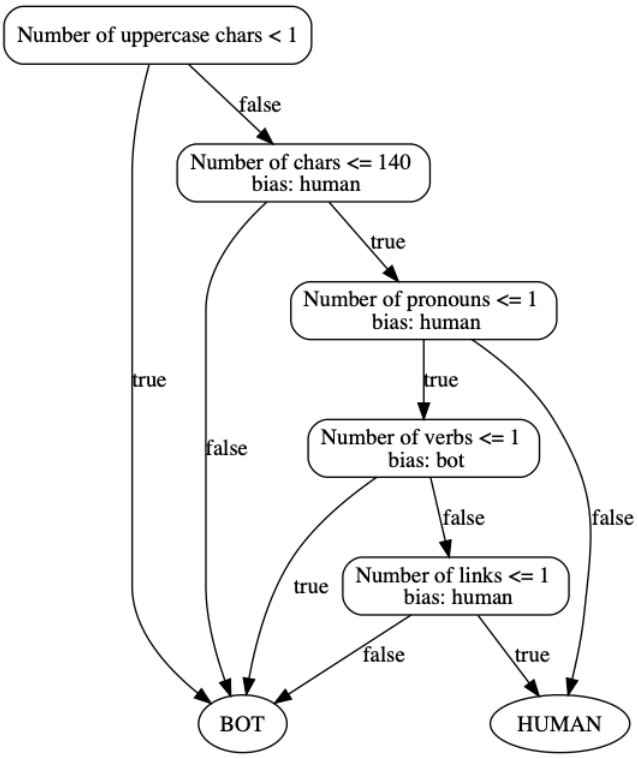

Figure 3: The complete decision tree. Bias values indicate the class that would be accepted if the decision path stopped at that node.

for the classification decision. This research step is intimately connected to the media literacy approach because it generates not only the message classification to the user but how this classification was achieved.

After the user installs the web browser plugin into his/her browser, it will be possible to select a message from the Twitter web site, and with a mouse left-click check if that message is a bot message.

The selection and classification process is shown on Figure 4. The web browser plugin sends the Twitter text to our API that accesses the trained model to check the message. The model returns the text classification with the explanation for the decision. The user will receive a communication that includes both returned information.

In Figure 5 we have an example of how the user will access the plugin. After selecting a message, with the mouse left-click it will be presented the menu with a "Check Message" option. After select this option, our API will receive the text and send to the model for classification. The return example is shown in figure 6, where the message is classified as written by a bot. In the case of a human message, the plugin returns the response shown in Figure 7.

In these figures, we used text from our datasets and explanation obtained from the decision tree classification path. In Figure 6, the message "@blabla[5] all of your tweets are answered in my blog postings." is classified correctly as a bot message, given that it has

[5]Username suppressed

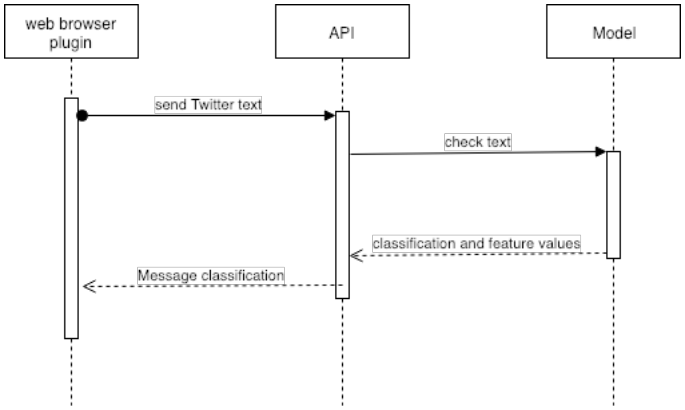

Figure 4: Sequence Diagram.

