Bot or Not: Predicting Reddit User's Humanity through Network Measures

Akio, Ben, Jonathan, Yi

Research question and motivation

Bots play an interesting role on Reddit. Hackers spend an incalculable amount of time trying to pass off computer programs to behave and interact like any other people, but they leave traces of their inhumanity here and there. While this can make users on Reddit more convenient due to automation, it can also leave room for malicious behaviors. Some malicious bot behaviors include spambots where bots can automatically add value to comments that don't deserve credit. There are even some bots that are only there to send users links to malicious sites. Our project aims to uncover the traits that make suspected bots act differently from humans from a networks perspective.

Question: Is it possible to apply network analysis to a collection of Reddit comments in order to uncover patterns in user behavior that might indicate if that user is a bot? Are there other patterns that might be uncovered?

Introduction

Reddit is a social media platform in which any user who creates an account can post content (i.e. create a "post"), and comment on other users' posts (i.e. create a "comment"). Users can also reply to already created comments with a new comment. The following lists the terminology we will use:

- *User*: Reddit account which can create posts and comments. Often referred to as a singular entity which represents the individual or group controlling the account.
- *Comment*: comment created by a user on Reddit. Includes text and metadata (time posted, username, etc).
- *Parent*: comment in which the relevant comment is a reply to. E.g. if comment B is posted in response to comment A, comment A is the *parent* of comment B. Comments made in direct reply to posts do not have parents.
- *Child*: comment that is a reply to the relevant comment. E.g. if comment B is posted in response to comment A, comment B is the *child* of comment A.
- *Human*: user in which the account is controlled by a human or humans. These users use Reddits front-end user interface to control their account.
- *Bot*: user in which the account is controlled by a computer. While there is obviously a human who created them, these users use software to control their account.

Related work

A.H. Wang - Detecting Spam Bots in Online Social Networking Sites: A Machine Learning Approach

Attempts to find content and network based features from a twitter data set in order to better classify users as either spam bots or not. Identifies several useful features, however several, such as "follower count" and "number of followers" are unique to twitter. We attempt to find an analogous measure in our dataset by looking at the number of times a user made a comment to another user

("weight" attribute of an out-edge) and the number of times another user made a comment in reply to a given user ("weight" attribute of an in-edge)

Clayton Allen Davis, Onur Varol, Emilio Ferrara, Alessandro Flammini, and Filippo Menczer - *BotOrNot: A System to Evaluate Social Bots*

Researchers created an API to provide a numerical score based on temporal, network, content, and sentiment features for a given twitter account in order to provide a probability that the user is a bot. The feature set suggested in this includes network features such as degree distribution, clustering coefficient, and centrality, along with metadata gathered from the twitter API and semantic analysis of the text.

Fred Morstatter, Liang Wu, Tahora H. Nazer, Kathleen M. Carley, and Huan Liu - *A new approach to bot detection: striking the balance between precision and recall*

Proposes a model which emphasizes recall over accuracy when identifying suspected bot accounts in order to optimise the F_1 score at the cost of false positives. In addition to typical text and metadata features, applies a latent Dirichlet allocation to the combined text of all of a users tweets. Given the resulting size of the feature set, they then apply a slightly modified adaptive boosting algorithm on top of a decision tree classifier in order to improve performance.

John P. Dickerson, Vadim Kagan, and V. S. Subrahmanian - Using sentiment to detect bots on Twitter: are humans more opinionated than bots?

Intentionally ignores syntactic and network features to focus instead on a combination of topic and sentiment analysis on a set of tweets to identify bot behavior. The authors suggest that sentiment analysis of the content of tweets can be among the most important when creating a classifier to identify bot behavior. They look specifically at the user's sentiment on a particular topic over time, and suggest that while humans will typically not change their view on a given topic in a short space of time, bot accounts aimed at influencing other users will.

