ABSTRACT

The impact of online social media on societal events and institutions is profound, and with the rapid increases in user uptake, we are just starting to understand its ramifications. Social scientists and practitioners who model online discourse as a proxy for real-world behavior often curate large social media datasets. A lack of available tooling aimed at non-data science experts frequently leaves this data (and the insights it holds) underutilized. Here, we propose birdspotter — a tool to analyze and label Twitter users —, and birdspotter.ml — an exploratory visualizer for the computed metrics. birdspotter provides an end-to-end analysis pipeline, from the processing of pre-collected Twitter data to general-purpose labeling of users and estimating their social influence, within a few lines of code. The package features tutorials and detailed documentation. We also illustrate how to train birdspotter into a fully-fledged bot detector that achieves better than state-of-the-art performances without making Twitter API calls, and we showcase its usage in an exploratory analysis of a topical COVID-19 dataset.

1 INTRODUCTION

Barely a decade old, social media in general — and Twitter in particular — are becoming increasingly important in shaping societal events. They serve as novel fora for a wide array of users to express themselves, discuss, promote agendas and attempt to influence the said societal events. As a result, social and political scientists, journalists, and communication scientists increasingly turn to social media as a proxy to study society. They carefully curate and label large social media datasets, and here a gap emerges. There is a limited offer of tools aimed at non-machine learning experts to analyze users in already existing datasets without making additional web API calls that limits the amount of retrieved data. This paper fills this gap by proposing birdspotter, a package aimed at non-computing practitioners with quantitative expertise (basic R or Python), to analyze, describe, and automatically label users in Twitter datasets.

This work addresses three specific open questions concerning analyzing Twitter users. The first question relates to the availability of user analysis tools. Existing tools are typically designed for Twitter branding and management, i.e. to either analyze a user’s or organization’s account (Twitter Analytics6), or Brandwatch Consumer Research7), or one given user account (Account Analysis Tool8). The question is whether a tool exists to retrospectively analyze and label all the users in Twitter dumps, aimed at non-data science experts with computational expertise? We address this question by proposing birdspotter1, an integrated Twitter user analysis tool, that can achieve three types of analysis in only a couple of lines of code. First, it processes existing Twitter datasets (e.g. jsonl data dumps collected through the Streaming API). Second, it describes users using three types of features (relating to the user, content semantics, and hashtag usage). Last, it allows training a classifier against a labeled user subset, which turns birdspotter into a general-purpose inferential user analysis tool.

1birdspotter source code, tutorial, and feature list: https://github.com/behavioral-ds/BirdSpotter
2birdspotter documentation: https://birdspotter.readthedocs.io
3COVID-19 tutorial: https://github.com/behavioral-ds/user-analysis
4Supplementary Material: https://arxiv.org/pdf/2012.02370.pdf#page=5
5Twitter Analytics: https://analytics.twitter.com/
6Brandwatch Consumer Research: https://www.brandwatch.com
7Account Analysis Tool: https://accountanalysis.app
We address this question by proposing APIs which are rate-limited by Twitter, 2) lack of reproducibility, (over deep learning) due to the heterogeneity of bots and the relative information?

knowledge of the social graph (often prohibitively expensive to analyzing the online social graph [8], and modeling information diffusion by predicting bots on pre-collected Twitter dumps. botometer since deactivated, protected, and suspended users can no longer be used, and recent activity-related features to train a Random Forest (formerly botometer) achieves a higher performance than the current state-of-the-art birdspotter 

tactic sparsity of training data. The de-facto bot detection tool is birdspotter, a tool — a software package designed for inferential analysis of online users in pre-collected data, and to estimate online user influence based on the reshare cascades.

The third open question is can we visualize and explore both broad and specific views of Twitter users and their activity? We address this question by proposing birdspotter.ml, a tool that provides both broad views of the user population and detailed inspections of user activity (see Fig. 1 for the main interface).

The main contributions of this paper are as follows:

- birdspotter — a software package designed for inferential analysis of online users in pre-collected data, and to estimate online user influence based on the reshare cascades.
- birdspotter.ml — an online visualizer designed to perform exploratory analysis of Twitter users.
- an offline bot detector, built using four public labeled datasets; we show that it achieves better than state-of-the-art performance and we showcase it on an example analysis of users discussing COVID-19.

