

Election-bots: A Propaganda Machine

By Paulino Diaz Mejia

Executive Summary

While extensive efforts have been made to identify automated Twitter accounts (A.K.A bots) in the 2016 and 2020 U.S. elections, understanding the influence these accounts have on their audiences and the population remains a challenge. This analysis expands on previous efforts by examining language patterns and language coordination between bots and real users to determine the influence and effectiveness of bot content. Using modern machine learning and computational content analysis methods this research achieves three objectives:

1. The identification of bots in Twitter posts surrounding the 2020 presidential election
2. An analysis of the content produced and amplified by these bots which discovers that automated Twitter accounts predominantly post content favorable to presidential candidate Donald Trump.
3. The creation of a measure of discursive influence that highlights the most impactful tweets for both bots and users within a 3-hour time window.

The results of the analysis suggest bot content is highly influential, but additional research is needed to understand the mechanisms by which it exerts such influence. Twitter conversations take place within a larger context that involves other media platforms and the events of the physical world. To generate stronger claims, future analysis should aim to disentangle these effects.

Background

As our physical world struggles to deal with the impacts of a public health pandemic, our digital world is being assaulted by an equally aggressive disinformation epidemic. Social media platforms are being routinely weaponized by ill-intentioned actors to push false or misleading narratives that create confusion, sow distrust and can lead to a deterioration of the social fabric. This phenomenon is particularly prevalent in online political discourse. An analysis of 14 million messages shared on Twitter during the 2016 election found that 31% of all low-credibility information on the network came from just 6% of the accounts ([Shao, C., Ciampaglia, G.L., Varol, O. et al., 2019](#)). These accounts were all bots. A similar study by the University of Southern California estimated between 9% and 15% of Twitter's monthly active users are bots ([Varol, O. et. al., 2017](#)).

What is a bot?

The simplest definition of a Twitter bot is that of an account that posts content automatically. These accounts can be used for benign tasks like automatically posting or retweeting information such as news and academic papers. These bots are easy to identify, as they only share one type of content and don't try to misrepresent themselves and their motivations ([Lokot, T. et al., 2016](#)). Other more problematic bots can adopt sophisticated strategies to imitate humans that make their detection harder. These bots mine the web for publicly available photos, usernames and profile information which they use to impersonate real users. These bots can also leverage advanced machine learning technologies and algorithms to automatically generate content that emulates the patterns of online activity of humans ([Hwang et al., 2012](#)).

The Social Media Observatory at Indiana University has created a tool called [Bot Sentinel](#) that tracks content tweeted by bots, highlighting the top trends by day. Going as far back as January 2019, hashtags like #MAGA (Make America Great Again) and words like ‘Trump’ and ‘@realDonaldTrump’ have been part of the top ten topics and phrases tweeted by bot accounts ([Bot Sentinel](#), 2020). Given the continued prevalence of bots in the political conversation, it is important to understand the intentions behind these accounts and uncover any efforts to alter the organic nature of our online political discussions.

In the following sections, I introduce the problems of bot identification and influence measurement, and present the dataset and methods used in the analysis.

Sampling

To build a reliable sample that includes a significant amount of bot accounts I made use of the [rtweet](#) package in R. This package provides a straightforward way to gather data from Twitter’s API, including a `search_tweets` function that allows users to query up to 18,000 tweets per request going back 7 to 9 days, or less. This function also lets users filter content by keywords, language, date and location, among other characteristics. By querying Twitter’s API, I was able to collect a sample of 335,299 tweets spanning 7 days, starting on May 2, 2020 and ending on May 8, 2020.

To optimize the data collection process and ensure bots were significantly represented, I created a search query that targets bot infested conversations using the top words, phrases and hashtags used by automated accounts according to [Bot Sentinel](#) data. The exact query can be found below:

"\"Fake News\" OR #FakeNews OR #Trump2020 OR Trump OR President OR 'President Trump' OR @realDonaldTrump OR @JoeBiden OR Biden OR 'Joe Biden' OR #WWGIWGAN OR #MAGA lang:en"

Bot Identification

After gathering a representative collection of tweets, I proceeded to classify each account in the sample based on its probability of being a bot. The most common approaches to bot classification are based on supervised machine learning algorithms. One of the best known ones is the OSoMe [Botometer](#) which I will be employing to classify my sample of tweets.

