University of Cincinnati

Date: 1/31/2020

I, Joshua L Shell, hereby submit this original work as part of the requirements for the degree of Master of Science in Information Technology-Online.

It is entitled:

Bots and Political Discourse: System Requirements and Proposed Methods of Bot Detection and Political Affiliation via Browser Plugin

Student's name: **Joshua L Shell**

This work and its defense approved by:

Committee chair: Shane Halse, Ph.D.

Committee member: Jess Kropczynski, Ph.D.

UNIVERSITY OF Cincinnati

36524

Bots and Political Discourse: System Requirements and Proposed Methods of Bot Detection and Political Affiliation via Browser Plugin

A thesis submitted to the

Graduate School at the

University of Cincinnati

In partial fulfillment of the

requirements for the degree of

Master of Science

in the school of Information Technology

of the College of Education, Criminal Justice, and

Human Services

by

Joshua Shell

B.A. Wilmington College

May 2012

Committee: Dr. Shane Halse (Advisor)
Dr. Jess Kropzynski

Abstract

In 2017, Twitter acknowledged the presence of bots – automated or fake accounts, controlled by either foreign governments or U.S. citizens posing as fake online personas. These accounts targeted and interacted with users using certain politically inclined keywords and posting massive amounts of false and misleading information. Consequently, bots posing as Americans were loud voices that led to a divisive social and political climate. Simultaneously, distrust in mainstream news sources was plummeting causing more people to use social media as their main source of information.

While tools exist that can determine if a given Twitter account is an authentic user or bot, they are not the most accessible products. Many require searching for an individual screenname on a separate web page, or advanced programming skills to analyze lists of users. This study examined this gap and determined the system and information requirements to develop a browser plugin that can detect bots, and the political leaning of a user's social media feed. By examining both open-source projects and public API's, this work was able to narrow down the requirements while providing the guidelines to build such a plugin.

Keywords: Bots, Fake News, Politics, Political Affiliation, Social Media, Trolls, Twitter

This Page Intentionally Left Blank

Acknowledgements

My advisor, Dr. Shane Halse, thank you for everything you have done for me in the last few months. Your guidance, humor, and insights really did inspire this work. Dr. Jess Kropczynski, thank you for never telling me “no”, but rather “let’s get to work”. You both have shown me the value of diligence and hard work. My committee, along with the University of Cincinnati, have not only given me the opportunity to change the trajectory of my career – but my life as well.

Russell, thank you for inspiring me to go back to school and finishing what I started. We can’t make any promises, but you can make me a drink.

Table of Contents

Figures and Tables

List of Figures

List of Tables

Background

After the 2016 Presidential and 2016 United Kingdom Referendum the social and technological landscape became unlike anything the world had seen before. As social media platforms, like Twitter, grew to an unpresented size the userbase and content became diluted with false and misleading content. Naturally, social media platforms like Facebook and Twitter become somewhat of an "echo chamber". Echo-chamber theory describes the naturally occurring separation of social circles (often referred to as *networks* or *clusters*) on social media. When users join a given platform, they tend to only follow and subscribe to users, political figures, and news organizations they tend to believe the most (Bode & Dalrymple, 2016). Once the user is surrounded by messages that only reinforce their social or political beliefs, they tend to accept content at face-value. Meaning that if a user is isolated in a particular cluster, they won't challenge a false or misleading post or story, but rather start to believe that others in their cluster are saying (Justwan et al., 2018). Finally, users start to disagree with any real information that challenges their social or political stances regardless of whether the information is true or not. In these clusters, political topics are not debated or challenged, but rather amplified and reinforced causing a natural social divide (Justwan et al., 2018).

The term "*Fake News"* became a part of almost every American's vocabulary due to the war unfolding on social media. However, few could find out where this fake news was coming from – until the presence of automated accounts, or "bots" became known. While tools and techniques used to detect bots had been developed pre-2016 and has continued to evolve, there is still a large gap between detecting bots and politically charged content for the average Twitter user.

Introduction

With the rise of internet usage and social media user demographics shifting, the way Americans consume news and form opinions has changed juristically in the last ten years. Consequently, with trust in mainstream news companies at a low, voters are turning to social media for information and news, especially in regard to politics (Cacciatore et al., 2018). In the major 2016 election cycle, which included the U.S. Presidential and United Kingdom Referendum, the use of automated accounts, or "bots", to spread misleading or false information across social media platforms was at all-time high. After major 2016 elections, which include the U.S. Presidential race and Brexit, Twitter released a list of 2,572 accounts linked to Russia that were used for the purpose of spreading propaganda in regard to U.S. politics (Stewart, Arif, & Starbird, 2018).While quite a few resources to detect bots like the list released by Twitter exist, many require quite a bit of programming experience to fully utilize. Highly popular versions of these tools, like Botometer for instance, can only search one user account at a time. Meaning the user must initiate the search on their own to find out if a given user or piece of content is from a bot. Other tools like developer API's can be used to detect bots but often times can confuse the user if they do not have a programming background, so they serve little use to average user. Browser extensions exist that inform the user of malicious content but often come up short or lack support for the ever-changing landscape of social media. This research will perform a comparative analysis of current and past bot detection and political affiliation screening to determine the software and information requirements to build a tool that can accurately detect bots and the nature of the content on their timeline.