Data

Google Big Query contains a complete set of reddit comments collected from the past two years and organized by month. The data can be retrieved from here: https://bigquery.cloud.google.com/table/fh-bigquery:reddit_comments.2018_10?pli=1 . We retrieved a month's worth of data from October using the Google Big Query interface. Some variables used here are: number of comments made, mean comment score, standard deviation of the minute after the hour the comment was made, standard deviation of the word counts, number of unique subreddits the user comments on, and the ratio of comments per subreddit. These variables will subsequently be called "text metadata."

The end goal of this research is to be able to classify bots that try to pass as humans, as bots. In order to do this, we need a training set that consists of both bots and humans. Putting this training set together was very difficult. We wanted the bots in the training set to be representative of bots who pass

as humans, as that type of bot is who we want to be able to accurately classify. However, that is not realistic, because if we were able to identify those bots so that we could confidently label them as “bots”, we would have no need for this research. Therefore, our bot training set is representative of “friendly” bots whose identify as a bot is obvious to other users.

In order to gather the data on bots or non-bots, we needed to identify the users who would be in our training set. On Reddit, humans often respond to bots with the comment “Good bot” or “Bad bot”. This is a way of telling the creator of the bot that they think the bot is performing a good service and acting how it should, or acting incorrectly. We gathered every “Good bot” and “Bad bot” comment made in October, 2010, and recorded the usernames who wrote the parent comments. We then used that collection of users as our “bot” dataset. Every user who was not in that dataset we put in our “human” dataset.

There are many qualifications with the dataset that come as a result of this method of data collection. One, some users in the human dataset are bots, and some in the bot dataset are humans. This is because not all bots will have gotten a “Good bot” or “Bad bot” reply during October, and thus they will be put into the human dataset. On the other hand, for various reasons, some humans on Reddit reply to other humans with “Good bot” or “Bad bot”. In order to try and combat these discrepancies, we did some manual curation of the dataset. Specifically, we combined through all data of the users that were labeled as bots, and removed users from the bot dataset who we could confidently say were in fact humans. While this curation in no way assures the complete validity of the dataset, we feel that it got us to a point where the dataset is good enough as to be able to separate humans from bots.

Methods

When running our classification algorithms, we randomly split the dataset described above into a training set and a test set, maintaining the balance of percentage of users who are bots and percentage who are humans in both the training and test sets. We trained our classifiers on the training set and classified users in the test set, using the test labels to judge the quality of the classifications.

To classify new users as bots or humans, we used a support vector machine classifier and a random forest classifier. The features used in classification came from metadata of the comments posted by the users, and network characteristics of a user-user graph that we constructed based on who users replied to. There are six metadata features that we calculated: the number of comments that the users made in our dataset (*num_comments*), the average score that those comments received (*mean_score*), the standard deviation of the minute in which the comments were posted (*std_minute*), the standard deviation of the length of their comments, where length equals the number of words (*stdnum_words*), the number of different subreddits their comments are posted in (*num_unique_subs*), and that number of different subreddits divided by the number of comments (*unique_subs_per_comment*). A list of all features used in our analysis is in the appendix. Motivation for including a few of these features is presented in the following paragraphs.

We use the standard deviation of the minute of the comments because we postulated that, while a user should post uniformly in the range of zero to sixty, bots are likely programed to post at specific time intervals, such as every hour or half hour. Therefore, the standard deviation of the minute of bot comments would be much smaller than that of users. Not only was this idea backed up by the related articles we reviewed, it was verified empirically (Figure 1). While those articles use Twitter data, they

mention that bots indeed are much more rigid in their post times. Looking at the histograms below we can see that bots do slightly favor a smaller variance in minute of the post.