Given a Twitter account, Botometer extracts over 1,000 features relative to the account from data provided by the Twitter API. The algorithm then uses these features to produce a Complete Automation Probability (CAP) score: the higher the score, the greater the likelihood that the account is a bot. To estimate the CAP, the model uses a random forest classifier to compute the fraction of trees that classify an account as a bot. Then it estimates the posterior probability that an account is a bot using Bayes’ rule:

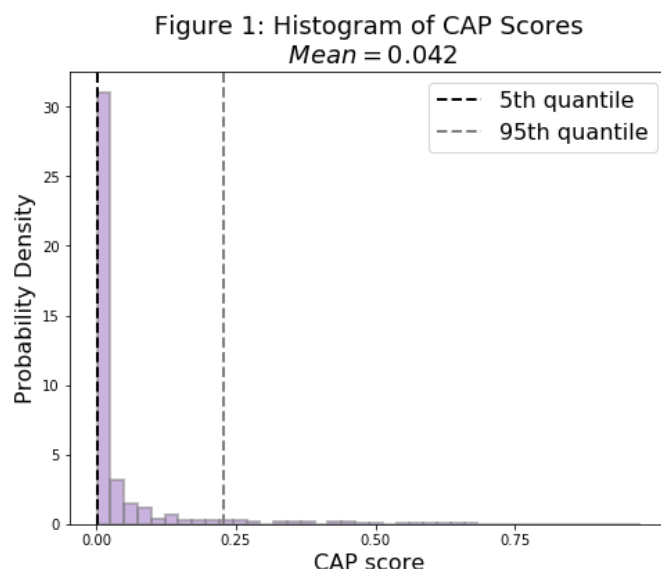
$$P(\text{Bot} \mid S) = P(\text{Bot}) \frac{P(S \mid \text{Bot})}{P(S)}$$

Where $P(\text{Bot} \mid S)$ is the posterior probability, $P(\text{Bot})$ is the background probability that a given account is a bot (in this case 0.15 based on [previous estimates](#) of the proportion of bots on Twitter), and $\frac{P(S \mid \text{Bot})}{P(S)}$ compares the likelihood that a bot has score $P(S \mid \text{Bot})$, with the probability that any account has that score. The posterior probability $P(\text{Bot} \mid S)$ is called the CAP score,

which is usually a conservative estimate (reflecting the relative rarity of bots) of the probability that a given account is a bot.

To train the system, the creators of Botometer initially used a publicly available dataset consisting of 15K manually verified Twitter bots identified via a honeypot approach ([Lee, Eoff, and Caverlee, 2011](#)) and 16K verified human accounts. This approach is not without issues, as there are a large number of accounts that lie in the grey area between human and bot behavior, where even experienced researchers cannot easily discriminate. Nonetheless, researchers were able to obtain 86% overall classification accuracy by carefully labelling each account, and enriching the training set with accounts belonging to highly sophisticated bots ([Varol, O. et. al., 2017](#)).

Figure 1 shows the resulting distribution of CAP probabilities after running all the accounts in my sample through the Botometer classifier. Most of the sample consists of accounts with very low CAP probabilities, with only 5 percent of accounts being assigned a probability of 0.23 or higher. This is expected, given the Botometer algorithm is designed to minimize false positives.



To label these accounts as bots or real users I used a CAP threshold of 0.43, which the creators of the algorithm found minimizes both type 1 and 2 errors ([Varol, O. et. al., 2017](#)). The resulting breakdown of classifications is presented below in the results section.

Content Analysis

To understand the type of content and narratives that bots are spreading in relation to the election I use the text data in each tweet and examined the word distributions, collocations and topics.

I begin by tokenizing and normalizing the text in each tweet using the [spaCy](#) package for natural language processing in Python. Tokenization is the process by which a sentence, paragraph or text is reduced to single word tokens, removing any punctuation or white spaces. These tokens are then normalized by removing stop words (using spaCy's English stopwords dictionary), transforming all letters to lower case, and lemmatizing words to their base form (e.g. the lemma of the word 'voting' is 'vote'). All these tasks are performed using spaCy's *en_core_web_sm* libraries.

After cleaning the data, I perform a comparison of word frequency counts and word collocations between content posted by bots and content posted by users. To minimize the possibility of type 1 and 2 errors in classification, this portion of the analysis only compares the top of the CAP distribution (those above a .75 probability of being bots) against the bottom 1% (users). These amounts to 870 accounts being labeled as bots, and 849 accounts being labeled as users.

An individual word frequency count is likely to be ineffective at uncovering differences between user and bot content. Recall that the sample of tweets was selected using a specific set of

keywords. This implies that both bot and user content is bound to have similar individual word frequency counts at the top of the distribution. To better assess the similarity or difference between bot and user content we will use the Kullback-Liebler (KL) Divergence, which describes the divergence between two probability distributions ([Kurt, W., 2017](#)). This measure is a non-commutative ‘distance’ between distributions computed using the likelihood ratio:

$$D_{KL}(p||q) = \sum_{i=1}^N p(x_i) \log\left(\frac{p(x_i)}{q(x_i)}\right)$$

In this context, for any sample x the ratio between the likelihoods indicates how much more likely the data-point is to occur in $p(x)$ as opposed to $q(x)$. This is not a ‘distance’ in the strict sense because the KL divergence from $p(x)$ to $q(x)$ is generally different from the KL divergence from $q(x)$ to $p(x)$. This ‘distance’ might be best understood as the surprise of finding data-point x in distribution $q(x)$ given the true distribution is $p(x)$. To find an actual measure of the distance between the distributions I also computed the Jensen-Shannon divergence, which is a symmetrized and smoothed version of the KL divergence ([Fuglede, et. al, 2003](#)). Both of these measures can be calculated using the [scipy](#) library in Python.

