CMSC396H Fall 2020: Twitter Bots

Mohammad Subhan
Sahil Dev
Gabe Margolis
Joshua Goldberg
Jordan Foster
University of Maryland, College Park
College Park, MD, USA

ABSTRACT

CCS CONCEPTS

• **Information systems → Social networks**; • **Human-centered computing → Social network analysis**.

ACM Reference Format:
Mohammad Subhan, Sahil Dev, Gabe Margolis, Joshua Goldberg, and Jordan Foster. 2020. CMSC396H Fall 2020: Twitter Bots. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Social media has become increasingly prevalent in the past decade, hosting a wide variety of content, ranging from family photos to online communities to discussions of current events. There are many social media platforms on the Internet today, including, Facebook, Youtube, Instagram, and WhatsApp which sit at the top of the most heavily used networking sites. As of July 2020, Twitter has the 15th most monthly active accounts with approximately 326 million users [4]. Twitter allows users to post and respond to posts dubbed "tweets", each limited to 280 Unicode characters, though images and videos may also be posted. Tweets are shared directly with a user's followers, who may then like or retweet content they are interested in to share it with their followers. This chain pushes popular tweets to reach millions of people. The largest portion of Twitter users comes from the United States and Japan [3]. Over the last several years, there have been reports of bot accounts creating and sharing tweets under the guise of being a real human. This has had serious consequences, allowing governments, corporations, and individuals to sway the public opinion about crucial events, such as the 2016 election. The goal of this research is to be able to accurately identify such bots on Twitter. By doing so, we can better identify and avoid mass manipulation propagated by computers. With regard to elections, this work could help to ensure a more democratic and transparent process without the involvement of companies or other countries.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

This problem has been tackled by many researchers before, and our goal is to improve upon the various standards achieved by previous models. By identifying recurring issues and various approaches in previous research, we create our own method that succeeds in identifying bots more accurately. Improving upon past work will of course allow future research to build off of the work here and develop into more complex and reliable classification methods, for example those described in the Future Work section.

We attempt to take a context-sensitive approach to solving the problem of identifying Twitter bots, adding several additional features that constitute a more comprehensive and content-aware dataset. It is worth noting that public research works such as this may allow bot creators to determine more effective approaches for misleading the public, which is why misinformation detection approaches [5] will ultimately be necessary to combat harmful manipulation.

We begin with Related Works as the backbone of our research. We then discuss our approach to the problem and how we use previous research coupled with novel text analysis techniques to develop a more context-sensitive approach to identifying Twitter bots. Next, we discuss our results, highlighting the most promising features of our research while also identifying potential improvements. Finally, we conclude with a summary of our research, along with an overview of future work.

2 RELATED WORKS

In "Detection of Novel Social Botsby Ensembles of Specialized Classifiers" Sayyadiharikandeh et al. seek to differentiate the behaviors of bots and separate them into different categories [12]. Each model is trained to recognize three specific categories of bots, which includes, but is not limited to, traditional spambots, social spambots, and fake followers. Traditional spambots bombard users with content, social spambots support and/or attack users, and fake followers frequently follow users [12]. This approach to detecting bots on Twitter proved to be fairly accurate when compared to other successful bot detection methods. The main metric used for determining the accuracy of the method of bot detection used in the paper, the Ensemble of Specialized Classifiers (ESC), is the area under the ROC curve. This metric, in this case, is how successful a model can determine a human account from a bot account with 1.0 and 0.0 being the highest and lowest scores respectively. ESC produced an AUC of 0.96 which is comparable to the AUC of Botometer which produced one of 0.97 [12]. The work of Knauth in "Language-Agnostic Twitter Bot Detection" focuses on the fact that there are no human limitations by twitter bots [6]. The data used in

this research is from the MIB dataset which contained 8375 twitter accounts. As a result, account features such as number of friends, geo enabled, and protected as a means to determine if a twitter account is a bot not. The features that proved to be most beneficial in bot determination include Levenshtein distance, geo enabled, and statuses count. The work done in this paper was able to predict the account type with an accuracy of 98-99 percent. However, as noted in the paper, bot developers will pick up on research such as this and future researchers must use new data as existing data may not work so well. This is an important thing to note as the paper was written over a year ago. The paper "Twitter Bot or Not?" [8], compares the performance of six machine learning models, with 28 account features, 12 features for individual tweets, and 19 features for aggregate data over the past 100 tweets for each account, providing for a somewhat context-aware solution that is only partially capable of interpreting the context of tweet. This differs somewhat from works mentioned previously, most of which focus on developing new methods of improving on a preexisting one. The six models used along with the cross-validation accuracy for each are as follows. A single decision tree, a flowchart-like structure where at each step a comparison or test is done to determine which path to take, and where the leaves represent a final prediction, achieved 89% accuracy. A random forest (with adaptive boosting) is a set of decision trees where the majority predicted class is selected; the random forest model achieved 92% accuracy. Logistic regression (L1) uses a linear gradient descent model with a sigmoid component for binary classification with lasso regularization; the L1 linear regression model achieved 89% accuracy. Logistic regression (L2) uses ridge regularization and achieved 88% accuracy. A multilayer perceptron is a multilayer logistic regression-like model, where each layer feeds as input into the next; the multilayer perceptron model achieved 88% accuracy. A voting classifier combines several models and chooses the majority predicted class extending the concept of a random forest to varying types of models; the voting classifier achieved 93% accuracy. In terms of F1 score, the voting classifier performed the best (0.93), followed closely by the random forest with adaptive boosting (0.92) and L1 logistic regression (0.90). Their analysis uses the majority class classifier as a baseline with accuracy of 50.4%.