Related work. Here, we present the prior work most relevant to birdspotter. For a complete related work discussion, please refer to the online appendix.

Tree-based ensemble methods dominate social bot detection (over deep learning) due to the heterogeneity of bots and the relative sparsity of training data. The de-facto bot detection tool is botometer (formerly BotOrNot) [4], which uses more than 1000 user- and recent activity-related features to train a Random Forest classifier. The main limitations of botometer are 1) usage of online APIs which are rate-limited by Twitter, 2) lack of reproducibility, since deactivated, protected, and suspended users can no longer be retrieved, and 3) botometer scores are likely to vary with user activity and botometer versioning. Birdspotter addresses the above by predicting bots on pre-collected Twitter dumps.

User influence is typically measured using static user attributes [3], analyzing the online social graph [8], and modeling information diffusion [14]. Closest to our work is ContInEst [5], which requires knowledge of the social graph (often prohibitively expensive to obtain) on which it performs random walks (very slow on large social graphs). Birdspotter estimates user influence from resharing dynamics in the absence of knowledge about the social graph.

2 PRELIMINARIES

In this section, we briefly outline prerequisites concerning influence estimation using point-process models. For a thorough construction of the influence estimation, please refer to the online appendix.

User influence estimation. birdspotter implements the algorithm in [9], estimating online influence as the mean number of retweets generated, directly and indirectly, by a user’s (re)tweet. Rizoiu et al. [9] estimate user influence, absent of the retweet branching structure, by assuming that retweets arrive following a Hawkes point process [10].

They estimate the probability that the tweet $v_j = (m_j, t_j)$ is a direct retweet of $v_i$ as $p_{ij} = \frac{\phi(m_{ij}, t_{ij})}{\sum_{k \neq i} \phi(m_{ij}, t_{ij})}$, where $m_{ij}$ is the associated user’s follower count, $t_j$ is the time of the event, and $\phi(m, \Delta t) = \frac{1}{\Delta t} \exp{-m^\beta} (\exp{-m^\beta} - 1)$ is the marked Hawkes exponential kernel of parameters $\kappa$, $\beta$ and $\theta$. The pairwise influence represents the probability that $v_i$ indirectly generates $v_j$, and is computed as $r_{ij} = \sum_{k=1}^{i-1} r_{ik} p_{kj}$ when $i < j$, $r_{ii} = 1$, and is 0 otherwise. Furthermore, a tweet’s influence is the sum of its pairwise influences, and a user’s influence is its tweets’ influences averaged.

3 PACKAGE OVERVIEW

In this section, we give an overview of birdspotter and birdspotter.ml, and describe their usage, functionalities, and design.

3.1 birdspotter

birdspotter labels users and measures influence on previously collected tweets in the standard jsonl or json format. Measuring influence. birdspotter measures user influence as outlined in Section 2, using by default a marked Hawkes exponential kernel with parameters $\kappa = \frac{1}{4}$ and $\theta = 6.8 \times 10^{-4}$. These were tuned on a large collection of real cascades [9], and can be customized using the function getInfluenceScores().

Usage and functionalities. Given a dataset of tweets collected externally (e.g. leveraging the Twitter Filter API), birdspotter’s core functionality revolves around two steps. In the first step, birdspotter loads the Twitter dataset, extracts retweet cascades, and compiles the user-level information. In the second step, it performs the influence analysis and user labeling. The former is achieved by simply invoking the BirdSpotter constructor, while the latter is achieved by calling the function getLabeledUsers(), which returns a table with the user features detailed above. For every observed cascade, birdspotter also computes the most likely branching structure (see $p_{ij}$ in Section 2). This can be achieved using the function getExpectedParents() and getExpectedParent(), which returns the reshare cascades (i.e. original tweet and all its retweets) with the additional column expected_parent indicating a retweet’s most likely parent tweet.