Since these are relative measures, I also estimate a baseline KL divergence and Jensen-Shannon divergence, comparing two random subsamples of our dataset, as opposed to user vs. bot samples.

Next, I proceeded to identify the most interesting collocations in each group. Collocations are expressions consisting of two or more words with a combined meaning that is dependent on the order of the words ([Manning and Schütze, 1999](#)). Word collocations are likely to be far more revealing than individual word counts due to the way in which we sampled the data. As we show in the results section, two-word collocations (bigrams) and three-word collocations (trigrams) highlight key differences in use of language between bots and real users.

Determining what constitutes a relevant collocation is not trivial, as two frequently occurring words are expected to co-occur a lot just by chance – even if they don’t share a combined meaning. To account for this, each co-occurrence is scored using its likelihood ratio. The likelihood ratio is a hypothesis test that tells us how much more likely one hypothesis is than the other. In the context of a bigram (word¹word²), hypothesis 1 is that the occurrence of word² is independent of the previous occurrence of word¹, and hypothesis 2 is that the occurrence is dependent – which is good evidence for an interesting collocation. These scores are estimated using the *collocations* Module in the [nltk](#) library for Python.

Topic Modelling

To continue my analysis of differences in language patterns between bots and real users, I employ an unsupervised learning approach for topic modelling called Latent Dirichlet Allocation (LDA) ([David Blei, 2012](#)). This model is part of the probabilistic topic modelling suite of algorithms which aims to annotate large archives of documents with thematic information. Topic modeling algorithms do not require any prior annotations or labeling of the documents, the topics emerge from the analysis of the original texts. This algorithm defines documents as probability distributions over topics, and topics as probability distributions over words. Through a generative and iterative process, it finds the maximum likelihood that each document is composed of a

given set of topics. The fact that topics can be shared among documents and words among topics, lets me create topic distributions for each tweet.

To estimate the LDA model I use the *LdaModel()* function that is part of the [gensim](#) library for Python. This algorithm receives three main hyperparameters as inputs: alpha, beta and the number of topics. The alpha parameter controls how many topics load into each document. Higher values imply that the documents are expected to have many topics. The beta parameter controls how many words are loading into each topic. High values imply that topics are expected to have many words. Finally, the model assumes that the number of topics is known and fixed.

Since the goal of the analysis is to find coherent topics that explain language differences between bot and user generated content, relatively low levels of alpha (.001) and beta (.001) are used. This ensures the model finds distinct words that characterize each topic, and distinct topics that characterize each document, helping the model achieve good interpretability. To choose the appropriate number of topics, I trained the model using a range of topic numbers going from 5 to 20. I evaluate the performance of each model using *gensim*'s *CoherenceModel* function, which estimates the topic coherence as the pointwise mutual information (PMI) of each word pair, estimated over the entire corpus. Of all the topic scoring methods, PMI (term co-occurrence via simple pointwise mutual information) is a good consistent performer for evaluating the quality of a given topic, in terms of its coherence to a human ([Newman](#) et al., 2010).

Before training the LDA model, the normalized tokens were filtered based on their term frequency – inverse document frequency (tf-idf). The tf-idf is a weighting factor that is intended to reflect how important a word is to a document in a collection or corpus. Its value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word. This weighting scheme allows us to remove words that are not very good at distinguishing documents (either because they appear too many times or rarely). I chose to filter words that appear in more than half the documents, in less than 100 of them or in more than 1500.

Finally, I turned our filtered, normalized tokens into a term document matrix in which the tokens are the rows and the documents are the columns, and each value represents the number of times a word appears in each document. This stream of document vectors is the corpus, which can then be fed into the *LdaModel()* function to estimate the topic distributions.

Figure 2 shows the results of the LDA models in terms of their coherence scores. The best performing model is the one with 14 topics. I labeled each of the 14 topics by inspecting the tweets that have the highest loadings for each of the topics, and the list of most representative words for each topic.

Measuring Influence

Finally, in order to quantify the influence that bot accounts have on the organic Twitter conversation, I analyze the evolution of language patterns across time, within our sample. The intuition behind this approach relies on the premise that influence can be measured by the extent in which one agent’s language patterns are used and copied by others. The methodology works in the following way:

1. First, the patterns of language are identified by a topic model, in this case, our previously trained LDA topic model.
2. Second, Kullback–Leibler Divergence (KL divergence) is used to measure how these patterns are propagated from tweet to tweet. KL divergence measures the extent to which a model is ‘surprised’ by new data. In this context, it measures how the expectations of a pattern are violated by latter patterns. This measure of surprise can be used to quantify the deviation of a tweet from the patterns of previous ones (novelty), and from patterns that appear in the future (transience). High surprise compared to the past indicates a tweet is providing new information, while high surprise compared to the future indicates this information was not adopted by future patterns. These two measures can then be combined to estimate the resonance of content, which is a characteristic of content that both differs from the past but leaves traces in the future.