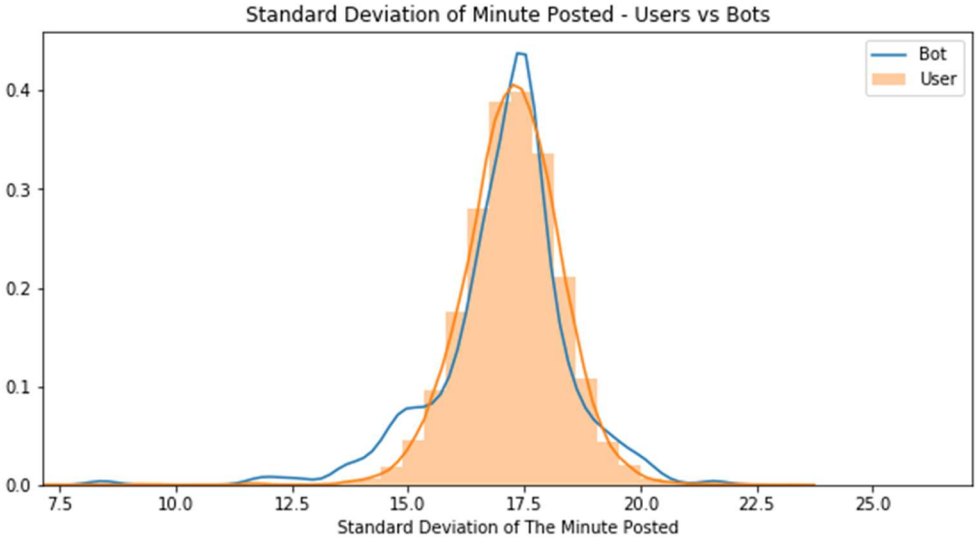

Figure 1. A histogram of standard deviation of the minute at which comments by each user are posted

Our hypothesis regarding the number of unique subreddits that a user posts to is that bots would be more focused on specific subreddits, while users would be most likely to spread out their comments amongst multiple subreddits. The metric we used for this was the number of unique subreddits divided by the number of comments. So if a user's value was 1, each of their comments would be in a different subreddit. If they had 100 comments and all were in the same subreddit, the value would be 0.01. We can see by the histograms in Figure 2 that our hypothesis was proven incorrect. The opposite is true, where bots are more likely to spread their comments out over many subreddits than users are.

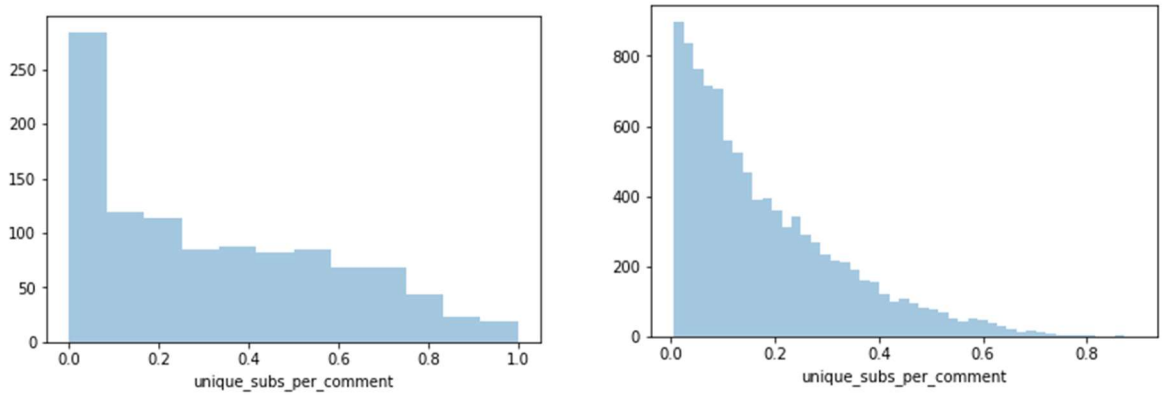

Figure 2. A histogram of unique subreddits divided by total comments for users is on the left, and one for bots is on the right.

