BotCamp: Bot-driven Interactions in Social Campaigns

Noor Abu-El-Rub
University of New Mexico
nabuelrub@unm.edu

Abdullah Mueen
University of New Mexico
mueen@unm.edu

ABSTRACT

Bots (i.e. automated accounts) involve in social campaigns typically for two obvious reasons: to inorganically sway public opinion and to build social capital exploiting the organic popularity of social campaigns. In the process, bots interact with each other and engage in human activities (e.g. likes, retweets, and following).

In this work, we detect a large number of bots interested in politics. We perform multi-aspect (i.e. temporal, textual, and topographical) clustering of bots, and ensemble the clusters to identify campaigns of bots. We observe similarity among the bots in a campaign in various aspects such as temporal correlation, sentimental alignment, and topical grouping. However, we also discover bots compete in gaining attention from humans and occasionally engage in arguments. We classify such bot interactions in two primary groups: agreeing (i.e. positive) and disagreeing (i.e. negative) interactions and develop an automatic interaction classifier to discover novel interactions among bots participating in social campaigns.

CCS CONCEPTS

• **Information systems → Online advertising**; **Web mining**; *Clustering*; Social networking sites.

KEYWORDS

Bots Interaction; Campaign Detection; Influence; Cluster Ensemble; Graph Mining

ACM Reference Format:
Noor Abu-El-Rub and Abdullah Mueen. 2019. BotCamp: Bot-driven Interactions in Social Campaigns. In *Proceedings of the 2019 World Wide Web Conference (WWW '19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3308558.3313420

1 INTRODUCTION

Social networking sites bring people closer to each other and facilitate fast and convenient information flow. However, modern social media sites suffer from user accounts that work towards fast and automated building of social capital and exploiting the social influence to sway public opinion. Such user accounts (commonly named as bots) perform scheduled posting [26], near-automated registrations [27], and chronological deletions [11] among many other unsocial and non-human behavior.

To multiply the effect, instead of creating super smart bots, botmasters employ a large number of naive bot accounts to attain their objectives. Not surprisingly, humans tend to believe repeatedly encountered information from diverse sources [22]. Thus, a swarm of bots can potentially run successful advertising campaigns to promote products, election campaigns to win races, and organizational campaigns to recruit for ideological groups. To understand the fullest potential of a swarm of bots, in this paper, we perform an empirical study on bot activities in social campaigns and develop a technique to detect and classify bot-driven interactions in social campaigns. Quantifying bot-driven interactions in social campaigns can be useful for political parties, advertising agencies, charitable organizations among many others. Early detection and characterization of bot participation in campaigns will help campaigns flourish organically.

Figure 1: An example of bot interactions. Politically motivated bots are discussing trend manipulation.

An example of bot behavior in Twitter at the time of U.S. Presidential Election in 2016 is given in Figure 1. The user account `@JaredWyand` was an active supporter of Trump campaign. The account has been detected by both DeBot [9] and Botometer [12] systems due to its high frequency of tweets and content similarity. The account is currently suspended by Twitter. The tweet shown here has been retweeted 1.2K times. The two other users copied the tweet shortly after that [6]. These users are also detected as bots by DeBot [1] *and* Botometer[2], however, they are not suspended by Twitter at the time of writing. The content of the tweets shows that bot accounts are promoting a specific topic in Twitter's ranking system by frequently tagging relevant hashtags. The content shows that bot accounts are tracking progress of competing political campaign.

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

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313420

[1] www.cs.unm.edu/~chavoshi/debot/
[2] https://botometer.iuni.iu.edu

Note that every bot account has a human owner who can post in natural language in between scheduled posts.

The example demonstrates that bot accounts collaborate towards an objective i.e. making a topic trending. It also demonstrates that bots exhibit negative sentiments towards competing campaigns.