Literature Review

I. Bots and the Current Political Climate

The rise of fake social media accounts, particularly on Twitter dates back to the early days of the platform. A simple internet search will turn up dozens of websites promising thousands of followers for very cheap. The website *Audience Gain* charges $75 per 1,000 followers, which one can assume are bots because normal, non-public figures do not usually obtain thousands of followers within hours of posting content. With basic programming knowledge, Twitter users can write scripts that create massive amounts of accounts. Early versions of bots could easily perform basic actions like follow massive amounts of users, and post content autonomously. Eventually, bots were programmed to mimic human behavior by responding to tweets with certain keywords, usually political or government related, and engage in conversation with predetermined responses (Haustein et al., 2016). This new power to send thousands of seemingly genuine messages to other users would now be used to mobilize users with false and misleading information.

Researchers as early as 2006 recognized the possibility of spreading misleading or false information via social media and effectively manipulating voter opinions (Howard, P. N., & Kollanyi, 2006). As technology advanced, source code for creating bots that can create content and interact as real people became readily available (Howard, Phillip, & Kollanyi, 2016). To the average user, these accounts seem to be real Americans with a deep interest in politics interacting with other real people. However, when looking at their activities and personality there are many attributes that set them apart from other users (Del Vicario 2017).

Key attributes are new accounts with high follower counts, non-genuine profile photos (or none at all), and uncommon usernames containing mostly numbers (Badawy, Ferrara, & Lerman, 2018). Since most users on social media use some form of their name, many bots especially those made with scripting will use uncommon names followed by a string of random numbers. For instance, "John Smith *(@JohnSmith1)*" vs *"@Smith65399733*". Their study also recognized the non-human like behavior of bots in terms of their profile page. Many people take time to customize their Twitter page with colors, backgrounds, etc., while bots are more likely to leave their page with default colors and images. What is particularly interesting about this finding is if bots are this influential in certain social clusters, many genuine human users were getting updates and political opinions from accounts with a vague name and no profile customization. Tools like the popular 'Botometer' use similar metrics to determine if a given Twitter account is a bot. Another tool, 'TweetCred', measures the validity of the content posted on twitter (Gupta, Lamba, & Kumaraguru, 2013). This tool is similar in nature to what we are analyzing in that it can detect "invalid" content, whether it be a news story or rumors about a large-scale crisis event (i.e. a natural disaster). While this study will further discuss methods in which we will detect bots and misinformation, these tools are important to mention because their functions will serve as the base of this study.

Studies performed in Europe observed a spontaneous emergence of politically charged accounts with massive amounts of followers focusing on segregating population along political party lines (Del Vicario et al., 2017). Interestingly enough, their discussions on echo-chamber theory and confirmation bias concluded that this was occurring naturally. Meaning that users, genuine and non-genuine, tend to only follow and interact with other accounts that polarizes their opinions. Which one can assume would allow groups of bots reinforcing genuine users' opinions

to flourish and gain traction. In regard to this study, it is important to note that is not a phenomenon only occurring in the United States but world-wide.

In the build-up to the major elections in 2016, which included the United Kingdom's referendum to leave the European Union -commonly referred to as 'Brexit' researchers began to question whether this phenomenon was happening outside of the United States. The popular scandal involving Cambridge Analytica, a London-based political consulting firm that found itself under fire for spreading false information on social media helped bring the conversation of bots and their voices in polarizing political climates outside of the U.S. Interestingly enough, in regards to Brexit it was bots spreading and amplifying the opinions of authentic users rather than the bots creating the misinformation themselves. A study examining these phenomena as it pertained to Brexit, *For Whom the Bell Trolls: Troll Behavior in the Twitter Brexit Debate* collected data from known trolls and compared their actions surrounding the U.S. 2016 Presidential election and Brexit. Their dataset found that 78% of the content discussing Brexit were retweet's of verified users (38.6%) then retweeted by bot. Another interesting finding by this group was that one bot account had been retweeted 186 times in their dataset, meaning 5.33% of the data captured was from one single account (Llewellyn, Cram, Favero et al., 2018). This is not to say the information bots were spreading was verified or true, but none the less bots were used to polarize social groups – those for and against Brexit.

Simultaneously, over in the during U.S., the Presidential primaries and elections the Twitter platform saw more activity from bots and automated accounts than ever before. At least 400,000 bots were creating over 3.8 million politically inclined tweets, making up almost 20% of Twitter's total volume of content (Bessi & Ferrara, 2016). Many of these accounts were traced back to foreign entities, like Russia's Internet Research Agency (or RU-IRA)(Stewart, Arif, &

Starbird, 2018). It is worth noting that in most studies involving Twitter and major elections researchers tend to group Twitter accounts and content into two clusters along U.S. political party lines. When looking at bot activity within their respective clusters, the top 2% of users were responsible for more than 60% of the activity (likes, retweets, comments) and content in their clusters (Chatfield et al.,2015). When we look at the activity of the two clusters, right-leaning (meaning U.S. Republican voters and pro-Brexit voters) were more than one and half times more active on the platform leading up to their respective elections (Bovet & Makse, 2019). Along U.S. political party lines, conservative or right-leaning users in southern U.S. states interacted with known bots significantly more than their northern, left-leaning counterparts. So much so, that bots targeting the U.S. South produced almost twenty times more content and interacted with 30 times more like-minded (real) users than left-leaning bots and users (Badawy, Ferrara, & Lerman, 2019).