For power users, birdspotter provides a number of robust configurations — such as changing the parameters of the Hawkes kernel or using user-defined word embeddings — documented using its readthedocs documentation. A usage tutorial is available on birdspotter’s repository. For users who prefer to analyze the results outside python, birdspotter can dump the user table and the
3.2 birdspotter.ml visualiser

birdspotter.ml is a visualizer built on top of birdspotter, and designed to analyze Twitter users engaged in online discussions. The visualisation provides both broad and specific views of the data, via the three components shown in Fig. 1: a scatter plot component, a user information component, and a cascade view component.

The Scatter Plot. The left panel contains the scatter-plot showing the influence percentile (on the y-axis) and botness (on the x-axis) of a random sample of the users from the dataset, and the underlying 2-D density over the entire data set. Users are colored based on the hashtag they use most and, when selected, the user and cascade views are populated. The plot is pan-able and zoom-able. The view starts with a random sample of 1,000 users and is dynamically populated as practitioners explore cascades.

The User View. The top-right panel provides information about a selected user, including their Twitter image (hyperlink to the user’s profile), screen name, location, the hashtags they used, and basic Twitter metrics (such as the number of followers or tweets).

The Cascade View. The bottom-right panel shows the cascades the selected user participates in, which are select-able via a carousel. The component shows the text of the original tweet, the retweets’ timing, and the most-likely branching structure inferred using birdspotter. The points on this component are select-able and hover-able in the same way as the scatter plot. The component also is pan-able and zoom-able.

4 BUILDING A BOT DETECTOR

In this section, we train birdspotter as a bot classifier with better performances than the state of the art botometer. We showcase birdspotter to profile a topical COVID-19 Twitter dataset.

Training data. birdspotter provides the functionality to retrain and update the current model, which we leverage to build a bot detector. We train on four public bot datasets, including [botometer-feedback-2019, political-bots-2019] [12], and {verified-2019, botwiki-2019} [13], sourced from Bot Repository9.

Training. The Bot Repository only provides account-level data, whereas birdspotter is designed to utilize tweet jsons. We use the tool twarc to acquire the timeline of each available user’s first 200
tweets, to construct jsonl training data. We extract and preprocess the data with BirdSpotter(), label the resulting dataframe with users' ground truth values, and run trainClassifierModel() on this training data to acquire our final model. We ship this model as the default at birdspotter’s installation.

**Botometer comparison.** We compare the derived model against botometer, by acquiring their bot scores (universal CAP [12]) for available users through their API. Fig. 2a shows that birdspotter out-performs botometer in terms of mean AUC, despite using less information to make predictions – botometer uses more user features extracted from the online API. Fig. 2b shows that birdspotter consistently out-performs botometer with respect to mean $F_1$ scores, over all bot score thresholds.

**Ablation study.** We test the importance of each set of features through various ablations of our main model. Fig. 2a shows the mean AUC obtained for subsets of features. It shows that Twitter user features and semantic features are both informative of bot-like behavior, while hashtag features show more variation. The hashtag model performance may be an artifact of training on the mixture of bot datasets (containing hashtags relating to different topics). We retain hashtag features in birdspotter, for better generalizability when users train and test on their own domain datasets. The best performing model uses Twitter user features and semantic features.

**SHAP analysis.** We use shap [6] for explaining the impact of features in our tree ensemble model. Fig. 2c shows that the Twitter user features form the majority, and semantic features a minority of the impactful features, in line with the ablation study.

**COVID-19 Application Dataset.** We apply birdspotter to a COVID-19 dataset [1], supplied as tweet IDs which were re-hydrated with twarc to a jsonl format, recovering 68.8%. We limit our analysis to the ~1.5M unique tweets relating to posts on January the 31st, resulting in ~0.28M users and ~0.42M cascades.

**Dataset profiling.** Fig. 3b shows the empirical distributions of botness, influence, and activity (i.e., the number of cascades a user participates in). The distribution of botness indicates two maxima; the larger indicating the humans and the smaller indicating the bots. Conforming with the literature [9], influence and activity are long-tailed (following a "rich-get-richer" paradigm).

**Labeling Users.** Exploring birdspotter.m1 we observe humans @DumplingSays, @eddissant, and @marat_dospolov — with bot scores of 0.873, 0.83, and 0.925 respectively. Using getAnnotationTemplate (see Fig. 3a, line 8) we label each user as human, and update the classifier with trainClassifier (Fig. 3a, line 10). The new bot scores are 0.375, 0.296, and 0.559, respectively. Practitioners can use birdspotter to classify any latent user attribute.