In this work, we develop a system to detect bot-driven interaction in campaigns categorized by general topics. For example, we have detected five major campaigns interacting under the "U.S. Election 2016 topic". Three of the bot-driven campaigns are taking sides of the candidates. The objective of the two of the remaining bot-driven campaigns is to gain human attention by adopting popular topics such as U.S. Election. Our system, named **BotCamp**, continuously collects bots for a given topic and detects bots using the DeBot system [9]. BotCamp identifies bots that are posting similar content on the campaign topic, and accumulates such bots over a long time to create graph structures on various aspects such as retweets, mentions, shared media and hashtags. We develop a heuristic cluster ensembling approach to combine communities detected from these graphs, which leads to discovering bot-driven interactions.

In this work, we have collected bot activities related to social campaigns in U.S. election. We analyze the campaigns to understand their information flow and status after campaigns are over. All data and code are made public [2].

The rest of the paper contains a discussion section in related work and background (2), an overview section describing the framework (3), an experimental section showing ensembling and interaction classifier evaluation (4), and the last section concludes the work (5).

Disclaimer: We do not address the question, "who" create and operate bot accounts. We define bots as the accounts that show signs of automation. We collect empirical evidence of "how" bots are involved in social campaigns and reason about "why" bots are involved. To the best of our knowledge, this work is one of the very first to generalize bot interactions on social media.

2 RELATED WORK AND BACKGROUND

2.1 Related Work

Our work combines two independent streams of research on social media: campaign detection and bot detection. Campaign detection works mostly focus on finding campaigns based on *one* specific aspect of messaging, such as message similarity [19][18][23], URL bursts [20], retweet structure [17]. We combine several other messaging aspects such as mentions, hashtags, and media sharing. All of these works detect clusters/communities in some messaging graphs. Such communities may include both bots and humans, hence existing work cannot separate the bot-driven part of the campaign.

DARPA Bot Challenge suggests an estimated 15% of Twitter accounts are bots [25]. Existing bot detection techniques are either supervised [12][14] and unsupervised [9]. Since our goal is to identify campaign specific bots, we opt for an unsupervised technique, DeBot [9].

Bot activities related to campaigns have been studied previously that associated bot activities with political entities [16][3][24]. Our goal is to explore beyond politics to sports, entertainment, marketing, etc., at a much larger scale of thousands of bot accounts.

Bots have been categorized based on their roles as individual users, independent of campaigns they take part in [21]. In contrast, we categorize bots based on their type of interactions in social campaigns.

2.2 Social Campaigns

Definition of campaign has been diverse in the literature, mostly attributed as unethical and illegal cases of social campaign. For example, coordinated campaigns [19], spam campaigns [13], promoted campaigns [15], fraud campaigns [8], and incentivized campaigns [4, 5] are some of the characterizations of campaigns.

In general, we define a social campaign as a group of concepts aligned to an objective that a group of people want to achieve. For example, `#antivax` and `#autism` are concepts supported by people who want to abolish vaccination. Another example is a fund-raising campaign started by Peter Dunn (`@PeterThePlanner`) in Indiana to support homeless people immediately before a blizzard hit Indianapolis. $41K was raised organically from various organizations and individuals for `@WheelerMission`. Therefore, a social campaign should not be perceived as purely inorganic or organic. Instead, considering that both humans and bots are involved together, we propose to quantify the level of bot and human participation in social campaigns. Quantifying organic participation in campaigns can be useful for political parties, advertising agencies, charitable organizations among many others.

2.3 Bot Detection

Automated accounts, a.k.a *bots*, are tweeting/re-tweeting always. Bots are controlled by computer programs. There may exist automated accounts which are not harmful such as `@countforever`, but most bots pretend to be human, entice people to follow them, and/or share ideas. DeBot is a parameter-free unsupervised system [9], that constantly collects data from Twitter and detects bots based on their synchronicity at intervals of 180 minutes. Number of bots DeBot detects in an interval depends on the topic, time of day, bot presence and sampling rate. Note that, Twitter streaming API provides a 1% of sample. In a successful interval, DeBot detects few bot-clusters containing tens of bots.

DeBot is a near real-time system that exploits highly unusual activity correlation across users as an indicator of bot behavior. The authors show that even if millions of active users interact at a time instance, human users are not likely to have more than tens of synchronous postings at random [10]. Although we use DeBot as an integral part of the detection system, we can replace DeBot by any other topic-specific near real-time system.