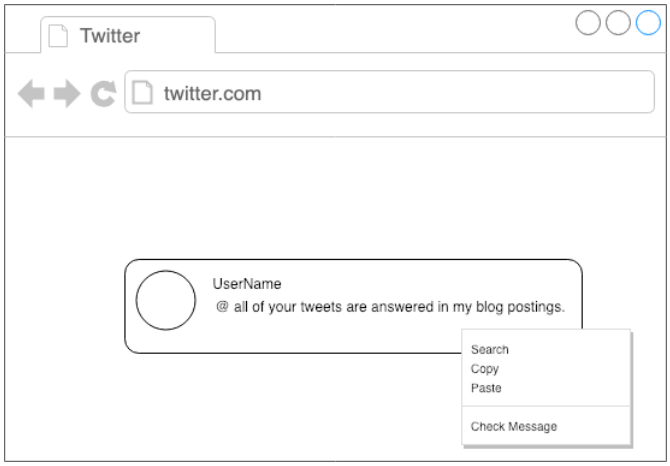

Figure 5: Plugin selection

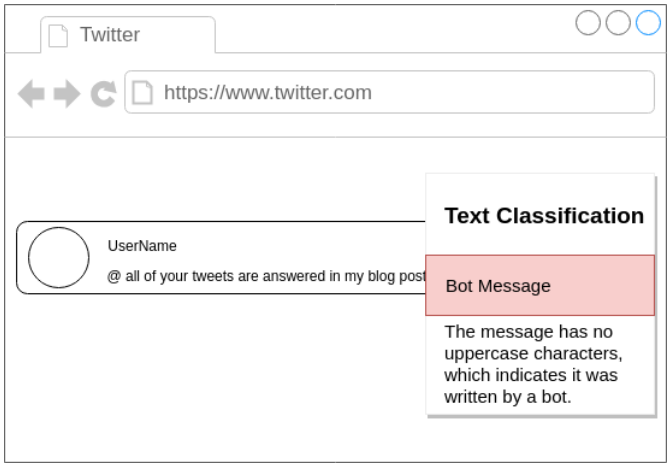

Figure 6: Plugin Response for Bot Message

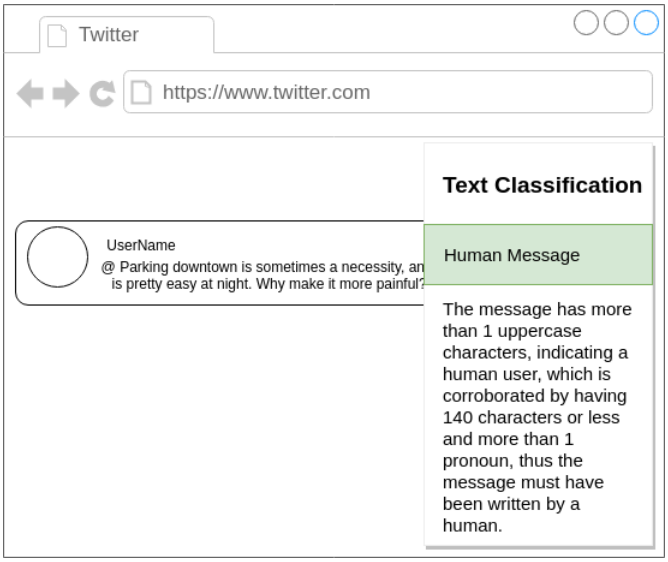

Figure 7: Plugin Response for Human Message

no uppercase letter, as explained in the previous section. It is an example of true positive test, where the model correctly detected a bot message.

The Figure 7, the message "@blabla[6] Parking downtown is sometimes a necessity, and usually is pretty easy at night. Why make it more painful? #SLC" is correctly classified as a human message, having 5 uppercase letters, a text length less than 140 characters and a pronoun. Based on our model, such characteristics are more related to the way that a human writes a message, rather then a bot. This case is an example of true negative test, where the classification is not a bot message (human message).

We observed in other example, the message "Supreme Court postponed to hear the appeal of Andualem Arage et al, who are jailed on terrorism charges, for November 22. #Ethiopia" has more than one uppercase letter (6), has less than 140 characters (131) and one pronoun. With this values, the model classify this message as a human message, however, it is a bot message. This classification is a false negative, where we have a wrong classification to a non-bot message.

A false positive example is a message that does not contain any uppercase letters, and is automatically classified as a bot message but instead belongs to a human user.

These misclassifications can be overcome through user feedback, which inform the tool about the mistake, so this can be used to improve our classifier. Additionally, collecting better textual features may improve the results, although those require further investigation.

Based on the main advantages introduced by [15], our proposed method shows how the message can be interpreted, given the message classification and how a user can look into media content before spreading a harmful messages that can have a serious impacts on Society.

[6]Username suppressed

6 CONCLUSIONS

This work presents a study that aimed to analyze textual features from bot and human written messages that were used in previous related studies, verifying the feasibility for agile bot detection while producing useful guidance for an ordinary user to recognize a bot message. This approach, based on media literacy, has the main objective to teach a user how one could detect an eventual bot message, which we believe contributes to non-dissemination of this kind of message.

We analyzed messages based on textual features of the dataset provided by Morstatter et. al. [14], composed by Twitter messages from the Arab Spring event and labeled as bot and human users. The use of only text is the simplest way for an ordinary user to be aware that there may be something wrong with the message. The combination of many characteristics, such as the number of followers, number of friends, among others, is much more difficult for a human to check the many possibilities and judge if the text was written automatically.

The decision tree machine learning algorithm was used to select the best features and generate the explanation to teach the user how to recognize a bot message. We simplify the tree structure to reach an intelligible model, at a minor cost on model accuracy.

A web browser plugin is proposed, that will help the user check if a message is a bot message or not. This plugin sends the text to our trained model through a service API, which in turn returns to the plugin user the message classification and an explanation on the steps to achieve the result. In this way, users will check the message classification and also learn how to classify messages on their own.

The method proposed in this paper is different from related works due to the focus being not only to bot and human messages using machine learning, but use the obtained model to teach an OSN user how to distinguish bots from humans.

As a limitation, the model accuracy is considerably under the current state of the art for Twitter data and need to be improved. The use of textual features only decreased the accuracy in exchange for simplicity. A way of improving this would be to include alternative textual features, such as event and sub event relation, among others. Such forward solutions should also cover the constant evolution of bot systems, which will adapt to the criteria exposed by our system, while keeping the users up-to-date with the ever changing criteria.