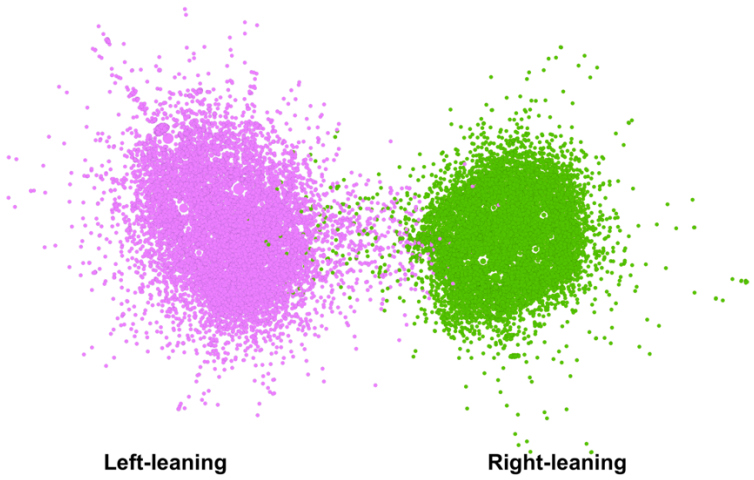
Aside from spreading misinformation to massive user groups, bots are particularly useful in reinforcing the beliefs of their particular political clusters. When groups of bots spread misinformation in their network, their messages are accepted at face-value (Bode & Dalrymple, 2016). Since distrust in mainstream news outlets is at an all-time low (Bode & Dalrymple, 2016), many people all over the world, including Americans, turn to social media for political commentary. Once a user joins a platform, like Twitter, that user is mostly exposed to content by sources they follow or subscribe to. This creates a filter-bubble in which the news one is consuming is filtered by confirmation bias, creating a network, or cluster in which they only hear news they agree with. Consequently, no one in a given cluster challenges the content or views of trolls, bots, or even real accounts.

Another social media platform, Facebook, saw substantial increases in new users that did not typically use the platform. By 2016 72% of U.S. adults aged 50-64 reported using the platform (Cacciatore et al., 2018). The information and news these new users were reading is heavily dependent on who they 'befriend', 'follow', or 'like'. Since users on Twitter or Facebook had once actively sought-out connections with these users, they accepted information from them at face value – whether it was true or not. Other studies, like those performed by Bode et al., concluded that Twitter usage involving politics is negatively correlated to mistrust in mainstream media, but also political *participation*. Meaning, those using the platform to consume and discuss political content are less likely to trust mainstream news sources, but also participate in the political process. Thus, not exposing these users to the factual, verified information about politics and the world around them (Bode & Dalrymple, 2016).

Parallel to the U.S. elections, other social movements were growing on Twitter in a seemingly organic, but quick fashion. *Black Lives Matter* and *Blue Lives Matter* were two major and often conflicting social movements growing at the same time. Studies performed by Stewart et al., concluded once again that major voices and active accounts in these movements were controlled by foreigners posing as U.S. citizens. Many suggest that these movements started organically, but were taken over and their views amplified by foreign states to create a social divide in the U.S. (Stewart, Arif, & Starbird, 2018). Figure 1 details political clusters on Twitter talking about *Black Lives Matter* (left) and *All Lives Matter/Blue Lives* Matter (right). Figure 2 decolorizes these clusters and isolates the known bots in these networks. Once again, reiterating that many of these bots position themselves in the center of the social and political discussion – thriving in an echo chamber environment.