3 APPROACH

We made use of the MIB dataset[11]. This dataset includes account and tweet data of over 3,000 human accounts and over 10,000 bots. Jupyter Notebook was used for all Python code along with GitHub for version control[7]. Scikit-learn was also used for all machine learning classifiers and evaluation[9]. All code and results (excluding the dataset due to terms of use) is available at https://github.com/sdev00/Twitter-Bots . We initialize the dataset by combining all users from individual datasets from the MIB dataset and adding a classification column signifying whether or not the user was confirmed to be a bot or human. Next, properties were added to each user based on Tweet data. Information added from all the tweets can be seen in Table 1.

For each user, the 25 most recent tweets were extracted from the MIB dataset for further text processing and analysis. In particular, the Python Natural Language Toolkit (NLTK) [2] has useful tools for string tokenization and sentiment analysis. In our feature extraction, we used RegexpTokenizer and SentimentIntensityAnalyzer [1] from NLTK to engineer four new features listed in the table below.

Table 1: Aggregate text properties based on values from the MIB dataset

Property	Description
num_tweets_dataset	Total number of tweets by user in dataset
num_retweets_post_dataset	Number of retweets by user
num_reply_post_dataset	Number of replies by user
retweet_post_percent	Percentage of retweets out of all tweets
reply_post_percent	Percentage of replies out of all tweets
avg_hashtags	Average number of hashtags used per tweet
avg_urls	Average number of urls used per tweet
avg_retweets_cnt	Average number of retweets per tweet
avg_reply_cnt	Average number of replies per tweet
avg_favorite_cnt	Average number of favorites per tweet
recent_tweet_text	Full text of the last 25 tweets from each user

Table 2: Additional properties from text analysis

Feature	Description	Explanation
unique_words_per_word	Number of unique words divided by the total number of words	Estimates the size of the user's vocabulary
avg_neg_sentiment	Average negative sentiment across a user's tweets	Evaluates how negative a user's language is
avg_neu_sentiment	Average neutral sentiment across a user's tweets	Evaluates how neutral a user's language is
avg_pos_sentiment	Average positive sentiment across a user's tweets	Evaluates how positive a user's language is

As mentioned in the explanation sections, we essentially recovered features that correspond roughly to the size of the user's vocabulary, and the intensity of a user's language for each of negative, neutral, and positive sentiment. We hypothesize that bot accounts will tend to have a more limited vocabulary and more intense non-neutral sentiment, given bots' incentive to post repetitive and inflammatory content. Finally, we included several basic account features from the MIB dataset, a summary of which can be found in Table 3.

We use our final processed dataset to train and test five main classifiers, each with a number of hyperparameters we attempt to tune roughly. These classifiers include gaussian naive bayes, random forest, logistic regression, multilayer perceptron, and voting classifiers, all of which were implemented with the scikit-learn

Table 3: User properties from MIB dataset

Property	Description
geo_enabled	Geo location enabled
default_profile	Whether the user has modified theme or background
default_profile_image	Default profile image used
followers_count	Number of followers
friends_count	Number of following
favourites_count	Number of tweets a user has liked
listed_count	Number of lists a user is a part of

library. We adopt naive bayes as our baseline classifier. Previous approaches have used a basic majority class classifier, though this provides a poor baseline as compared to a simple probabilistic model like naive bayes. Ultimately, the baseline we use is of little importance.