This method for measuring influence was first used by Simon DeDeo to analyze speech patterns and their cultural evolution within transcripts of parliamentary speeches in the French National Constituent Assembly ([DeDeo et al., 2017](#)). Under this framework the novelty (N) of a given tweet (s) is its mean surprise (KL divergence) given other tweets within a window (w) beforehand:

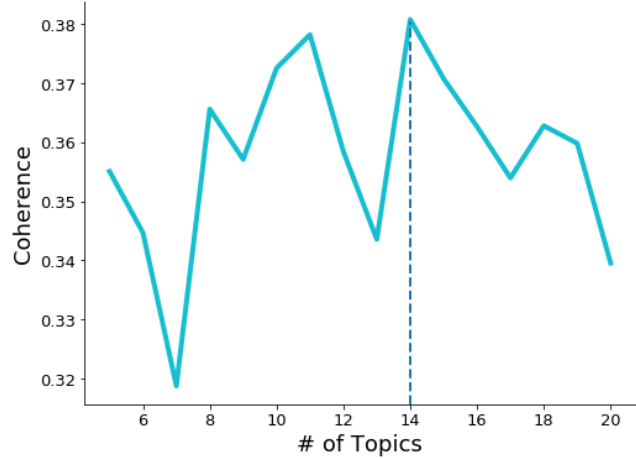
$$\mathcal{N}_w(j) = \frac{1}{w} \sum_{d=1}^w \text{KLD} \left(s^{(j)} | s^{(j-d)} \right),$$

Transience is the same as the novelty, under time reversal, and the resonance is then the difference between the novelty and the transience:

$$\mathcal{R}_w(j) = \mathcal{N}_w(j) - \mathcal{T}_w(j)$$

I estimated measures of resonance for a window of tweets $w = 1000$. This is equivalent to approximately three hours of twitter activity in our sample. This decision was made for convenience, given larger windows require a significant amount of time to compute. The corresponding influence measures for the top 1% of the CAP distribution (bots) and the bottom 1% (users) can be found below under the results section.

Figure 2: Coherence Scores for Different # of Topics



For this last portion of the analysis, I only focused on original content (no retweets). My primary goal is to understand if bot generated narratives change the way in which other accounts communicate their ideas. Performing this analysis on retweets would likely upward bias our measures of resonance for bots. As I will show below, a majority of bot content comes in the form of retweets. In the case that bots coordinate their behavior and purposefully retweet each other's content, analyzing their content through this approach could artificially inflate our estimates by producing very low measures of transience for bot content. However, this wouldn't necessarily be an indication of other users adopting bot's language patterns.

Results

The output of the Botometer classifier provides a partition of the data that allows us to identify key differences and similarities between bot and user accounts. As seen in **Table 1**, only 2.1% of the accounts in our sample are bots. However, these accounts tend to be significantly more active than real users, accounting for almost 4.2% of all content. This is also reflected in the average posts per type of account. While users post an average of 3.97 posts within a week (the window of time for which we sampled Twitter), bot accounts reach an average of almost double that amount (6.78 posts per week).

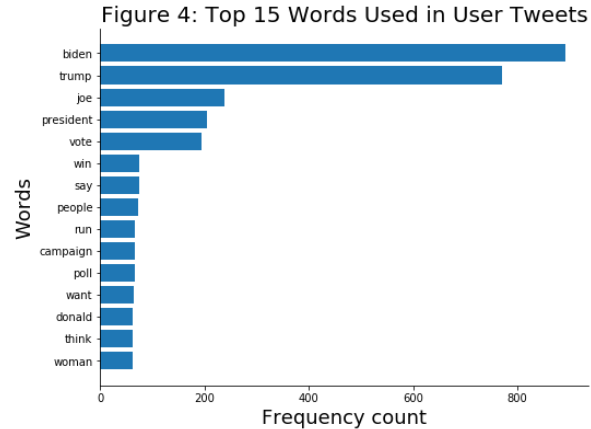
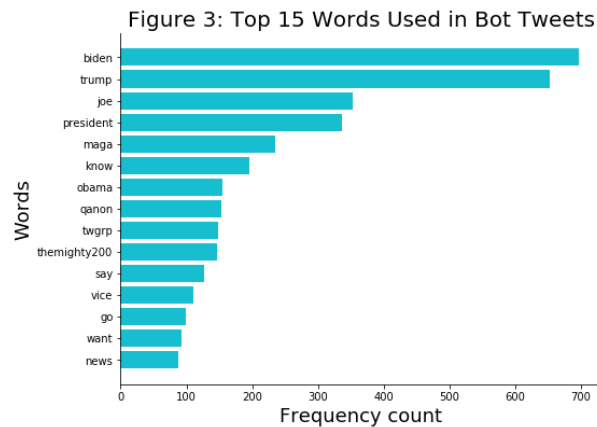
Another important characteristic of bot accounts is their tendency to post retweets as opposed to original posts. Out of the 51,764 original tweets in the sample only 550 (1.1%) belong to bots – 98.9% of the original content is produced by users. In contrast, bots were responsible for 13,480 retweets in the sample, amounting to 4.8% of all retweets.