Figure 1

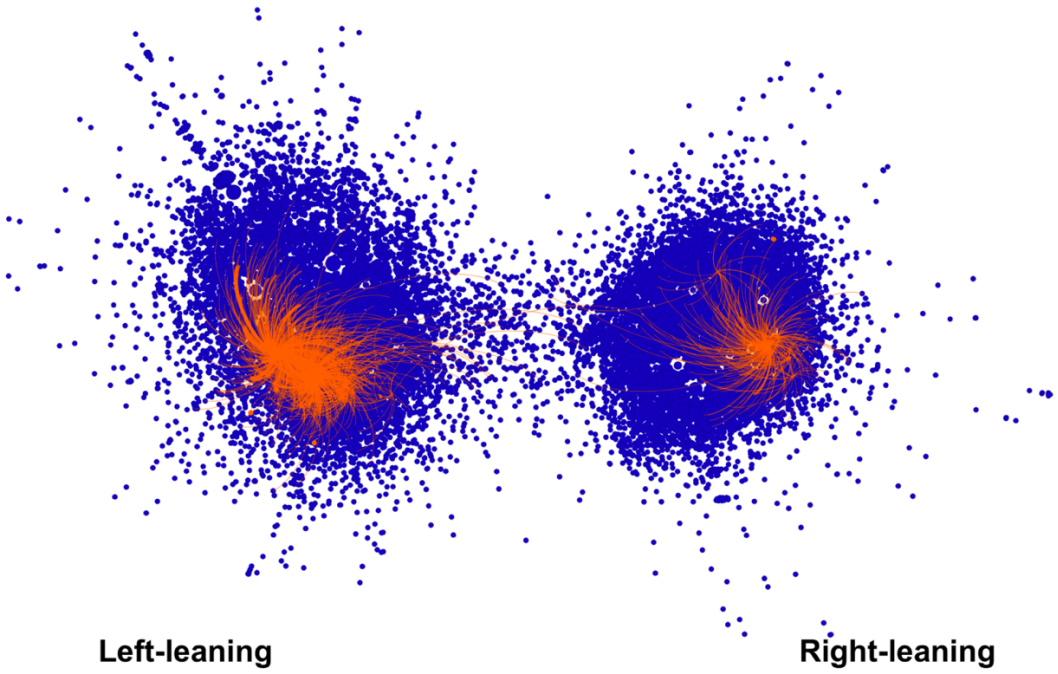
Echo Chamber Clusters



Source: Stewart, Arif, & Starbird, 2018

Figure 2

Echo Chamber with Bots



Source: Stewart, Arif, & Starbird, 2018

While there are tools like Botometer, there is no seamless way to determine if a Twitter account is a bot while scrolling through a timeline. If a user is suspicious an account on their feed is indeed a bot, they will have to navigate away from Twitter to Botometer where they search that particular user's handle (stylized with an '@' character) and view their score. The score given has high accuracy, but many users will not want, or know they can take these extra steps to verify users in their networks.

RQ1: What are the information requirements necessary to detect bots and political leanings of users by comparison analysis

Determining Political Leaning

While the tools exist that measure the probability a given account is a bot or a real user, we are able to determine the political leanings of a user and their network in a number of ways. According to the Pew Research Center and their research on U.S. Politics and Policy (2018), there are known differences between the two major U.S. political parties. As shown in Table 1:

Table 1

U.S. Political Party Demographics

Democratic	Republican
Women are more likely to identify as a Democrat (56%)	Men are more likely to identify as Republican (44% of men)
Black (84%), Hispanic (63%), and Asian (53%) likely to vote Democratic	White voters more likely to vote Republican (51%) than Democratic (43%)
College graduates mostly identify as Democratic (35%), Postgraduates identify even more (63%).	Highschool graduates mostly identify as Republican (47%), almost half of adults with some college experience as well (45%)
Millennials (born 1981-1996) are 51% more likely to vote Democratic	Persons born between 1921 and 1945 are more than likely to vote Republican
"Baby Boomers" (1946-1964) and "GenX" (1965-1980) are like to vote Democratic at 48%	Evangelical (77%) Protestant (53%) and Catholic (46%) are more likely to vote Republican
Likely to live in northeastern, and coastal states	Likely to live in the southern or central U.S. states

Source: Pew Research Center 2018

The above table explains that those who tend to vote or identify as Democratic tend be younger, have a college or postgraduate degree, and live in coastal or northeastern states. Those who identify as Republican tend to be older, have a high school education, and live in the southern or central U.S. Interestingly enough Pew also obtained user demographic data for the Twitter platform, amongst other social network platforms. According to their surveys, Twitter was the seventh most popular social media platform among U.S. adults – only ranking higher than WhatsApp and Reddit (2018). As shown in Table 2, the demographics of Twitter skew younger, and are college educated (2019).

Table 2

Social Media Demographics by Platform

Use of different online platforms by demographic groups

% of U.S. adults who say they ever use the following online platforms or messaging apps

	YouTube	Facebook	Instagram	Pinterest	LinkedIn	Snapchat	Twitter	WhatsApp	Reddit
U.S. adults	73%	69%	37%	28%	27%	24%	22%	20%	11%
Men	78	63	31	15	29	24	24	21	15
Women	68	75	43	42	24	24	21	19	8
White	71	70	33	33	28	22	21	13	12
Black	77	70	40	27	24	28	24	24	4
Hispanic	78	69	51	22	16	29	25	42	14
Ages 18-29	91	79	67	34	28	62	38	23	22
18-24	90	76	75	38	17	73	44	20	21
25-29	93	84	57	28	44	47	31	28	23
30-49	87	79	47	35	37	25	26	31	14
50-64	70	68	23	27	24	9	17	16	6
65+	38	46	8	15	11	3	7	3	1
<$30,000	68	69	35	18	10	27	20	19	9
$30,000-$74,999	75	72	39	27	26	26	20	16	10
$75,000+	83	74	42	41	49	22	31	25	15
High school or less	64	61	33	19	9	22	13	18	6
Some college	79	75	37	32	26	29	24	14	14
College+	80	74	43	38	51	20	32	28	15
Urban	77	73	46	30	33	29	26	24	11
Suburban	74	69	35	30	30	20	22	19	13
Rural	64	66	21	26	10	20	13	10	8

Note: Respondents who did not give an answer are not shown. Whites and blacks include only non-Hispanics. Hispanics are of any race.
Source: Survey conducted Jan. 8-Feb. 7, 2019.