For the next works we will improve the model, adding more textual features, without losing interpretability. We will also apply this approach in other datasets, we will compare messages along time, subject or events and also messages in different languages. In this way, we can examine how such textual patterns are build and changed. Conducting acceptability tests with real users based on the web browser plugin will be a future task.

REFERENCES

[1] Abdulrahman Alarifi, Mansour Alsaleh, and Abdul Malik Al-Salman. 2016. Twitter turing test: Identifying social machines. *Information Sciences* 372 (dec 2016), 332–346. https://doi.org/10.1016/j.ins.2016.08.036

[2] Mansour Alsaleh, Abdulrahman Alarifi, Abdul Malik Al-Salman, Mohammed Alfayez, and Abdulmajeed Almuhaysin. 2014. TSD: Detecting sybil accounts in twitter. In *Proceedings - 2014 13th International Conference on Machine Learning and Applications, ICMLA 2014*. IEEE, 463–469. https://doi.org/10.1109/ICMLA.2014.81

[3] S. Aslan. 2018. Twitter by the Numbers: Stats, Demographics & Fun Facts. https://www.omnicoreagency.com/twitter-statistics/

[4] Sajid Yousuf Bhat and Muhammad Abulaish. 2013. Community-based features for identifying spammers in online social networks. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining - ASONAM '13*. ACM Press, 100–107. https://doi.org/10.1145/2492517.2492567

[5] Aaron Black. 2018. A new study suggests fake news might have won Donald Trump the 2016 election. https://www.washingtonpost.com/news/the-fix/wp/2018/04/03/a-new-study-suggests-fake-news-might-have-won-donald-trump-the-2016-election/?noredirect=on&utm_term=.1029a721270f

[6] P. Cooper. 2019. 28 Twitter Statistics All Marketers Need to Know in 2019. https://blog.hootsuite.com/twitter-statistics/

[7] John P. Dickerson, Vadim Kagan, and V. S. Subrahmanian. 2014. Using sentiment to detect bots on Twitter: Are humans more opinionated than bots?. In *ASONAM 2014 - Proceedings of the 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. IEEE, 620–627. https://doi.org/10.1109/ASONAM.2014.6921650

[8] Jennifer Fleming. 2014. Media literacy, news literacy, or news appreciation? A case study of the news literacy program at Stony Brook University. *Journalism and Mass Communication Educator* 69, 2 (jun 2014), 146–165. https://doi.org/10.1177/1077695813517885

[9] C.J. Hutto and E.E. Gilbert. 2014. VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. *Eighth International Conference on Weblogs and Social Media (ICWSM-14)* (jun 2014).

[10] Rodrigo Augusto Igawa, Sylvio Barbon, Katia Cristina Silva Paulo, Guilherme Sakaji Kido, Rodrigo Capobianco Guido, Mario Lemes Proenca Junior, and Ivan Nunes da Silva. 2016. Account classification in online social networks with LBCA and wavelets. *Information Sciences* 332 (mar 2016), 72–83. https://doi.org/10.1016/j.ins.2015.10.039

[11] Bernardo Pereira Lauand. 2016. *Abordagem Experimental para a Localizacao e Detecao de Eventos em Tweets*. Master's thesis.

[12] Sonia Livingstone. 2004. Media literacy and the challenge of new information and communication technologies. *Communication Review* (2004).

[13] Juan Martinez-Romo and Lourdes Araujo. 2013. Detecting malicious tweets in trending topics using a statistical analysis of language. *Expert Systems with Applications* 40, 8 (jun 2013), 2992–3000. https://doi.org/10.1016/j.eswa.2012.12.015

[14] Fred Morstatter, Liang Wu, Tahora H. Nazer, Kathleen M. Carley, and Huan Liu. 2016. A new approach to bot detection: Striking the balance between precision and recall. In *Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2016*. IEEE, 533–540. https://doi.org/10.1109/ASONAM.2016.7752287

[15] Art Silverblatt, Andrew Smith, Don Miller, Julie Smith, and Nikole Brown. 2014. *Media Literacy: Keys to Interpreting Media Messages* (4 ed.). 549 pages.

[16] J. Sommerland. 2018. Russian Hackers Used Tumblr To Spread 'fake News' During Us Election, Company Reveals. https://www.independent.co.uk/life-style/gadgets-and-tech/news/tumblr-russian-hacking-us-presidential-election-fake-news-internet-research-agency-propaganda-html

[17] A. Sulleyman. 2017. TWITTER INTRODUCES 280 CHARACTERS TO ALL USERS. https://www.independent.co.uk/life-style/gadgets-and-tech/news/twitter-280-characters-tweets-start-when-get-latest-a8042716.html

[18] J. Swaine. 2018. Twitter admits far more Russian bots posted on election than it had disclosed. https://www.theguardian.com/technology/2018/jan/19/twitter-admits-far-more-russian-bots-posted-on-election-than-it-had-disclosed