3 BOT INTERACTIONS IN CAMPAIGNS

3.1 The Framework

Figure 2 shows the BotCamp framework. There are three components of the system: Keyword Generator, Campaign Detector and Interaction Detector. We describe each of the components below.

Keyword Generator: BotCamp continuously collects trending hashtags to maintain related keywords to a seed set of keywords. The motivation behind such a keyword generator is to adapt with

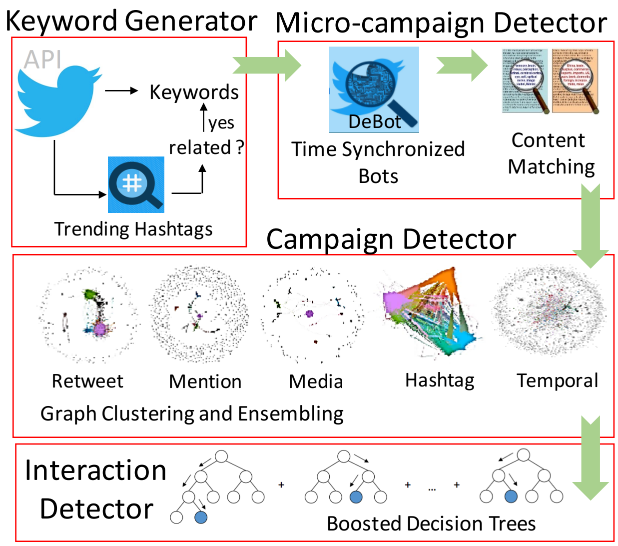

Figure 2: BotCamp framework

changing campaign dynamics. Trending keywords related to a campaign can be changed frequently. For example, to monitor the U.S. election, we started with a seed of twenty keywords including general topics such as `election`, `Trump`, `Clinton`, etc. After the U.S. election, the seed set grew to 231 keywords including `MAGA` ("short form of Make America Great Again"), `PodestaEmails`, and `CrookedHillary`. We collect the top 50 trends in twitter in every three-hour interval. If more than 50% of the tweets containing a trend also contains a seed keyword, we add the trend to seed set. In short campaigns, the seed set remains almost identical for the lack of dynamics in the campaigns. In long campaigns (i.e. election campaigns), keywords can be weighted based on their recency. The keywords can be both in support or in favor of parties involved in a campaign. We have labeled the sentiment associated with the keywords manually for the U.S. dataset.

Campaign Detector: We use the keywords in an instance of DeBot system that detects synchronized bots within an interval of three hours. We use the recommended threshold of 0.99 correlation to detect bots. DeBot outputs clusters of bots that we further analyze to detect clusters of bots that are both temporally synchronous and textually similar. BotCamp accumulates bots for a duration that is sufficient for the campaign to reach a stable state. We have accumulated at least one week of bots for all of our experiments. After bots are collected, we produce five graphs capturing various aspects of campaigns (e.g. retweet graph, hashtag graph, etc.). BotCamp detects communities in these graphs based on modularity optimization algorithm [7]. We develop a cluster ensembling technique that combines the communities from different aspects into consensus communities representing campaigns.

Interaction Detector: BotCamp consists of a classifier that categorizes the interaction between pairs of campaigns. The classifier is trained on a manually labeled set of interactions. We consider two types of interactions: *agreeing* and *disagreeing* interactions. We produce a set of 94 novel features that are indicative to various interaction types. The classifier is AdaBoost ensemble classifier, we use the classifier to categorize all possible pairs of interacting campaigns, and quantify bot participation in a campaign. Figure 3 shows examples of such interactions.

In the next two sections, we elaborate on the the campaign and interaction detectors.

3.2 Campaign Detector

Our campaign detection system is a two step process: Content matching and Graph clustering.