Given the design of the Botometer classifier, which places a higher weight on minimizing false positives than false negatives, these are likely to be conservative estimates. Other studies of the prevalence of bot accounts on Twitter have yielded higher approximations, ranging from 6% ([CNetS](#), 2019) to 16% ([Zhang & Paxson](#), 2011).

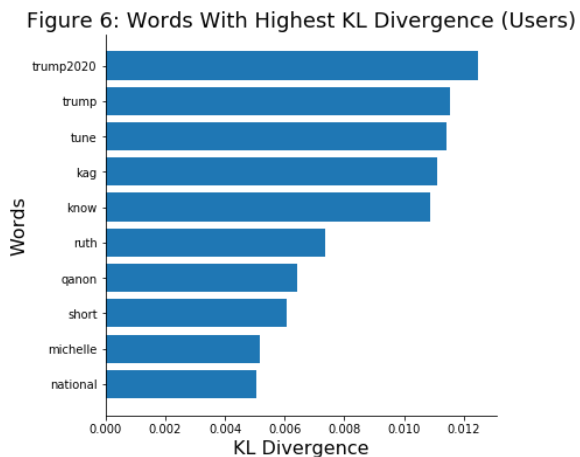
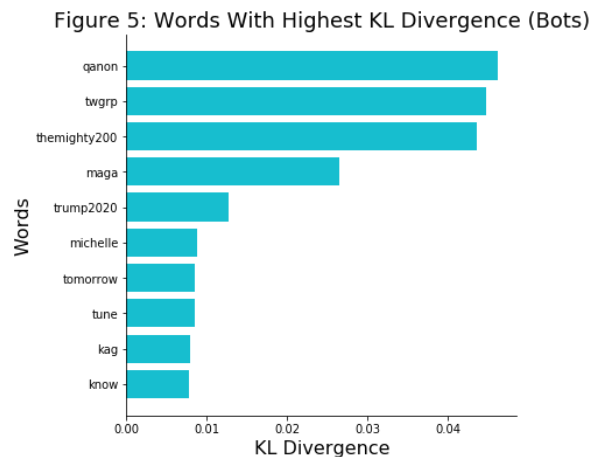
I now turn my attention to the actual content of the tweets, starting with the most frequent words used by each group. As expected, the top words across groups, shown in **Figure 3** and **4**, are very similar both in terms and frequency. After all, the sample was created by filtering content using keywords like 'Trump' and 'Biden.' As one moves down the list of frequency counts, however, the terms begin to change, pointing to actual differences in the way each group talks about the candidates.

Table 1: Bot vs. User Data

Account Type	user	bot
# of Accounts	80561	1737
# of Accounts (Percent)	97.9%	2.1%
# of Posts	321269	14030
# of Posts (Percent)	95.8%	4.2%
# of Tweets	51214	550
# of Tweets (Percent)	98.9%	1.1%
# of Retweets	270055	13480
# of Retweets (Percent)	95.2%	4.8%
Average Posts (Per Account)	3.97	6.78
Average Tweets (Per Account)	0.63	0.27
Average Retweets (Per Account)	3.34	6.52

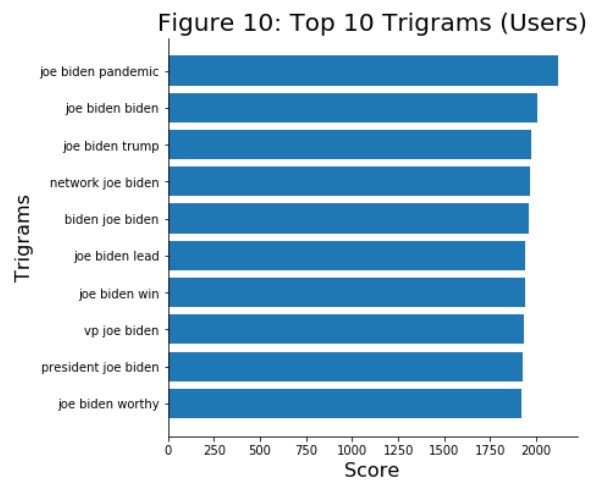
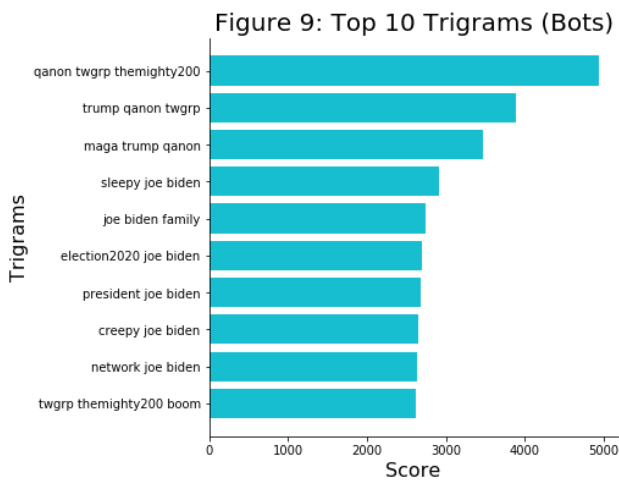
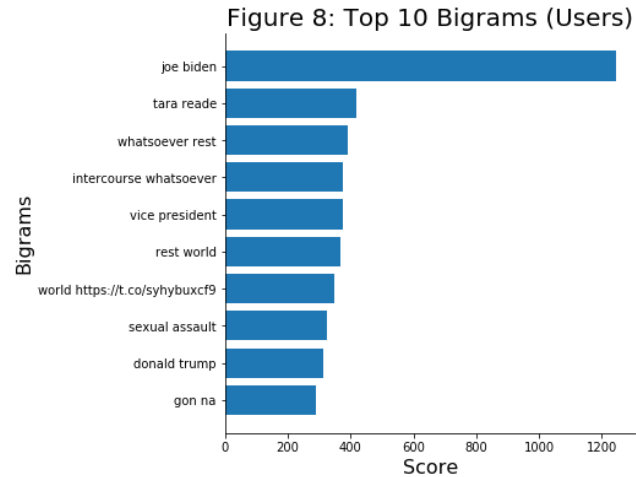
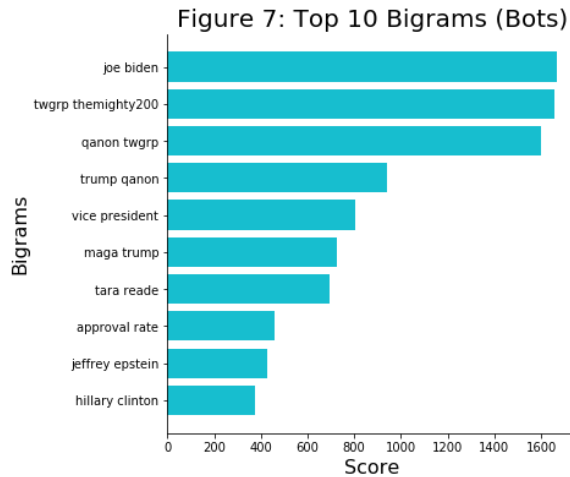