PEW RESEARCH CENTER

Datapoints like this are crucial to understanding the makeup of the Twitter user base. With recent developments in more advanced technologies, like data mining, scrubbing, and

machine learningit is now easier than ever to determine a user's political leaning and that of their friends. A Twitter users' network can already be used to detect attributes like gender, ethnicity, and sexual orientation (Colleoni, Rozza, & Arvidsson, 2014) and with the development of machine learning researchers Pennachiotti and Popescu were able to apply these same techniques to determine a user's political orientation (2010). In other words, their team was able to determine the orientation of their network (accounts a user follows and is followed by) to confidently determine that user's political orientation. The *TwitterSearch* API is a great tool that allows the public to search for information on the social media platform. After registering for a Twitter Developer Account, one can use a Java program hosted locally on a personal computer to search for users, hashtags, or other data points. In Table 3 we see how each Twitter API function will be used to determine a user's political leaning:

Table 3

Twitter API Function Examples

Parameter/Reference	**Description**	**Purpose in This Study**
followers_ids	Returns the numeric ID that follows a user	Examine the user's followers
followers_list	Returns the screennames that follow a user	
friends_ids	Returns the ID's a user follows	Examine who the user follows
friends_list	Returns the screennames a user follows	
blocks_list	Returns the screennames and ID's a user has blocked	Determine who the user has blocked from their timeline
mutes_list	Returns the words, ID's and screennames a user blocked from appearing in their timeline	Determine the nature of the words and users the user has blocked from appearing from friends and followers

Another method involves capturing data in real-time from Twitter users as they post their content using the *TwitterStream API*. It can capture any non-protected (meaning from accounts that are not *private*) that contain any of the search terms determined by the researchers running the search. Maynard and Funk performed these actions but used an advanced method of machine learning to analyze the tweets they captured after searching for popular names and hashtags during an election in the United Kingdom. Once their data set had been thoroughly read by their algorithm, the tweets were given a "key" which labeled them by political party. An example of this Figure 3:

Figure 3

Machine Learning Example

"When they get a Tory government they'll be sorry."
<Person, Party, Affect> …anti Tory

Meaning their program knew the word "they" was distancing in nature, followed by a political party ("Tory"), then a negative affect ("sorry") which would deduce that this user is distancing themselves from the Tory political party this labeling them as pro-labour (Maynard & Funk, 2011).

Aside from users' content posted in their tweets, some tools collect basic metadata to classify users. A recent project completed at Virginia Tech by Pickett, Worden, and Wilborn (2019) discovered they could determine if a user was male, female, or a corporate brand. Datapoints like a user's name, screenname, profile image, brightness of profile image, and even number of "emoji" used were able to classify users with an accuracy of 89.72% (Pickett, Worden, & Wilborn, 2019). While the machine learning components of Maynard and Funk

(2011) are extremely useful in dissecting and classifying users and their content, metadata along with basic Twitter API functions will be enough to confidently classify most users.

When discussing detection of bots and genuine users' political inclination it's worth noting there can be quite a few inconsistencies in bot behaviors that make it difficult to determine their political leanings or intent. Pre-2016, many bots served multiple purposes meaning they could re-brand themselves and make it harder to determine their purpose. For instance, a bot could be used for a phishing scam by messaging random users to click on a link claiming to give away electronics or gift cards. If whomever is running the account needs to use that bot for another purpose they can re-brand the account to be a political bot (Grier, Thomas, & Paxson, 2010). However, when looking into changed screennames and profile branding we can see that a user is given an *ID* which is a searchable field when using popular Twitter API's. Meaning that if a bot has rebranded themselves by changing their screenname they can still be traced back to their *ID*. Using these services is more than likely the most accurate way to analyze patterns on social media (Llewellyn, Cram, Favero et al., 2018), instead of relying on archived datasets that may be incomplete. According to Twitter's Developer Agreement, content produced by now-suspended accounts is deleted within 24 hours. Although Twitter Inc. does archive their content, the dataset is usually incomplete (Llewellyn, Cram, Favero et al., 2018).

Bot Detection Methods

Before the methods in which bots are detected can be addressed, it is important to discuss how bots survive and find their place in a social network. Previous studies have noted that 92% of the accounts Twitter suspends for bot-like activities are suspended within three days of their first post (Chavoshi, Hamooni, & Mueen, 2016). Therefore, if a bot makes it more than one week

after producing content it will likely survive months, or even years on the platform (Chavoshi, Hamooni, & Mueen, 2016). Once a bot survives detection by Twitter and makes its way into a social network (meaning a network of mutually-followed and interactions between a particular group of users) the bots actions begin to differentiate it from authentic users.