3.2.1 **Content Matching.** DeBot produces a set of unusually synchronous bots. Although a group of unlikely synchronous bots typically works towards a campaign, there can be spurious synchronous groups that are just naively periodic. In this step, we detect bots that are posting not only at close time instances, but also similar content. We consider each synchronous cluster detected by DeBot, and calculate the text and hashtag similarity among the bots in the cluster. *Text similarity* between two users u and v is defined by the Jaccard similarity of their set of unigrams. More precisely, if $G(u)$ is the set of unigrams extracted from the tweets made by u, excluding the stop words, the text similarity between u and v is:

$$SimText(u, v) = \frac{G(u) \cap G(v)}{G(u) \cup G(v)}$$

The similarity within a cluster C is

$$SimText(c) = \frac{\sum_{\forall u,v \in C} SimText(u, v)}{|C| * (|C| - 1)/2}$$

Hashtag similarity between two users u and v is defined by the Jaccard similarity of their set of hashtags. More precisely, if $H(u)$ is the set of unigrams extracted from the hashtags made by u, the hashtags similarity between u and v is:

$$SimHashtag(u, v) = \frac{H(u) \cap H(v)}{H(u) \cup H(v)}$$

The hashtag similarity within a cluster C is

$$SimHashtag(c) = \frac{\sum_{\forall u,v \in C} SimHashtag(u, v)}{|C| * (|C| - 1)/2}$$

We define a micro-campaign as a cluster of temporally synchronous bots, C, having either $SimText(C) \geq 0.5$ or $SimHashtags(C) \geq 0.5$. Note that such micro-campaigns are formed based on three hours of information.

3.2.2 **Graph Construction.** Once BotCamp accumulates micro-campaigns detected in three hour batches for over the duration of the campaign, the system creates five graphs namely: *retweet*, *media*, *hashtag*, *mention* and *temporal* graphs. The objective is to study the underlying interaction among micro-campaigns on various aspects over the duration of the campaign. Since the graphs are based on three hour long captures, the graphs are crude approximations of the graphs that we could produce if we had all data available. We describe each of these graphs below.

Retweet Graph

Retweets usually mean endorsement. Hence, we create a undirected retweet graph where nodes are bots, and we add an edge between two bots when they retweet from each other at least once, encoding

Figure 3: (left) Agreement interaction among bots. (middle and right) Disagreement interactions.

their mutual endorsements. In contrast, we can create a directed retweet graph by adding edges from the retweeting node to the original author node.

Mention Graph

In public conversations, bots mention (i.e. adding @ before an account name) other accounts in tweets. Mentions are typically used to draw attention of the person being mentioned. Thus, mentions are useful to express agreement, disagreement, endorsement, promotion, etc. We create a mention graph by adding an edge between two bots if they mentioned each other.

Media Graph

Bots in the same campaign proliferate the same information. Memes, photos, and videos are typically more expensive to create compared to tweets, however, such media are more attractive. Determined campaigns spend resources to create media and employ automated accounts to share the media. We create the media graph on bots by connecting two bots that share the exact same URL media.

Hashtag Graph

Hashtag is a powerful way to organize content for better searching. Information seekers often use hashtags to learn discussion items about a topic. Competing campaigns fight for strong position on common discussion topics (e.g. `#Oscars`). Campaigns also want to make hashtags trending (See Figure 1). Thus, tagging the same hashtag may mean either agreement or disagreement; at weaker level than mentions. If two bots have more than 50% of their hashtags common (i.e. $SimHashtags(u, v) = \frac{H(u) \cap H(v)}{H(u) \cup H(v)} \geq 0.5$), we add an edge between them to create the hashtag graph.

Temporal Correlation Graph

Synchronous bot activities indicate that bots using the same scheduler (e.g. a random posting interval generator or a human leader). We add an edge between two bots if they have been correlated at least once in their campaign lifetime for three hours interval regardless of their content similarity. Since we are using Debot we know that all bots are temporally correlated at least once with other bots, however, this graph exposes further correlations that could happen along the campaign life duration.

3.2.3 **Graph Clustering and Ensembling.** We consider building larger campaigns from the micro-campaigns by clustering the individual graphs mentioned in the previous section and ensembling the clusters across various aspects.

To cluster the graphs, we use a state-of-the-art technique called Louvain Modularity to cluster bots [7]. The algorithm uses greedy modularity optimization method and has linear complexity, thus it run fast on large dataset. We run the algorithm on the five graphs respectively. For each graph, we produce clusters of bots, therefore, each bot will belong to five clusters of various aspects.