To better understand the differences between these word distributions I measure their divergence. As a point of reference, two random subsamples of 1% percent of the data have a Jensen-Shannon divergence of 0.15. In contrast, the bot and user samples have a Jensen-Shannon divergence of 0.317 – almost double that of a comparison of random samples. **Figures 5 and 6** show the top scoring words in terms of their KL divergence for each group.



Hashtags that are usually related to favorable tweets about Trump make it to the top of the list based on their divergence. Let's unpack what this means. As I explained in the methodology section, KL divergence can be best understood as a measure of surprise. In this sense, it is much more surprising to see hashtags like 'qanon' (a pro-trump [conspiracy](#) theory), 'twgrp' ([Trump World Groups](#)), 'themighty200' (pro-Trump [Twitter group](#)) in the bot distribution, than it is surprising to *not* see them in the user distribution. The values along the x-axis for the bot divergence measures are also an order of magnitude larger than those of users. In other words, these hashtags account for the biggest differences in the word distributions of bot and user content.

To place the top words in each distribution in context, I also look at the top bigrams and trigrams for bots and users. **Figures 7 through 10**, highlight these results.



In **Figure 7**, bigrams with the highest likelihood ratio for bots are once again hashtags that accompany tweets in support of Trump ('qanon', 'twgrp', etc.) or which have positive connotations about the president ('maga'). On the other hand, the user generated content (**Figure 8**) seems to be focused on the sexual allegations against Joe Biden, which were top of mind at the time the sample was collected. This suggests that bot content might be disconnected from the organic, user generated content (which at the time was focused on one of the biggest news stories for the Joe Biden campaign). Additionally, bot content seems to be pushing narratives that are more favorable to Trump, but it is hard to tell by simply looking at word pairs.

Figure 9 and **10** provide an even starker contrast. The top scoring trigrams in Figure 9 are either hashtags used in support of Trump, or negative attributes and insults towards Joe Biden (such as 'Sleepy Joe' and 'Creepy Joe'). On the other hand, high-scoring Trigrams in **Figure 10** are predominantly associated with positive words about Joe Biden. The fact that our model assigns a high score to pairs or trios of words which are hashtags is in itself interesting. In a normal context, one would think that the order of hashtags doesn't matter and would expect these to appear in random order. However, if the content is being generated in an automatic fashion or

being repurposed over and over (as retweets) then the log likelihood ratio for the cooccurrence of these words in this exact order is likely to be high.