In order to calculate the network features, we first created a network where each node represents one user in our dataset (both bot and human). Directed edges in the network represent

comments that one user makes in response to a comment of another user. For example, if user A makes a comment and user B responds to that comment with a comment of their own, there is a directed edge from node B to node A. The weight of the edge is equal to the number of replies user B made to comments by user A in the dataset. Based on this network, we calculated features such as triad frequency (Figure 3), ego-density (Figure 4), in degree and out degree, and more.

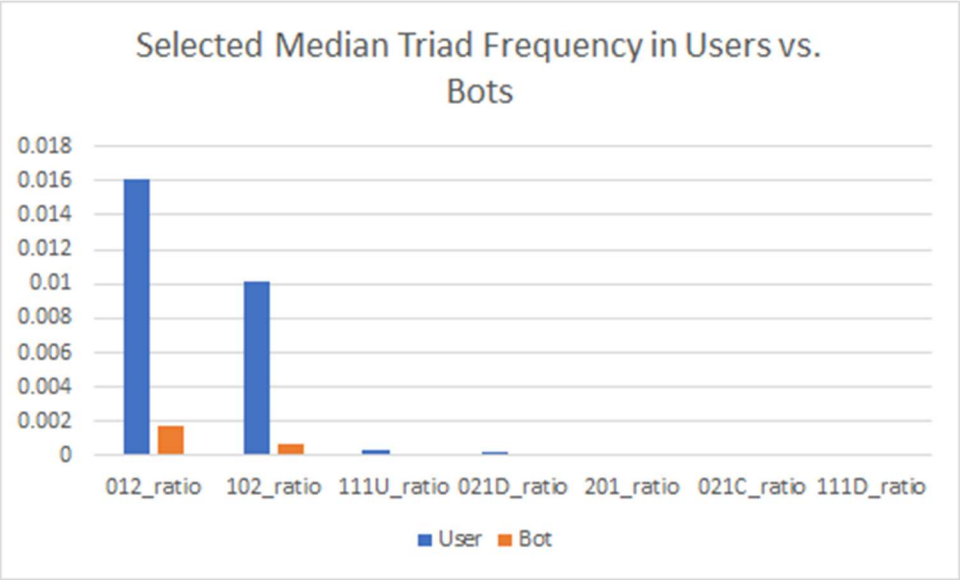

Figure 3. Bar chart indicating the relative ratios of graph Triads for Users and Bots.

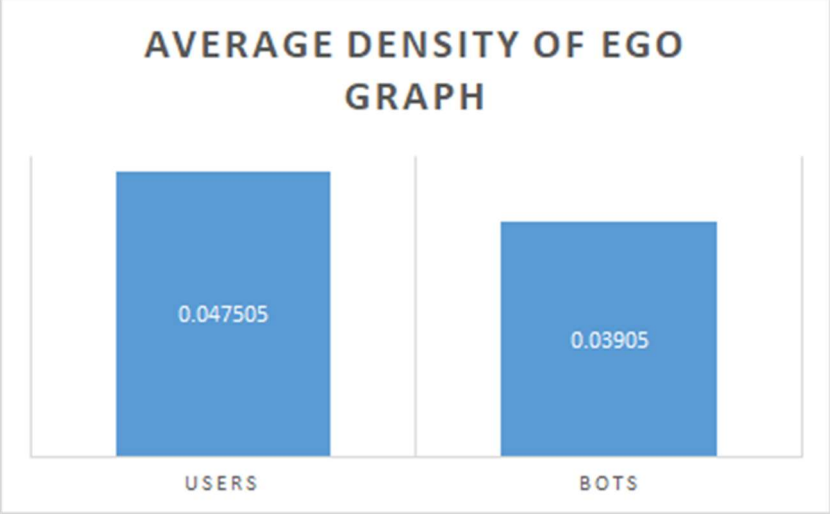

Figure 4. Bar chart indicating the average density of the Ego Graph for Users and Bots.