For each classifier we split the data 70/30 into a training and test set. For evaluating our models, scikit-learn provides a few useful metrics. Accuracy is the percentage of accounts that were identified correctly as either bot or human; precision is the percentage of actual human accounts out of all accounts that were identified as being human; recall is the percentage of actual human accounts that were successfully identified as human; and F1 score takes both precision and recall into account by computing the harmonic mean of the two values [10]. In practice, the F1 score is close to the mean of the precision and recall, but weighted more heavily towards the lesser of the two [13]. We select overall accuracy and recall on the human set as our main evaluation metrics, though all of these metrics were computed for the classifiers on the MIB dataset with and without text analysis. We use overall accuracy and recall in particular, as accuracy provides a good measure of overall performance, especially on a roughly balanced dataset, while recall is a necessity in such a system where misclassifying a human as a bot is potentially much worse than misclassifying a bot as a human.

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{TP}{\text{TP} + \frac{1}{2}(\text{FP} + \text{FN})}$$

4 RESULTS

In this section we will summarize the results that were obtained from each classifier on the initial MIB dataset without text analysis and the dataset with text analysis.

Table 4: Gaussian Naive Bayes Summary

	Precision	Recall	F_1 Score
Bot, no text analysis	97%	94%	95%
Human, no text analysis	84%	92%	88%
Bot, with text analysis	97%	99%	98%
Human, with text analysis	98%	94%	96%

The accuracy for the dataset without text analysis is 93.3% and for the dataset with text analysis it increased to 97.2%. Since we adopt naive bayes as our baseline, it is promising that the accuracy improved by 3.9% and the human recall improved by 2% with the added features in the text analysis dataset. This suggests that the added features are useful in predicting whether an account is a bot or a human.

Table 5: Random Forest Summary

	Precision	Recall	F_1 Score
Bot, no text analysis	99%	99%	99%
Human, no text analysis	98%	98%	98%
Bot, with text analysis	99%	99%	99%
Human, with text analysis	98%	98%	98%

The random forest classifier performed well for both datasets. The accuracy for the dataset without text analysis is 98.7%, compared to 98.4% for the dataset with text analysis. Random forest performed best when the hyperparameters for the number of estimators (decision trees) used was set to 5, with the maximum depth for each tree set to 6. We also tested a random forest with 5 estimators and a max depth of 4, as well as 25 estimators and a max depth of 4. With the initial dataset, the best performing random forest achieved higher recall than only the random forest with 5 estimators and a max depth of 4. With the text analysis dataset, the best performing random forest achieved 1% higher recall than both of the other random forest models.

Table 6: Logistic Regression Summary

	Precision	Recall	F_1 Score
Bot, no text analysis	98%	97%	97%
Human, no text analysis	91%	95%	93%
Bot, with text analysis	98%	96%	97%
Human, with text analysis	92%	96%	94%

We implemented logistic regression with L1 regularization and with L2 regularization. The model with L1 regularization achieved higher accuracy under both datasets at 96.1% accuracy on both, as compared to the model with L2 regularization, which achieved 92.3% and 95.9% accuracy on the initial and text analysis datasets, respectively. The recall for the model with L1 regularization also increased from 95% to 96% under the text analysis dataset, while the recall for the L2 regularization model decreased from 98% to 96% under the text analysis dataset.

Table 7: Multilayer Perceptron Summary

	Precision	Recall	F_1 Score
Bot, no text analysis	99%	99%	99%
Human, no text analysis	97%	97%	97%
Bot, with text analysis	99%	98%	98%
Human, with text analysis	96%	97%	97%

We trained four multilayer perceptron classifiers, two with one hidden layer of size 20 and two with two hidden layers of size 20. For

each architecture, one used an Adam solver and one used an L-BFGS solver. A ReLU activation was used for all multilayer perceptron models. The Adam solver with two hidden layers performed the best on the initial dataset, at 98.4% accuracy and 97% recall, while the Adam solver with one hidden layer achieved the highest accuracy on the text analysis dataset, at 98.1% accuracy and 98% recall.

Table 8: Voting Classifier Summary

	Precision	Recall	F_1 Score
Bot, no text analysis	99%	99%	99%
Human, no text analysis	98%	98%	98%
Bot, with text analysis	99%	99%	99%
Human, with text analysis	98%	98%	98%

Our final classification method implemented a voting classifier. We used six types of voting classifiers on each dataset, where each classifier used in the voting classifier was the best performing model of each other type. There are also two methods of voting. Hard voting takes the majority vote, while soft voting combines the likelihood predictions before determining the most likely prediction. For the initial dataset, the hard voting classifier with the random forest, multilayer perceptron, and logistic regression performed the best, achieving 98.7% accuracy and 98% recall, while in the text analysis dataset, the soft voting classifier with the random forest, multilayer perceptron, and naive bayes performed the best, achieving 98.8% accuracy, just 0.1% higher than in the initial dataset.