Ensembling clusters from the five graphs enable detection of interesting patterns that independent aspect alone cannot reveal. We propose an ensembling method to detect campaign. First, we define a dissimilarity matrix A between bots participating in a campaign, where we compute the pairwise distance between two bots as:

$$A_{i,j} = 1 - \frac{\|Community(i) \cap Community(j)\|}{\|Community(i))\|}$$

Where $Community(i)$ refers to the set of communities that user i belongs to. The resulting distance matrix A contains normalized values range between 0 to 1. Where 0 indicates that the two bots appeared in the same cluster in all graphs, and 1 means the two bots did not appear in any common cluster. Then, we use average linkage hierarchical clustering to cluster bots and choose a restrictive threshold to stop unnecessary cluster merging. The merging starts with the most similar bots and stop when threshold is *0.8*. The selected threshold is chosen because it ensemble bots in one community if all bots share one common community on average with all other bots within the campaign. For verification, we conducted a small experiment on a sample of labeled bots in the U.S. Election, where labels are either Trump or Clinton supporter, we used different threshold and reported the Normalized Mutual Information (NMI) with the labeled data. the largest NMI was reported at 0.8.

3.3 Interaction Detector

Once we find a set of campaigns, we are interested to study the interactions among them. The simplest starting point is to consider pair-wise interactions. We consider developing a machine learned classifier to automatically classify interactions in agreeing and disagreeing categories.

We label interactions between a pair of campaigns by manually checking the tweets, replies and retweets where bots from both campaigns participated. Such interactions can be largely categorized in two classes: agreeing and disagreeing. The example in the Introduction can be treated as a disagreeing interaction between the Trump and Clinton supporting bots. One may consider creating a full scale of classes between -3 and 3, 0 being the neutral class, instead of a two-class problem. However, the cost associated with labeling hundreds of pairs of campaigns is significant. In contrast, any neutral interaction can also be thought of as weak agreement, and thus, a two-class formulation is chosen. We manually labeled 80 campaign interactions where 57.5% are disagreement interaction and 42.5% are agreement.

3.3.1 **Feature Generation and Selection.** We start with a set of 94 features. The features are from four categories: time-based, sentiment-based, user-based features and network-based features. We describe a subset of features from each category below.

(1) *Time-based Features:* Temporal features help revealing bots that are collaboratively working toward the same objective or operated by the same software. Features such as the number of temporally correlated bots can be useful to understand the relation between a pair of campaign. Similarly, average interval time between mentions and number of bots involved in conversational interaction can indicate the interaction type. Usually, conversations with small intervals between mentions can be an indication for argument and disagreement.
(2) *Sentiment-based Features:* While retweet interaction almost always indicates agreement, mention interaction is controversial in nature. Bots and cyborgs could engage in arguments to support or attack a certain topic. To understand the nature of these conversations, we investigate entities sentiment within each conversation using IBM Watson Natural Language Understanding API [1]. For each conversation, we create various features describing the number of sentiment disagreement, difference of average sentiment over all entities to understand bots opinion polarity towards topics.
(3) *Content-based Features:* Usually, campaigns that share common objective tend to have more agreement than disagreement towards specific topics, and vice versa. We create features that characterize the relationship between two interacting campaigns. Examples include number of common topics, hashtags and media between two campaigns.
(4) *Network-based Features:* In addition to these three categories, we summarize bots and campaign connectivity using features that describe network topology such as ratio of friends to followers for a user.

Although indirect interactions are possible, we consider only direct interactions in the forms of retweets and mentions between campaigns. We obtain 94 features from four categories. We perform feature selection to identify the best features based on their importance weights in a decision tree model using Gini importance. After feature selection, the set is reduced to 15 features. Most informative features are content-based, temporal-based and sentiment-based features. The complete list of features is available in the supporting webpage [2].