**5 CONCLUSION**

We presented birdspotter, a Twitter user analysis tool aimed at non-data science experts who analyze discourse and user activity on social media. It provides an end-to-end analysis of users’ online characteristics, and populates a visualizer facilitating both broad views of a user population and individual exploration. As with many open-source classifiers, we know that birdspotter could be leveraged to infer sensitive features. However, we are currently not aware of any protections that we could implement to prevent this.

Tools like birdspotter are integral to the timely, performant, and reproducible analysis of social media users for understanding discourse and society.

**Acknowledgments**

This research was partially funded by Facebook Research under the Content Policy Research Initiative grants and the Defence Science and Technology Group of the Australian Department of Defence.

**REFERENCES**

[1] Emily Chen, Kristina Lerman, and Emilio Ferrara. 2020. Tracking Social Media Discourse About the COVID-19 Pandemic: Development of a Public Coronavirus Data Set. JMR (2020).

[2] Tianq Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In KDD ’16.

[3] Jean-Valère Costu, Vincent Labatut, and Nicolas Dugue. 2016. A review of features for the discrimination of twitter users: Application to the prediction of offline influence. SNAM (2016).

[4] Clayton Allen Davis, Onur Varol, Emilio Ferrara, Alessandro Flammini, and Filippo Menczer. 2016. Botometer: A system to evaluate social bots. In WWW.

[5] Manuel Gomez-Rodriguez, Le Song, Nan Du, Hongyan Zha, and Bernhard Schölkopf. 2016. Influence Estimation and Maximization in Continuous-Time Diffusion Networks. ACM Transactions on Information Systems (2016).

[6] Scott M Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In NeurIPS’17.

[7] Tommas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in Pre-Training Distributed Word Representations. In LREC 2018.

[8] Fabián Riquelme and Pablo González-Cantergiani. 2016. Measuring user influence on Twitter: A survey. Information processing & management (2016).

[9] Mariam-Andrei Rizoiu, Timothy Graham, Rui Zhang, Yilei Zhang, Robert Ackland, and Lexing Xie. 2018. # DebateNight: The Role and Influence of Socialbots on Twitter During the 1st 2016 US Presidential Debate. In ICWSM.

[10] Mariam-Andrei Rizoiu, Young Lee, Swapnil Mishra, and Lexing Xie. 2017. A Tutorial on Hawkes Processes for Events in Social Media. In Research Frontiers of Multimedia.

[11] Mohsen SayyadAhkandeh, Onur Varol, Kai-Cheng Yang, Alessandro Flammini, and Filippo Menczer. 2020. Detection of Novel Social Bots by Ensembles of Specialized Classifiers. CIKM (2020).

[12] Kai-Cheng Yang, Onur Varol, Clayton A Davis, Emilio Ferrara, Alessandro Flammini, and Filippo Menczer. 2019. Arming the public with artificial intelligence to counter social bots. Human Behavior and Emerging Technologies (2019).

[13] Kai-Cheng Yang, Onur Varol, Pak-Mai Hui, and Filippo Menczer. 2020. Scalable and generalizable social bot detection through data selection. In AAAI.

[14] Zizhu Zhang, Weiliang Zhao, Jian Yang, Cecile Paris, and Surya Nepal. 2019. Learning Influence Probabilities and Modelling Influence Diffusion in Twitter. In WWW.

Figure 3: Quantifying user botness and influence analysis on COVID-19 dataset. (a) Code required to load a Twitter dump, generate cascade and user information, annotate and fine-tune the bot classifier. (b) A density plot of user botness scores, and complementary cumulative density plots (CCDF) of user influence and user activity. The red lines show the mean values.
A ADDITIONAL RELATED WORK

In this section, we outline other approaches to bot detection and influence measurement in the literature.