5 CONCLUSION

We trained five main classifiers and evaluated them on the metrics of overall accuracy and recall. Our best performing model (soft voting classifier) achieved 98.8% accuracy and 98% recall on the dataset with added text analysis features. While the performance of the models did not change significantly with the extracted text analysis features, most of the minor changes were increases in accuracy or recall. Overall, it is still unclear whether these added features help to predict whether an account is run by a bot. More work must be done on extracting valuable features from the tweet data and achieving context-sensitive classification. Unfortunately due to time constraints, we were unable to implement some of the ideas we had going into the project, but several such concepts are discussed below. Despite our text analysis dataset not providing much use in classification, we still achieved quite high accuracy, only misclassifying 1.2% of account instances, which does mark a solid improvement over past results, likely attributable to a combination of having a more reliable dataset than those publicly available and improved tuning of hyperparameters. Similarly high was our recall, only misclassifying 2% of humans as bots, a marked improvement from previous approaches as well. However as mentioned previously, recall must be significantly higher than this to avoid issues with misclassifying humans as bots. There is still plenty of work to be done in the Twitter bot classification space, but we have provided our analysis of a number of hyperparameters and several interesting extracted features.

6 FUTURE WORK

One limitation of this study is the age of the dataset. The MIB dataset was collected at varying times ranging from 2009 to 2014. Since previous research has been done with this dataset, and given the length of time, it is likely that many bots have adapted new techniques to make them less distinguishable from human accounts. Applying these techniques to a dataset augmented with more recent data, for example from the 2020 election, we would be better able to determine if the bot identification techniques used here are viable today, and of course, additional training data would help significantly in improving our models and identifying flaws. The text analysis covered in this paper is very basic, and more complex text analysis techniques may prove more useful in identifying bots. For example some sort of similarity matching between tweets from the same user could help in identifying the repetitive content that bots tend to post, since a bot is more likely to focus on a single subject across multiple tweets. This could be implemented in a couple of ways, for example, from least to most context-aware: string matching, TFIDF vectorization, and CBOW models. It would also be helpful to analyze the posting patterns that bots tend to have. A network analysis could improve our ability to identify bots, as well as identifying particular user mentions, hashtags, general phrases, or sentence construction patterns that are more likely to be associated with a bot than a human. Even extracting data from the replies to a post could help with bot identification. On the side of developing models, improved feature selection may help to reduce overfitting. By identifying and removing predictors that do not help much in classifying bots, the models will be better at avoiding responses to unimportant background noise in the data. Additionally, class balancing can be used to improve the recall, though not necessarily the overall accuracy of the model. But since the human class is more important to get right than the bot class, as long as we can achieve very high recall, such a predictive model may actually be useful in an applied setting.

REFERENCES

[1] 2020. Bot Repository. https://botometer.osome.iu.edu/bot-repository/datasets.html

[2] Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python.* O'reilly.

[3] J Clement. 2020. Most Used Social Media 2020. http://statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/

[4] J Clement. 2020. Twitter: Most Users by Country. http://statista.com/statistics/242606/number-of-active-twitter-users-in-selected-countries/

[5] Md Rafiqul Islam, Shaowu Liu, Xianzhi Wang, and Guandong Xu. 2020. Deep learning for misinformation detection on online social networks: a survey and new perspectives. *Social Network Analysis and Mining* 10 (09 2020). https://doi.org/10.1007/s13278-020-00696-x

[6] Jürgen Knauth. 2019. Language-Agnostic Twitter Bot Detection. *Proceedings - Natural Language Processing in a Deep Learning World* (10 2019). https://doi.org/10.26615/978-954-452-056-4_065

[7] F. Loizides and B. Schmidt (Eds.). 2016. *Jupyter Notebooks – a publishing format for reproducible computational workflows.* IOS Press.

[8] Revanth Mattapalli, Varun Elango, and Vignesh Ramesh. 2017. Twitter Bot or Not? https://github.com/Vignesh6v/Twitter-BotorNot/blob/master/twitter-bot_or_not.pdf

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

[10] David Powers. 2011. Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness Correlation. *Journal of Machine Learning Technologies* (2011).

[11] Proceedings of the 26th International Conference on World Wide Web Companion 2017. *The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race*. Proceedings of the 26th International Conference on World Wide Web Companion, International World Wide Web Conferences Steering Committee. https://doi.org/10.1145/3041021.3055135

[12] Mohsen Sayyadiharikandeh, Onur Varol, Kai-Cheng Yang, Alessandro Flammini, and Filippo Menczer. 2020. Detection of Novel Social Bots by Ensembles of Specialized Classifiers. *Proceedings of the 29th ACM International Conference on Information Knowledge Management* (10 2020). https://doi.org/10.1145/3340531.3412698

[13] Da-Feng Xia, Sen-Lin Xu, and Feng Qi. 1999. A proof of the arithmetic mean-geometric mean-harmonic mean inequalities. *RGMIA Research Report Collection* 2 (1999).