[16]

BotOrNot: A System to Evaluate Social Bots

Clayton A. Davis[1,§] * Onur Varol[1,§]

Emilio Ferrara[2] Alessandro Flammini[1] Filippo Menczer[1]

[1]Center for Complex Networks and Systems Research, Indiana University, Bloomington, IN (USA)
[2]Information Sciences Institute, University of Southern California, Marina del Rey, CA (USA)

ABSTRACT

While most online social media accounts are controlled by humans, these platforms also host automated agents called social bots or sybil accounts. Recent literature reported on cases of social bots imitating humans to manipulate discussions, alter the popularity of users, pollute content and spread misinformation, and even perform terrorist propaganda and recruitment actions. Here we present *BotOrNot,* a publicly-available service that leverages more than one thousand features to evaluate the extent to which a Twitter account exhibits similarity to the known characteristics of social bots. Since its release in May 2014, *BotOrNot* has served over one million requests via our website and APIs.

Keywords

social bot; sybil account; social media

1. INTRODUCTION

A *social bot*, also known as a *sybil account*, is a computer algorithm that automatically produces content and interacts with humans on social media. These agents and their interactions have been observed in online social media for the past few years [3, 1]. Recenly DARPA organized a bot detection challenge to developed techniques for early detection of malicious organized activities [5]. Some bot accounts are entertaining, helpful, or at least harmless, but nefarious uses for social bots abound, especially when multiple bot accounts are used in a coordinated fashion to perform an orchestrated campaign. The adoption of social bots has been reported for the purpose of astroturf, that is creating the illusion of artificial grassroots support for political aims [4]. In another case, a bot campaign created fake "buzz" about a tech company: automated stock trading algorithms acted on this chatter, resulting in a spurious 200-fold increase in market price.[1] An extensive review about social bots and their roles in online social networks is presented in a forthcoming article [2].

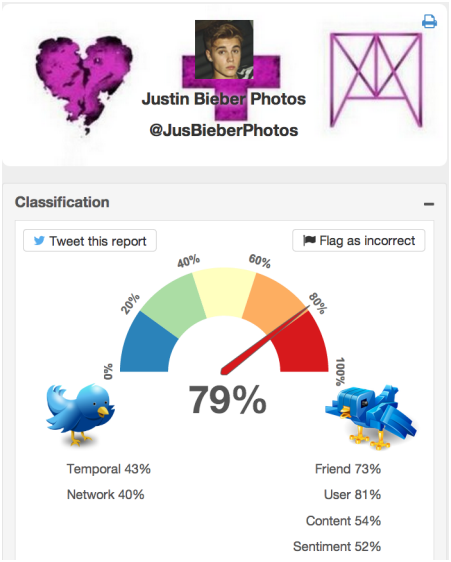

Figure 1: *BotOrNot* classification scores interface

In this paper, we present *BotOrNot*, our platform to evaluate whether a Twitter account is controlled by human or machine. This service is publicly available via the website[2] or via Python or REST APIs.[3,4] *BotOrNot* takes a Twitter screen name, retrieves that account's recent activity, then computes and returns a bot-likelihood score. For website users, this score is accompanied by plots of the various features used for prediction purposes. API tutorials can be found at the pages linked in the footnotes.

2. RELEASE TIMELINE

We made the *BotOrNot* web service public in May 2014. Initially our service was only available for users via the website — there was no public API due to capacity concerns. With the help of some press coverage, the service was used about 18k times in the first eight months. As part of a larger effort to address robustness issues causing occasional downtime, we noticed that certain IP addresses were using the service markedly more than others, so we implemented rate limits. System stability increased as a result of

*Contact: claydavi@indiana.edu
§Authors contributed equally.

[1]The Curious Case of Cynk, an Abandoned Tech Company Now Worth $5 Billion — mashable.com/2014/07/10/cynk

[2]truthy.indiana.edu/botornot
[3]github.com/truthy/botornot-python
[4]truthy.indiana.edu/botornot/rest-api.html

WWW'16 Companion, April 11–15, 2016, Montréal, Québec, Canada.
ACM 978-1-4503-4144-8/16/04.
Include the http://DOI string/url .

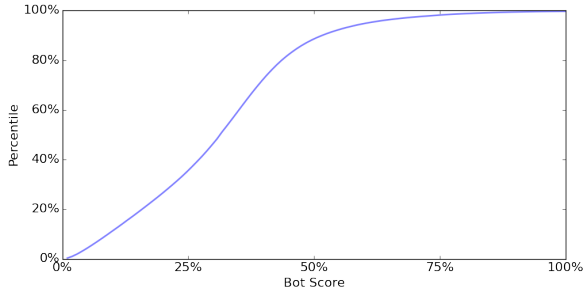

Figure 2: Cumulative distribution of bot scores

these changes. The added uptime had the unexpected consequence of increasing overall volume of use.