3.3.2 **Training the Classifier.** We employ an AdaBoost model trained on decision tree classifier (CART) with ten weak learners. Information Gain is used to measure the quality of splits, then predictions from different learners are combined using weighted majority vote to produce the final prediction. Our choice for Adaboost is a result of lack of labeled data. Quantifying the sentiment of an interaction needs significant effort because of short length of tweets (The character limit for tweets is 280) and many alternative usages (emoji, abbreviation, etc.). Boosting the decision tree helps tackle these challenges.

4 EXPERIMENTAL EVALUATION

4.1 BotCamp by Numbers

We describe the BotCamp framework in numbers for the U.S. Election campaigns. First, we start with 20 seed keywords, the keyword generator component expands the set to 231 keywords in 60 days. Using the campaign detector, we collect a set of 75 million tweets from 6 million users talking about the election. The number of bots detected is 120K. We exclude clusters that are not matching in content and identify 29K bots from different micro-campaigns. We construct five graphs: retweet (7162 bots with 30811 edges), mention (1137 bots with 785 edges), hashtags (4122 bots with 731687 edges), media (954 bots with 10385 edges) and temporal (29840 with 73623). Graph clustering and ensembling are performed to obtain clusters of 29K bots and ensemble them into 231 campaigns. From the interaction classifier, we identify 87 disagreement interactions and 2700 agreement interactions between bot campaigns.

The above set of numbers are reproducible using the dataset provided in our supporting webpage [2]. However, U.S. Election 2016 has already happened, which limits comparison to alternative methods. To facilitate experimental comparison, we made our code public in the supporting webpage, it only requires a set of keywords to run for days to weeks, and produce interacting campaigns.

4.2 Evaluation of Interaction Classifier

We evaluate the boosted decision tree classifier using a 10-fold cross-validation technique. The average and standard deviation of classification performance is described in the Table 2. The results strongly suggest that the feature set can capture the manually labeled training data. The low standard deviation suggests consistency across random samples.

We consider applying the classifier to the unlabeled pairs of campaigns that have some form of interactions (i.e. retweet, mentions, etc.) between them. The results are shown in the Table 3

We have identified 87 disagreeing pairs of campaigns during U.S. election that include disagreement over debate results, email controversy, etc. The result suggests that while some campaign domains are non-competitive in nature, others are controversial leading to disagreement interactions.

4.3 Example Campaigns

This project has identified several small to large campaigns with bot participation in Twitter. Are they meaningful campaigns? - is

Table 1: Example campaigns in the U.S. Election. The bot accounts may be suspended currently.

Campaign	Bots Examples	Number of Bots
Trump Supporters	RWB4Trump, usfortrump Vegans4Trump, TXChiks4Trump MyVoteIs4Trump, Hyperslw4Trump	480
Clinton Supporters	Hillary2016MN IL4Hillary, voteforourlives Liberalibrarian, Hope012015	304
Sanders Supporters	1Birdie4Sanders BernieEvents, BernItUpTV i_AM_theChange, drJimWas4Bernie	100

Table 2: Model Performance

	Accuracy	Precision	Recall
Average	87.5%	98%	83.6%
Variance	0.025	0.003	0.049

Table 3: Interaction Summary

Number of Interactions	Interactions Agreement	Interactions Disagreement
2,790	6.88%	3.11%

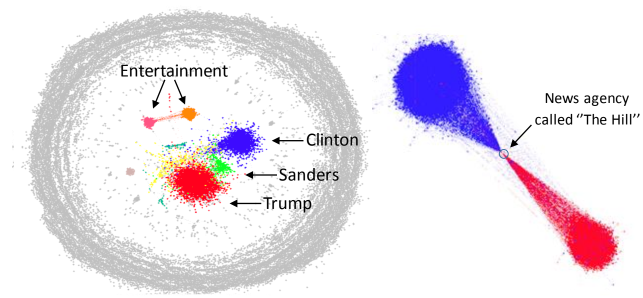

Figure 4: (left) Detected campaigns are shown on an undirected retweet graph (red for Trump supporters, blue for Clinton supporters, green for Sanders supporters). (right) Campaigns found by considering a directed retweet graph. Node in the middle is a news agency called The Hill. Colors indicate strong sentiment polarity towards different candidates based on hashtags.

the natural follow up question. We have investigated the campaigns manually to identify their objectives. Tagging all accounts in all campaigns is labor intensive. We take a 10% random sample of bot accounts to identify the objective, by navigating through their profile and rendering the last 15 tweets. In this section, we first show examples of a few campaigns (see Table 1). We name the campaigns based on the objectives we identified.