First, the patterns and frequency in which bots post content on the platform are non-authentic. Excessive, automatic production of content usually outpaces what an authentic user would post (Chavoshi, Hamooni, & Mueen, 2016). On the contrary, a unique attribute of bots is the *lack* of original content. Meaning many bots produce little to no content of their own, but rather act as a vessel to spread content (by re-Tweeting) on the platform. This kind of bot, when looking at it's profile page, will show an excessive number of re-Tweets from other similar accounts in a very short period of time (Chavoshi, Hamooni, & Mueen, 2016) once again not acting in a genuine, human way.

Next, at surface level the differences between a suspected bot account and an authentic user are easier to catch. Most bot accounts that are easiest to spot have fake, or no profile images, along with usernames that are not commonly used. For example, a screenname with mostly numbers and a profile image of a blurry landscape – or none at all.

Once again, the industry and academic standard for Twitter bot detection is 'Botometer', out of Indiana University and the University of Southern California. Made public in 2014 via a simple website, the Botometer team has released a public API that many studies use to detect these accounts (Davis, Varol, & Ferrara, et al. 2016). While Botometer uses a very complex algorithms to determine the probability a given account on Twitter could be a bot, the criteria can be broken down by their six main classes: *Network, User, Friends, Temporal, Content, and Sentiment*.

Network, measures a user's network cluster, the usage of similar hashtags (stylized with the '#' symbol), number of re-tweets and degree of separation from accounts it is re-tweeting. Meaning if a small user cluster is constantly re-tweeting each other with similar content and hashtags there is a high probability it's operating in a cluster of bots. *User* measures the simple metadata for a given user. Metadata can be simplified as basic account information like screenname, account creation date, language, and geographic location. Other literature in this study has concluded that most bots have their geographic location turned off (Bessi & Ferrara, 2016). *Friends* measures a user's followers and friends. In the context of this study and Twitter as a whole, '*friend*' is defined as an account a given user follows – *follower* is an account that simply follows that given user. Normally a bot will follow (friend) hundreds, if not thousands of accounts but have very few followers. The Botomer *Friend* class also measures the number of posts a user is producing in relation to their follower count. If an account is producing an excessive amount of tweets but has very little followers, that would lead the program to believe the account could be a bot. This is closely related to *Temporal*, which measures things like how often a user tweets about a certain topic or uses a certain hashtag. The rate at which the account consumes content is taken in account here as well. Meaning a user is quickly re-tweeting and liking content faster than a genuine human account would.

The last parameter, *Sentiment*, was mentioned in previous literature when discussing Brexit (Maynard & Funk, 2011). This measures the emotion of a user's content by picking up on aggressive syntax, hate speech, emoji and exclamation mark usage. If a user is constantly posting aggressive content regarding the same topics (usually politically inclined) they could be a *troll* (an account that frequently engages in online arguments) which could be automated by a bot.

One all of these parameters are taken into account and measured, the tool can confidently predict if an account is a bot or not. However, given how accurate their measurements may be, the tool can only be used in two ways: evaluating one account at a time via their webpage, or using advanced programming that the average social media user may not understand. What is lacking in industry is a tool, particularly a browser plugin that uses Botomer's API to determine if an account a user follows is a bot in real-time via their timeline – as well as measure the political inclination of their feed.

It is important to note which data points are used to confidently determine if a user is a bot or not, however that is not the purpose of this work. While detection methods are becoming more and more advanced, the average Twitter user can most likely not use many of them. Which is why we want to determine the information and system requirements for such a tool.

Research Questions

The purpose of this study is to answer the following questions:

I. **What are the information requirements necessary to detect bots and political leanings of users by comparison analysis?**

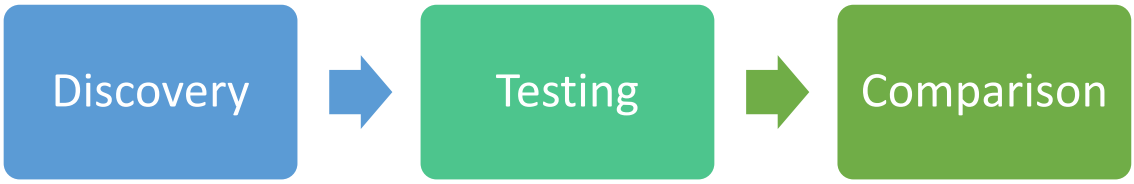
II. **What are the system requirements necessary to detect bots and political leanings of users on Twitter in real-time by comparison analysis?**

Methods

The main objective of this thesis is, by methods of comparative analysis, determine the design and system and information requirements to build a tool, more specifically a browser plugin, that can detect bots while also determining the user's political orientation in real-time. In the context of this work, *Information Requirements* encompasses the information the tool can discover as well as its delivery to the user. *System Requirements* are the "back-end" architecture of these tools, and how they programmatically measure the information and deliver it to the user. As noted in Figure 4, we will begin by searching and discovering which tools are publicly available. Next, we will detail how these tools work and how they were tested. Comparing installation, interfaces, support, and over all functionality. Finally, we will compare all necessary components of each tool before concluding our comparative analysis in *Results*.