Detecting Twitter bots. There have been a myriad of approaches to detect bots on Twitter. There are three motifs within the literature. The first motif are supervised methods used to determine if an individual user is a bot, usually employing feature construction. Such approaches include NLP approaches [4, 13], deep-learning approaches [14], feature-engineering [3, 6, 23] and other methods [9, 15]. The second motif are unsupervised methods used to discover coordinated online behavior/real-time online campaigns; and the third motif are adversarial methods which achieve better bot detection by generating better bots.

birdspotter falls in the first category. It uses a supervised approach to retrospectively analyze datasets. It satisfies a different use case than coordinated online behavior tools like BotSLayer [12]. Adversarial approaches are fairly novel, however it is unclear whether they might simply improve bot technology, as they provide recipes to build better bots.

The de-facto bot detection tool in the social science community is Botometer (formerly BotOrNot) [6], which uses more than 1000 user- and recent activity-related features to train a Random Forest classifier. Botometer is currently at version 4, at the time of writing, and serves half a million queries a day [21].

The main limitation of botometer for practitioners is its dependence on an online API. It cannot be used to profile the users in offline Twitter datasets which have been collected in the past (like used in many works [1, 8, 22]). Furthermore, the API is rate-limited by Twitter, and requires registration through both Twitter and RapidAPI service. For scientific purposes, botometer makes local reproducibility difficult to achieve, since deactivated, protected, and suspended users can no longer be retrieved, and botometer scores are likely to vary with user activity and botometer versioning.

Birdspotter addresses the above-stated shortcomings by producing bot predictions on already collected Twitter dumps, and exposing a simple interface to allows researchers to annotate their own Twitter user collection.

Tools for quantifying online influence. There are many features used to score the influence, reputation or popularity of online users. We delineate these into three areas: those using static user attributes (including lexical features and information on a user’s profile) [5], those that analyze the online social graph (e.g. degree, PageRank, HITS, etc) [2, 18], and those modeling information diffusion [25]. However, few of these have translated into accessible tools for the non-experts in the field. For instance, Cossu et al. [5] provide a set of scripts to perform their influence measurement method. Other tools, like ConTinEst [7, 10], require knowledge of the social graph (which is often prohibitively expensive to obtain) on which it performs random walks (which are very slow on large social graphs). Birdspotter estimates user influence from reshare dynamics, in the absence of knowledge about the social graph, and provides an end-to-end tool to analyze Twitter users.

B INFLUENCE MEASURE

We review the theoretical prerequisites concerning modeling reshare cascades using point processes, and estimating reshare influence.

Reshare cascades. birdspotter analyzes the spread of online information in the form of online reshare cascades. A reshare cascade consists of an initial user post and some reshare events of the post by other users. On Twitter, for example, this can happen when users use the retweet functionality. We denote a cascade observed up to time $T$ as $\mathcal{H}(T) = \{t_i, t_i, \ldots\}$ where $t_i \in \mathcal{H}(T)$ are the event times relative to the first event ($t_0 = 0$). We denote cascades with additional information about events — dubbed here as event marks — as marked cascades. We use the notation $\mathcal{H}_m(T) = \{(t_0, m_0), (t_1, m_1), \ldots\}$, where each event is a tuple of the event time and the event mark. For example, for retweet cascades, the numbers of followers of a Twitter user are commonly adopted as event marks [16, 17, 26].

The Hawkes processes. birdspotter models reshare cascades using Hawkes processes [11] — a type of point processes with the self-exciting property, i.e., the occurrence of past events increases the likelihood of future events. The occurrence of events in a Hawkes process is controlled by the event intensity function:

$$\lambda(t | \mathcal{H}(T)) = \mu(t) + \sum_{t_i < t} \phi(t - t_i)$$

where $\mu(t)$ is the background intensity function and $\phi: \mathbb{R}^+ \to \mathbb{R}^+$ is a kernel function capturing the decaying influence from a historical event. We note that, for reshare cascades, all events are considered to be offspring of the initial event, i.e. there is no background event rate $\mu(t) = 0$. Two widely adopted parametric forms for the kernel function $\phi$ include the exponential function $\phi_{\text{EXP}}(t) = k \theta e^{-\theta t}$ and the power-law function $\phi_{\text{PL}}(t) = k (t + c)^{-1(\theta + 1)}$.