(1) **Trump Supporters**: In this campaign, all bots supported candidate Trump in U.S. presidential election in 2016. Their names, profile pictures and tweets show that they mostly care about politics. The bot accounts show strong retweet interactions among them.
(2) **Clinton Supporters**: All bots in this campaign are supporting Clinton. All their tweets are mostly political tweets. Number of Clinton supporting bots is less than that of Trump supporting bots.
(3) **Sanders Supporters**: All bots in this campaign supported candidate Sanders in the U.S. election 2016.

To provide a comprehensive picture, we show all of the campaigns detected by BotCamp in U.S. election dataset on an undirected retweet graph in the Figure 4. In addition to the supporters of three prominent candidates, several other small campaigns exist. The two loosely connected campaigns that we labeled as *Entertainment* campaigns are consisted of bots interested in variety of topics including politics, but mostly celebrity news. We explain the weak communication between the campaigns as an artifact of partially complete dataset. Note that Twitter API provides 1% of tweets.

The Figure 4(left) shows that politically motivated bots rarely retweet mutually across parties. This is not surprising, however, the directed retweet graph in the Figure 4(right) shows that the campaigns retweet from a common news source named *The Hill*.

5 CONCLUSION

Online social media is tremendously important for the future of democratic governance. Automated activities on social media create opportunities for manipulation, misinformation and distrust. This paper demonstrates that social campaigns can be corrupted by inorganic interactions among bots and develops a technique to classify inorganic interactions among and within campaigns. We show empirical evidence of various interactions among campaigns. However, this work is merely one step towards better monitored social media, significant effort must be made to protect human users from inorganic interruptions.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 1527127.

REFERENCES

[1] 2016. Natural Language Understanding. https://www.ibm.com/watson/services/natural-language-understanding

[2] 2019. Supporting web page containing data, code, results for BotCamp. http://cs.unm.edu/~nabuelrub/BotCamp.

[3] Norah Abokhodair, Daisy Yoo, and David W. McDonald. 2015. Dissecting a Social Botnet: Growth, Content and Influence in Twitter. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing - CSCW '15*. ACM Press, New York, New York, USA, 839–851. https://doi.org/10.1145/2675133.2675208

[4] Noor Abu-El-Rub, Amanda Minnich, and Abdullah Mueen. 2017. Anomalous Reviews Owing to Referral Incentive. In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*. ACM, 313–316.

[5] Noor Abu-El-Rub, Amanda Minnich, and Abdullah Mueen. 2017. Impact of referral incentives on mobile app reviews. In *International Conference on Web Engineering*. Springer, 351–359.

[6] ADORABLEDEPLORABLE. 2016. ChristiChat: RT JaredWyand: Twitter pushing ... https://twitter.com/GoldStarMomTX55/status/795287405804879872

[7] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment* 2008, 10 (2008), P10008.

[8] Zhu Sun Chang Xu Jie Zhang. 2017. Online Reputation Fraud Campaign Detection in User Ratings. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, {IJCAI-17}*. 3873–3879. https://doi.org/10.24963/ijcai.2017/541

[9] Nikan Chavoshi, Hossein Hamooni, and Abdullah Mueen. 2016. DeBot: Twitter Bot Detection via Warped Correlation.. In *ICDM*. 817–822.

[10] Nikan Chavoshi, Hossein Hamooni, and Abdullah Mueen. 2016. Identifying correlated bots in twitter. In *International Conference on Social Informatics*. Springer, 14–21.

[11] Nikan Chavoshi, Hossein Hamooni, and Abdullah Mueen. 2017. On-Demand Bot Detection and Archival System. *Proceedings of the 26th International Conference on World Wide Web Companion* (2017), 183–187. https://doi.org/10.1145/3041021.3054733

[12] 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*. International World Wide Web Conferences Steering Committee, 273–274.