Figure 4

Testing Methodology

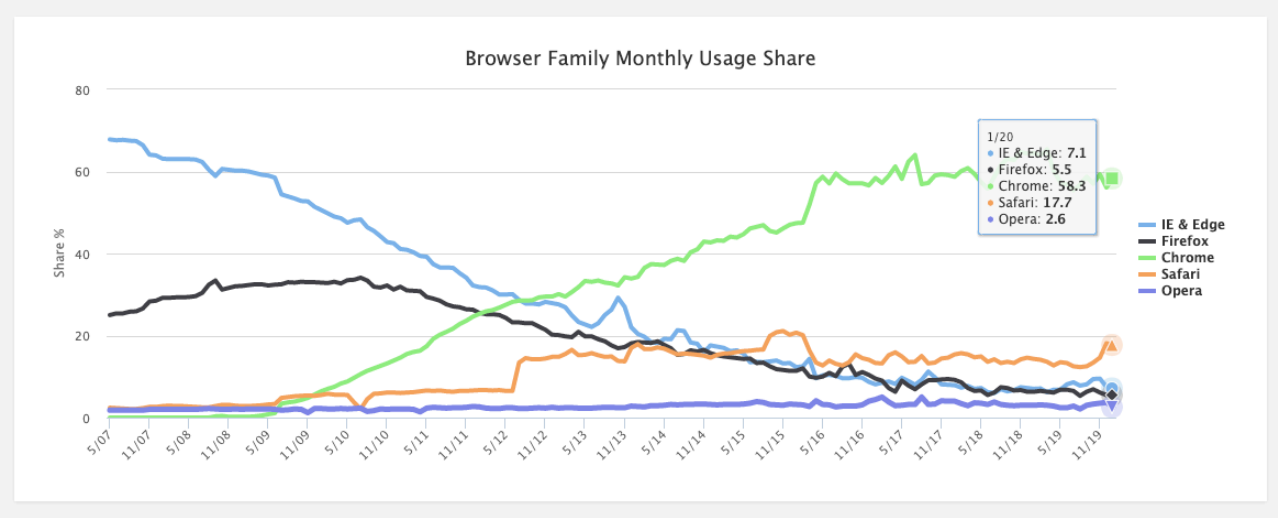


Browser Plugins

Based on the gap between highly complex systems that detect bots and political orientation and the average user, the use of a browser plugin will be the most efficient way to accurately help our users. The methodology behind researching browser plugins began by looking into the most popular browsers for desktop computers. W3Counter, a platform that measures the usage of different web browsers reported that Google Chrome was, and has been the most popular desktop browser in since 2012. As shown in Figure 5, the dominance of Google Chrome will lead for us to only reference plugins compatible with said browser.

Figure 5

Web Browser Usage



Since Google refers to browser plugins as *Extensions*, our platform will be referred to as such in this study from this point forward. Although an additional tool, Botometer, will be tested which is not supported on the Web Store or as an extension but will most likely be used in our final requirements.

Discovery

The search for these extensions was performed on the Google Chrome Web Store. This is an online marketplace where Chrome users can download free or paid add-ons for their browser. These include things like color themes, accessibility tools, language translators, weather forecasting icons, and many more. Our search terms used to discover the current offerings on the Web Store were “bot”, “bots”, “Twitter”, “political”, and “politics”. Our search yielded *Bot Sentinel*, *PolitEcho, BotCheck.me*, *Botson,* and Botometer will be tested but is not supported on the Web Store. The methodology behind using primarily Chrome extensions is to find the gap in the product offerings to the public who want to analyze their Twitter feed.

Testing

The selected applications will then be tested on a 2017 MacbookPro with 8GB of memory running macOS Mojave, 10.14.6. Google Chrome will be used for searching on the Web Store as well as testing of the extensions and other web-based tools. We are also currently using the most up to date version of Chrome (80.0.3987.116 64-bit) at the time of this study. For research integrity, all products will be tested on the same machine with the same accounts. Accounts we can confidently assume are bots, as well as genuine users will be used to gauge the product’s scoring. Then, the way in which the product alerts the user of bots or political scores will be compared. Certain features of each tool will be considered more important than others. For instance, the way in which the user is notified of bot-like or politically charged content will be thoroughly analyzed because delivering these findings to users can be an information bottleneck. Finally, we will compare which products are still supported by their developers and if upcoming releases are probable.

Analysis

Finally, all of the tools and products will be compared using the same criteria. We will be looking for the type of interface, does it detect bots, does it detect political leaning, how does it notify the user (*alert type*), and if the product is still available and supported. These comparisons will help us better understand the information and system requirements addressed in our research questions (*RQ1*, *RQ2*). A secondary analysis will be conducted regarding the programming architecture of these tools. This will also help us understand the *System Requirements* portion of our reearch questions. Comparing the programming languages and known API usage will better help us answer our research questions and understanding how these tools programmatically analyze information then render it to the user is crucial to this study. Without a solid understanding of these processes, we would have incomplete requirements.