Marked Models. birdspotter implements marked versions of the point processes, where the mark is the number of followers that the user emitting the tweet has. This is because the mark of each event governs the number of future events, e.g., a tweet from a largely followed user is likely to attract more retweets. The marked versions of Hawkes processes [16] are then derived by rescaling the kernel functions with the marks, i.e., $\phi(m, t) = m^\beta \phi(t); \beta$ controls the warping effect of the mark.

User influence estimation. birdspotter adopts the following definition for user influence, widely used in literature [7, 19, 24]:

Definition B.1. Online user influence $\varphi(u)$ is defined as the mean number of reshares generated directly and indirectly by a message posted by $u$, irrespective if it is an original message or a reshare.

Estimating influence from retweet cascades has the additional difficulty of not observing the branching structure of the diffusion — i.e., the Twitter API attributes all retweets to the original tweet. birdspotter estimates Twitter user influence using only the observed retweet cascade $\mathcal{H}_m(T) = \{v_0 = (t_0, m_0), v_1 = (t_1, m_1), \ldots\}$, where marks correspond to users’ number of followers.

Rizoiu et al. [19] propose a method to estimate user influence in the absence of the branching structure by assuming that retweets arrive following a Hawkes point process [20]. We can quantify the probability that an event $v_i$ is generated by a previous event $v_j$ as the ratio of the event intensity generated by $v_j$ and the total

Accompanying the submission Birdspotter: A Tool for Analyzing and Labeling Twitter Users.
intensity at time $t_j$. Formally, the probability $p_{ij}$ that user $v_i$ retweets $v_j$ is
\[ p_{ij} = \frac{\phi(t_j - t_i)}{\sum_{k=1}^{\infty} \phi(t_j - t_k)} \tag{2} \]

Rizoiu et al. [19] also introduce the pairwise influence score $m_{ij}$, intuitively defined as the amount of influence that $v_j$ exerts over $v_i$ either directly (when $v_i$ is a direct retweet of $v_j$) or indirectly (when $v_i$ is a retweet of a descendant of $v_j$):
\[ m_{ij} = \begin{cases} \sum_{k=1}^{j-1} m_{ik} p_{kj}, & i \leq k < j \\ 1, & i = j \\ 0, & i > j \end{cases} \tag{3} \]

Finally, the influence of $v_j$ is $\varphi(v_j) = \sum_{k=1}^{\infty} m_{jk}$, and the influence of a user $u$ is the average of the influences of all of their tweets:
\[ \varphi(u) = \frac{\sum_{v \in T(u)} \varphi(v)}{|T(u)|} \tag{4} \]

where $T(u)$ is the set of all the tweets emitted by user $u$.

REFERENCES
[1] Alessandro Bessi and Emilio Ferrara. 2016. Social bots distort the 2016 US Presidential election online discussion. First Monday (2016).
[2] Meyoung Cha, Hamed Haddadi, Fabrizio Benevenuto, and Krishna P Gummadi. 2010. Measuring user influence in twitter: The million follower fallacy. In ICWSM.
[3] Zi Chu, Steven Gianvecchio, Aaron Koehl, Haining Wang, and Sushil Jajodia. 2015. Sifting robotic from organic text: a natural language approach for detecting automation on Twitter. Journal of Computational Science (2016).
[4] Jean-Valère Cossu, Vincent Labatut, and Nicolas Dugué. 2016. A review of features for the discrimination of twitter users: Application to the prediction of offline influence. SNAM (2016).
[5] Eric M Clark, Jake Ryland Williams, Chris A Jones, Richard A Galbraith, Christopher Torres-Lugo, Kai-Cheng Yang, Alessandro Flammini, and Filippo Menczer. 2016. Botornot: A system to evaluate social bots. In WWW (2016).
[6] Zizhu Zhang, Weiliang Zhao, Jian Yang, Cecile Paris, and Surya Nepal. 2019. Learning Influence Probabilities and Modelling Influence Diffusion in Twitter. In WWW (2019).
[7] Qingyuan Zhao, Murat A. Erdogdu, Hera Y. He, Anand Rajaraman, and Jure Leskovec. 2015. SEISMIC: A Self-Exciting Point Process Model for Predicting Tweet Popularity. In KDD (2015).