Analysis of usage by power users revealed that they were using the non-public internal API endpoint for the website. After a period of serving over 8k requests per day, we decided to explicitly allow programmatic access to *BotOrNot*. On 11 Dec, 2015, the *@TruthyBotOrNot* Twitter account announced the availability of our public API endpoint with higher rate limits. In the month since, we have served over 540k requests, bringing the total to over a million queries so far.

3. SYSTEM DESIGN

3.1 BotOrNot Service

The use of the *BotOrNot* service starts with a client specifying a Twitter screen name. The *BotOrNot* website and API use Twitter's REST API[5] to obtain the account's recent history, including recent tweets from that account as well as *mentions* of that screen name. Users are required to have a Twitter account in order for *BotOrNot* to make requests to Twitter's REST API on their behalf. Our API matches Twitter's rate limit of 180 requests per 15 minutes. Once the requested data is received from Twitter's API, the *BotOrNot* website or API forwards it to the *BotOrNot* server.

The server computes the bot-likelihood score using the classification algorithm described below. If the request originates from the website, the server generates data for the plots to be displayed and returns the resulting report with plots (Fig. 1). API users receive the classification results in JSON format suitable for post-processing.

While *BotOrNot* does not collect data about the users submitting the requests, we do store the computed classification results. As a result, we now have over 900k unique user account classifications. A distribution of bot scores is shown in Fig. 2. We plan to use these collected results to improve the classifier in the future.

3.2 Classification System

BotOrNot's classification system generates more than 1,000 features using available meta-data and information extracted from interaction patterns and content. We can group our features into 6 main classes: **Network** features capture various dimensions of information diffusion patterns. We build networks based on retweets, mentions, and hashtag co-occurrence, and extract their statistical features, *e.g.* degree distribution, clustering coefficient, and centrality measures. **User** features are based on Twitter meta-data related to an account, including language, geographic locations, and account creation time. **Friends** features include descriptive statistics relative to an account's social contacts, such as the median, moments, and entropy of the distributions of their number of followers, followees, posts, and so on. **Temporal** features capture timing patterns of content generation and consumption, such as tweet rate and inter-tweet time distribution. **Content** features are based on linguistic cues computed through natural language processing, especially part-of-speech tagging. **Sentiment** features are built using general-purpose and Twitter specific sentiment analysis algorithms, including happiness, arousal-dominance-valence, and emoticon scores.

To classify an account as either social bot or human, the model is trained with instances of both classes. As a proof of concept, we used the list of social bots identified by Caverlee's team [3]. We used the Twitter Search API to collect up to 200 of their most recent tweets and up to 100 of the most recent tweets mentioning them. This procedure yielded a dataset of 15k manually verified social bots and 16k legitimate (human) accounts. We used this dataset consisting of more than 5.6 millions tweets to train our models and benchmark classification performance.

BotOrNot's classifier uses Random Forest, an ensemble supervised learning method. Extracted features are leveraged to train seven different classifiers: one for each subclass of features and one for the overall score. Ten-fold cross-validation yields a performance of 0.95 AUC (Area Under ROC Curve). Note that such a high accuracy is likely to overestimate current performance, given the age of the training data.

4. CONCLUSION

In offering a free social bot evaluation service, we aim to lower the entry barrier for social media researchers, reporters, and enthusiasts. Ready-made reports on individual users are available via our website, or one can use our API to easily check multiple accounts, up to the rate limit. While using the API does require some scripting experience, using our service lets users skip the significant step of setting up their own classifiers.

One example application for our service would be a browser plugin adding a context menu option to fetch the *BotOrNot* report for a selected Twitter username. We welcome such applications from the social media community on top of our public bot classification service.

Acknowledgments. This work was supported in part by NSF (grant CCF-1101743), DARPA (grant W911NF-12-1-0037), and the J.S. McDonnell Foundation.

5. REFERENCES

[1] Y. Boshmaf, I. Muslukhov, K. Beznosov, and M. Ripeanu. Design and analysis of a social botnet. *Computer Networks*, 57(2):556–578, 2013.

[2] E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini. The rise of social bots. *Commun. ACM*, in press. Preprint arXiv:1407.5225.

[3] K. Lee, B. D. Eoff, and J. Caverlee. Seven months with the devils: A long-term study of content polluters on Twitter. In *Proc. 5th AAAI Intl. Conf. on Web and Social Media (ICWSM)*, pages 185–192, 2011.

[4] J. Ratkiewicz, M. Conover, M. Meiss, B. Gonçalves, S. Patil, A. Flammini, and F. Menczer. Truthy: mapping the spread of astroturf in microblog streams. In *Proc. 20th ACM Intl. World Wide Web Conf. Companion (WWW)*, pages 249–252, 2011.

[5] V. Subrahmanian, A. Azaria, S. Durst, V. Kagan, A. Galstyan, K. Lerman, L. Zhu, E. Ferrara, A. Flammini, F. Menczer, et al. The darpa twitter bot challenge. *arXiv preprint arXiv:1601.05140*, 2016.

[5] `dev.twitter.com/rest/public`