[13] Son Dinh, Taher Azeb, Francis Fortin, Djedjiga Mouheb, and Mourad Debbabi. 2015. Spam campaign detection, analysis, and investigation. *Digital Investigation* 12, Supplement 1 (2015), S12 – S21. https://doi.org/10.1016/j.diin.2015.01.006

[14] Emilio Ferrara, Onur Varol, Clayton Davis, Filippo Menczer, and Alessandro Flammini. 2014. The Rise of Social Bots. *CoRR* abs/1407.5 (2014). http://arxiv.org/abs/1407.5225

[15] Emilio Ferrara, Onur Varol, Filippo Menczer, and Alessandro Flammini. 2016. Detection of Promoted Social Media Campaigns.

[16] Michelle C Forelle, Philip N. Howard, Andres Monroy-Hernandez, and Saiph Savage. 2015. Political Bots and the Manipulation of Public Opinion in Venezuela. *SSRN Electronic Journal* (2015). https://doi.org/10.2139/ssrn.2635800

[17] Kiran Garimella, Ingmar Weber, and Munmun De Choudhury. 2016. Quote RTs on Twitter: usage of the new feature for political discourse. In *Proceedings of the 8th ACM Conference on Web Science*. ACM, 200–204.

[18] Kyumin Lee, James Caverlee, Zhiyuan Cheng, and Daniel Z. Sui. 2011. Content-driven detection of campaigns in social media. In *Proceedings of the 20th ACM international conference on Information and knowledge management - CIKM '11*. ACM Press, New York, New York, USA, 551. https://doi.org/10.1145/2063576.2063658

[19] Kyumin Lee, James Caverlee, Zhiyuan Cheng, and Daniel Z. Sui. 2013. Campaign extraction from social media. *ACM Transactions on Intelligent Systems and Technology* 5, 1 (12 2013), 1–28. https://doi.org/10.1145/2542182.2542191

[20] Huayi Li, A Mukherjee, Bing Liu, R Kornfield, and S Emery. 2014. Detecting Campaign Promoters on Twitter Using Markov Random Fields. In *Data Mining (ICDM), 2014 IEEE International Conference on*. 290–299.

[21] Richard J. Oentaryo, Arinto Murdopo, Philips K. Prasetyo, and Ee-Peng Lim. 2016. On Profiling Bots in Social Media. Springer, Cham, 92–109. https://doi.org/10.1007/978-3-319-47880-7{_}6

[22] Gordon Pennycook, Tyrone D Cannon, and David G Rand. 2017. Prior exposure increases perceived accuracy of fake news. (2017).

[23] Eduardo J Ruiz, Vagelis Hristidis, Carlos Castillo, and Aristides Gionis. 2013. Measuring and Summarizing Movement in Microblog Postings.. In *ICWSM*.

[24] Pablo Suárez-Serrato, Margaret E Roberts, Clayton Davis, and Filippo Menczer. 2016. On the influence of social bots in online protests. In *International Conference on Social Informatics*. Springer, 269–278.

[25] V. S. Subrahmanian, Amos Azaria, Skylar Durst, Vadim Kagan, Aram Galstyan, Kristina Lerman, Linhong Zhu, Emilio Ferrara, Alessandro Flammini, Filippo Menczer, Rand Waltzman, Andrew Stevens, Alexander Dekhtyar, Shuyang Gao, Tad Hogg, Farshad Kooti, Yan Liu, Onur Varol, Prashant Shiralkar, Vinod Vydiswaran, Qiaozhu Mei, and Tim Huang. 2016. The DARPA Twitter Bot Challenge. (Jan. 2016). arXiv:1601.05140 http://arxiv.org/abs/1601.05140

[26] A Nicole Sump-Crethar. 2012. Making the most of Twitter. *The reference librarian* 53, 4 (2012), 349–354.

[27] Kurt Thomas, Vern Paxson, Damon Mccoy, and Chris Grier. 2013. Trafficking Fraudulent Accounts : The Role of the Underground Market in Twitter Spam and Abuse Trafficking Fraudulent Accounts :. In *USENIX Security Symposium (SEC'13)*. 195–210.