Using the metadata and network features, we trained random forest and support vector machine models on a subset of our dataset, and then used the trained models to classify users in a held-out test set as either bots or humans. We determined the accuracy of the models by calculating the percentage

of bots that were correctly classified as bots (“sensitivity”), and the percentage of humans that were correctly classified as humans (“specificity”). Our final “accuracy” metric is the average of the sensitivity and specificity of the model. We used the accuracy to tune hyperparameters for the algorithms and identify the set of hyperparameters that gives us the best accuracy.

The reason for using the average of sensitivity and specificity to judge the strength of the model as opposed to the general accuracy (number of correctly classified users divided by number of users in the test set) is that there are many more humans in our dataset then bots. With a dataset of 90% humans, a model that predicted that each new user was a human would have 90% accuracy. Using the average of sensitivity and specificity as our accuracy metric, that model would get 50% accuracy (just as a model that made predictions at random would). Therefore our baseline accuracy to compare our model to is 50%.

Results

Our initial analysis revolves around identifying features that indicate bot behavior. Because our goal is to be able to predict whether or not a user is a bot based on network characteristics, we first attempted to do the same prediction based on non-network characteristics. We used these results as a baseline to determine if using network analysis improves prediction ability.

Prediction Accuracy of Various Model/Feature Set Combinations		
	Random Forest	*Support Vector Machine*
Text Features Only	*0.681*	*0.781*
Network Features Only	*0.570*	*0.775*
All Features Combined	*0.630*	*0.801*

Figure 5. Accuracy of random forest and support vector machine models. Accuracy using the hyperparameter combination that predicts with the best accuracy is presented. The number displayed is the average of the accuracy of 100 runs of the algorithm.

The results are shown in Figure 5. When using just the text metadata as features, the Random Forest model had an accuracy of 0.681, while the SVM did 0.781. When analyzing just the network features, the Random Forest model had a 0.57 accuracy, and SVM had 0.775 accuracy. Finally, in the combination model using both network and text features, the Random Forest had a 0.63 accuracy, and 0.801 for the SVM. Interestingly, the SVM model does far better than the Random Forest model. Additionally, the network features improved the SVM model, but did not improve the Random Forest model. Despite our best efforts we could not uncover the reason behind these outcomes.

Average Feature Importance *(Combined Text and Network Features with a Random Forest Classifier)*	
Number of Comments	*0.1185*
Ego Graph Density	*0.0969*
Ratio of Triad 102	*0.8889*
Standard Deviation of the Number of words per comment	*0.0878*
Ratio of Triad 012	*0.0820*
Mean Comment Score	*0.0799*

Figure 6. The top six features based on feature importance in the random forest classifier. Values are the average importance over 100 runs for each feature.

The feature importance results, shown in Figure 6, demonstrate a few things about the difference in how bots and humans act on Reddit. For one, bots comment more often than humans do. While this may seem like an obvious conclusion to reach, it is nevertheless important to consider that bots are able to digest post and comment information and react to it quickly and without a cost, demonstrating they are used so much in activities such as moderation and spamming.

Additionally, while humans comment less than bots, they continue participation in comment chains more often than bots do. This is shown by the fact that ego-density is higher for humans than bots. While bots typically leave their comment and don't respond throughout the rest of the comment chain, humans typically interact with other users who respond with them.

Challenges

The biggest challenge we faced was in determining the validity of the training and testing datasets. Because we were not sure whether the users inside the sets were actually bots or humans we had to manually determine its identity by reading through each user's comments. Users that we flagged as bots were users with comments that purport that they were generated automatically, or comments that included text such as "beep" or "boop" or even comments that had similar signatures in the comments, such as consistently commenting "Hi __, I'm dad".