To better elucidate to what extent these conversations differ from one another, let's look at the most prevalent topics. **Table 2** includes the list of words that the 14 topic LDA model discovered to be the most representative of each topic:

Topic 0	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9	Topic 10	Topic 11	Topic 12	Topic 13
reade	flynn	poll	know	like	lie	china	win	joe	maga	woman	obama	president	vote
tara	campaign	lead	joe	supporter	oval	key	beat	old	trump2020	rape	anti	joe	party
sexual	lawyer	joe	president	look	news	pandemic	need	public	kag	accuse	entire	run	bernie
assault	russia	drop	obama	people	donor	coronavirus	want	fuck	president	believe	january	say	people
allegation	investigation	point	donald	rapist	like	death	lose	talk	qanon	accuser	white	vice	democrat
claim	justice	election	people	democrat	fake	country	think	say	wwg1wga	assault	office	obama	want
joe	hoax	test	american	thing	barr	america	debate	force	god	ad	house	endorse	win
story	record	new	care	hold	bad	response	pick	let	america	question	presidency	office	november
accusation	attack	race	corrupt	support	know	americans	amash	year	obama	allegation	administration	step	think
false	pay	state	america	try	want	campaign	vp	video	economy	metoo	die	year	voter

Based on these words, and the top tweets with the highest loadings for each topic, I created the following list of topic labels:

- Topic 0: Tara Reade Allegations
- Topic 1: Candidate Records and Investigations
- Topic 2: The Horse Race and Polling
- Topic 3: Lies, Corruption and Trust
- Topic 4: The Best Option (Lesser of Two Evils)
- Topic 5: False Accusations and Fake News
- Topic 6: The Appropriate Response to COVID-19
- Topic 7: Electability and Ideology
- Topic 8: Candidate Statements on Sexual Allegations
- **Topic 9: MAGA “Patriotism”**
- Topic 10: Believing the Accusers
- Topic 11: Free-for-all (Biden vs. Bernie vs. Trump vs. Obama)
- Topic 12: Endorsements and Gaffes
- Topic 13: Voting Strategy

I then proceeded to assign each tweet to the topic with the highest loading. Figure 11 shows the percentage of posts that belong to each of the topics, broken down by the type of account that posted the tweet.

Figure 11: Proportion of Posts By Topic


Topics	Bot	User
0	5.5	8.0
1	5.5	5.0
2	5.0	8.0
3	8.5	10.0
4	3.0	5.5
5	6.0	6.0
6	7.5	9.0
7	4.0	6.0
8	4.5	5.5
9	25.0	7.5
10	3.5	4.5
11	5.5	5.5
12	9.5	9.5
13	3.5	6.5

[illegible]

This should come as no surprise given the previously uncovered differences in both word distributions and collocations. Nonetheless, this result strongly suggests that bot content in our sample is primarily focused on artificially inflating positive narratives of presidential candidate Donald Trump. Now a key question remains: Is this content influential?

A histogram showing the probability density of Resonance values. The x-axis is labeled 'Resonance' and ranges from -1.5 to 0.75. The y-axis is labeled 'Probability Density' and ranges from 0.000 to 0.020. The histogram bars are cyan. A dashed black vertical line is drawn at Resonance = 0.0, labeled 'mean' in the legend.

Resonance Bin Center	Probability Density
-1.4	0.0010
-1.2	0.0000
-1.0	0.0010
-0.8	0.0000
-0.6	0.0045
-0.5	0.0010
-0.4	0.0065
-0.3	0.0010
-0.2	0.0020
-0.1	0.0055
0.0	0.0080
0.1	0.0075
0.2	0.0090
0.3	0.0065
0.4	0.0020
0.5	0.0045
0.6	0.0020
0.7	0.0010
0.8	0.0045
0.9	0.0055



A histogram showing the probability density of Resonance values. The x-axis is labeled 'Resonance' and ranges from -1.0 to 1.0. The y-axis is labeled 'Probability Density' and ranges from 0.0 to 1.0. The histogram bars are blue. A vertical dashed black line is drawn at Resonance = 0.0, labeled 'mean' in the legend.

Figure 12 and **13** show histograms of resonance for both bot and user content. User content appears to be slightly more influential, having both a positive mean and a higher maximum value of resonance (approximately twice as large as that of bot content). Nonetheless, some of the bot content seems to be resonating. We can further decompose these measures to see how they map to specific topics.