Another related issue is the heuristic we used in flagging bots is not robust. We noticed that occasionally when users think they are responding to a bot (i.e. "Good bot" or "Bad bot"), they were

actually talking to a human user. However by adhering to this heuristic, we are ultimately relying on the judgement of Reddit users to identify who are bots or people. As a result, we were concerned that although we had picked up a number of bots in our initial flagging, that there were also a significant number of humans who were included in that set. We ultimately decided to hand-curate the flagged set of users in order to ensure that our classifier would be running on an labeled set of data that we deemed as accurate as possible.

Additionally, by far the most common post author identified in the dataset is “[deleted]”, which indicates that either the comment was deleted or the user who wrote the comment has been removed from the website. As this is a not a single user but rather a placeholder name for multiple accounts, we removed comments identified with this author from the network. However, as reddit communities are policed by volunteer users who have a vested interest in removing unhelpful posts, we suspect that a significant portion of malicious bot posts fall under the “[deleted]” author in our dataset. As a result, the majority of the bots which we were able to detect and label in the data are “helpful” bots, which may exhibit distinctly different text and network characteristics from non-helpful bots. A dataset with incorporates the original author of these deleted comments might provide more significant results in identifying malicious bots.

Conclusions

Our analysis was a foray into utilizing Reddit commenting behaviors from a network perspective to predict whether a user was a bot or a human. We conclude that adding network variables as features sometimes improves the accuracy measure, but this is dependent on the machine learning model. One point of improvement is in data gathering-- because our analysis was computed on imprecise datasets, this could have influenced the outcome of the study. A possible future project could be compiling a list of Reddit accounts which are known to be bots, but which don’t self-identify as bots. While this would be a time-consuming and difficult task, it would prove useful in training algorithms to identify users as nefarious bots.

Citations

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). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 273-274. DOI: https://doi.org/10.1145/2872518.2889302.

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). IEEE Press, Piscataway, NJ, USA, 620-627.

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 '16). IEEE Press, Piscataway, NJ, USA, 533-540.

A.H. Wang. 2010. *Detecting Spam Bots in Online Social Networking Sites: A Machine Learning Approach*. In: Foresti S., Jajodia S. (eds) Data and Applications Security and Privacy XXIV. DBSec 2010. Lecture Notes in Computer Science, vol 6166. Springer, Berlin, Heidelberg https://link.springer.com/chapter/10.1007/978-3-642-13739-6_25.

Appendix

A. Description of features used in analysis.

Text Based Features	
Number of comments	*The total number of comments posted by a user that were captured in our dataset.*
Mean score	*The average score (upvotes less downvotes) a user received on their posts.*
Standard Deviation of Minute Posted	*The standard deviation of the minute a user posted their comment, independent of the hour.*
Standard Deviation of the Number of Words	*The standard deviation of the total number of words a user's posts contained.*
Number of unique subreddits	*The number of individual subreddits a user posted comments in.*
Number of comments per unique subreddit	*The number of comments a user posted on average among all of the subreddits they were active in.*
Network Based Features	
Node Degree	*The total number other users this user interacted with*
Node In/Out Degree	*The number of other users who commented on a post by this user / this user commented on*
Average Degree of Neighbor Nodes	*The average number of users that users adjacent to this user interacted with*

Sum of In/Out Edge Weight	*The total number of comments made by this user in response to another user's comment / total number of comments made by other users in response to this user's comments*
Average of In/Out Edge Weight	*The average number of comments this user made as replies to another user / other users made as replies to this user*
Degree Centrality & In/Out Degree Centrality	*The number of users who interacted with / commented on / were commented on by this user, divided by the total number of users in the network*
Clustering Coefficient of the Ego Graph	*The number of triplets of users in a graph who all interacted with each other divided by the total number of potential such triplets.*
PageRank	*The relative "importance" of a given user in the network. How likely it is when traversing the network randomly to arrive at this user.*
Ego Density	*How complete the Ego graph of a given user is - will be higher if more users this user interacted with also interacted with each other*
Ratio of Triads	*For each of the 16 possible triads, what percentage of each is present among a nodes neighbors.*