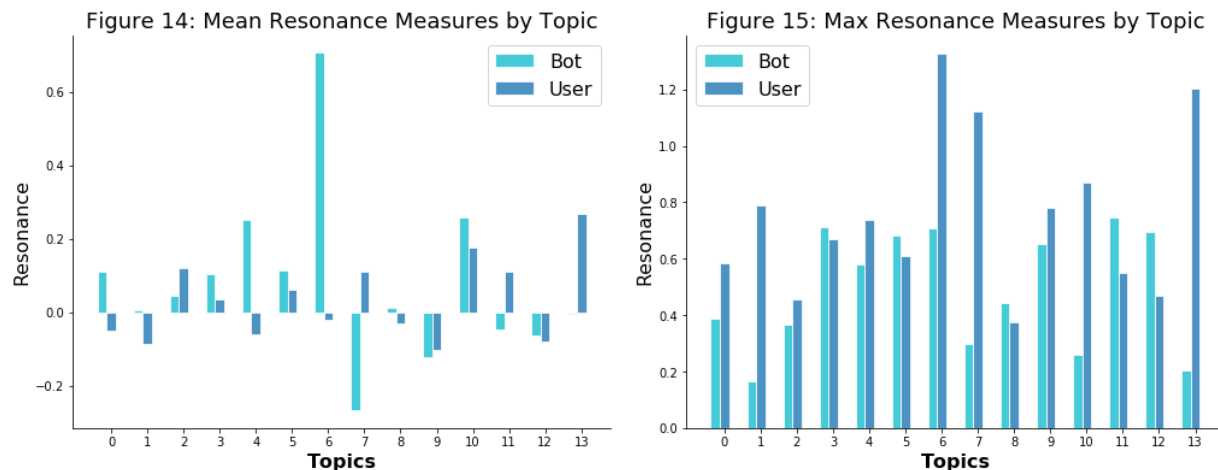


Figure 14 and **15** measure the mean and maximum influence, respectively, of bot content compared to user content, using resonance as the unit of measurement. Bots are on average more influential than users, having a higher average resonance on 9 out of the 14 topics. On the other hand, users are responsible for the most resonant content, owning the most influential pieces of content in 9 out of the 14 topics. Surprisingly, topic #9 has negative average resonance for bots, even though a quarter of bot content is focused on this specific topic (**Figure 11**). By far the most influential topic for bots is Topic 6, which is primarily focused on the response to COVID-19. The fact that such a sensitive topic is being influenced by automated accounts is worrisome and deserves further inspection. However, this is outside of the scope of this project.

Overall, the results of this analysis show that while bots constitute a small portion of the overall accounts of Twitter, their high activity levels allow them to have a disproportionate impact on the overall conversation. Additionally, bot content differs from user content in significant ways, the most obvious one being in its systematic production of positive content in support of candidate Donald Trump. This can bias the perception of the individuals exposed to this content, suggesting that there exists an organic, grassroots support for this candidate, while in reality it is all artificially generated. Fortunately, this topic doesn't appear to be amongst the most influential compared to other narratives.

This is not to say that bot content is not influential. On the contrary, on average, content posted by automated accounts outperforms content posted by users. The reasons behind this are unclear, but a potential explanation could be the use of coordinated behavior to amplify false narratives and intentionally disrupt the organic conversation.

Conclusion

The results of this analysis suggest that automated accounts are not only prevalent within Twitter, their content also appears to be highly influential. Further analysis is needed in order to understand the mechanisms by which these accounts achieve such influence. Another important aspect to consider, and which hasn't been explored in this analysis, is the unexplained third variable. Twitter conversations take place within a larger context that involves other media platforms and the events of the physical world. All of these different environments undoubtedly exert some degree of influence in the narratives seen in the platform and our analysis doesn't distinguish these effects. Nonetheless, these findings suggest that automated accounts do have an impact, and their actions and motives should be taken seriously.

As disinformation and propaganda campaigns become more and more sophisticated, political strategists and policymakers need better ways to identify both the presence and potential impact of such campaigns. This project aims to bridge the gap between state-of-the-art content analysis methods and those working to protect society from disinformation.

Bibliography:

- Barron, Alexander TJ, Jenny Huang, Rebecca L. Spang, and Simon DeDeo. 2018. "Individuals, institutions, and innovation in the debates of the French Revolution." (Links to an external site.) *Proceedings of the National Academy of Sciences* 115(18): 4607-4612.
- Bent Fuglede and Flemming Topsøe. Jensen-Shannon Divergence and Hilbert space embedding. *University of Copenhagen, Department of Mathematics* (2003)
- Blei, David. Probabilistic Topic Models. *Communications of the ACM*, Vol. 55 No. 4, Pages 77-84 2012
- Botometer: <https://botometer.iuni.iu.edu/#!/faq>
- Bot Sentinel: <https://botsentinel.com/>
- Gensim. Topic Modelling for Humans. (2009)
- Hwang, T., Pearce, I., Nanis, M. Socialbots: Voices from the fronts. *The Social Media Mediator Forum* (2012)
- Kearney Michael W., Heiss, A., Briatte F. Package 'rtweet'. Version 0.7.0. January 2020
- Kurt Will. Kullback-Leibler Divergence Explained. *COUNT BAYESIE* (2017)
- Kyumin Lee, Brian David Eoff, James Caverlee. Seven Months with the Devils: A Long-Term Study of Content Polluters on Twitter. *Fifth International AAAI Conference on Weblogs and Social Media* (2011)
- Lokot, T., Diakopoulos, N. News Bots: Automating news and information dissemination on Twitter *Northwestern University* (2016)
- Manning and Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press: selections from Chapter 5 ("Collocations"): 151-163, 172-176, 183-186.
- Natural Language Toolkit. NLTK Project. © Copyright 2020
- Newman, D., Han, J., Grieser, L. K., and Baldwin T. Automatic Evaluation of Topic Coherence. *Dept of Computer Science, University of California, Irvine* (2010)

- Onur Varol, et al. Online Human-Bot Interactions: Detection, Estimation, and Characterization. *Center for Complex Networks and Systems Research, Indiana University*, Bloomington, US (2017)
- SciPy, Scientific computing tools for Python. (2020)
- Shao, C., Ciampaglia, G.L., Varol, O. et al. The spread of low-credibility content by social bots. *Nat Commun* 9, 4787 (2018).
- Spacy. Explosion AI (2020)