Results

The results of this information and systems analysis will be tabled using the comparison methodology described in the previous section. Then, from our comparison analysis, we will list out the system requirements for such a tool that detects bots and political leaning of a user's feed simultaneously as it pertains to each research question.

Comparison Results

The first product tested was Bot Sentinel, which is highest-rated bot detector on the Chrome Store. While their source code is not open source, their tool was available for testing and

analyzation. This tool does not appear to use Botometer or any other bot detection API but does edit the rendering of the Twitter webpage. Placing their *Check User* box on the profile page is convenient, but we found no way to check a user without clicking on their actual profile. Figure 6 (left) shows the Reuters News Twitter page and the rendering of their tool, Figure 6 (right) details their pop-up element used to display their results:

Figure 6

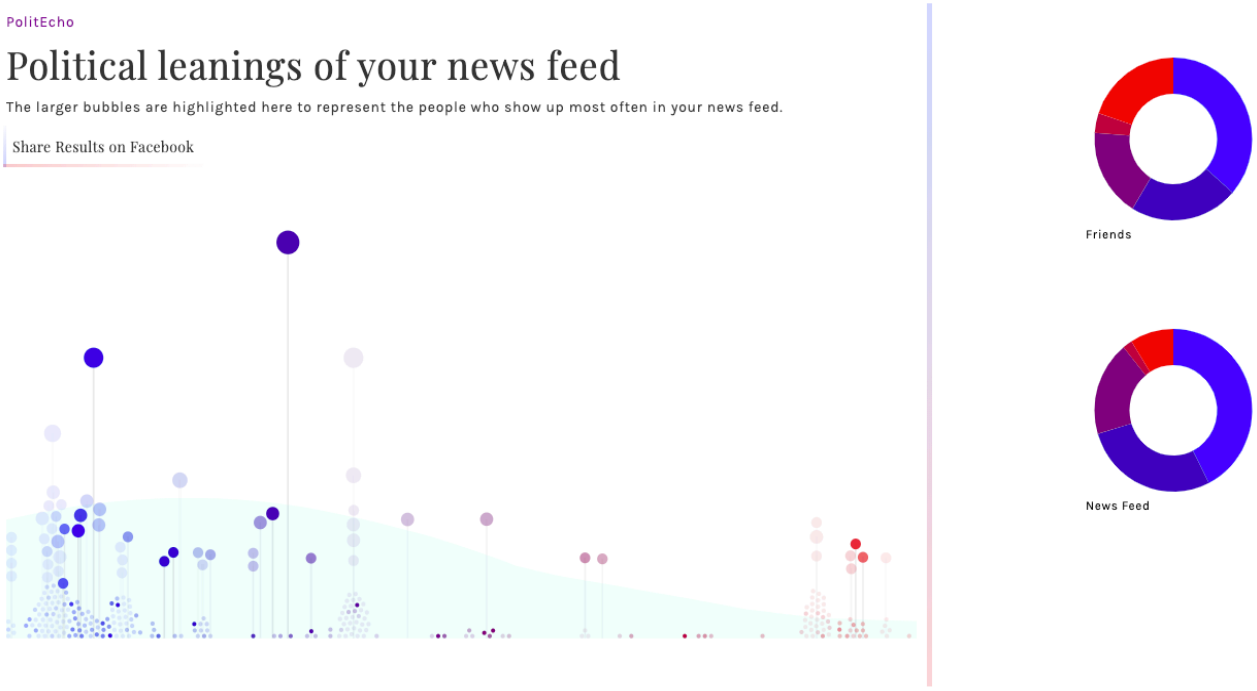
Bot Sentinel Elements

According to the Bot Sentinel *Frequently Asked Questions,* their tool uses machine learning to detect bots and lists no specific API or other service (Bot Sentinel, 2020). Also, Bot Sentinel manages a Twitter account *(@BotSentinel*) which posts content relating to bots, trending topics among bots, and suspected bot accounts that have been suspended by the platform. Bot Sentinel also supports a plugin for Mozilla's Firefox browser, but as explained in *Figure 3* Firefox is only used by 5.5% of desktop users.

The second product tested was *PolitEcho*. It had a similar installation to the other extensions' process via the Chrome Store. Next, we clicked on the button PolitEcho provides on Chrome's user interface. It guides the user through some prompts (service agreements) before analyzing the user's friend list. Much like Botometer this required the user to navigate away from their social media feed and use a completely different webpage. After a few seconds, the users' friends data is displayed in a graph (Figure 7). This data collection method is is an accurate reflection of the user's social network (Facebook page follows and friends' page follows) but does not allow for simultaneous browsing of a timeline and analysis.

Figure 7

PolitEcho Graph Interface



Although the source code has not been updated in almost three years (March of 2017) we can analyze its build and how we could utilize this open-source code (PolitEcho, 2017). This particular tool used JavaScript to read through a dictionary of news outlets, popular news pages, and politicians' Facebook pages and scored them categorized them into four groups. Then, using

a scoring method the script would render a message in HTML format to tell the user the score of their friend list (PolitEcho, 2017). The scoring method will be beneficial to this project because our proposed tool will show the political leanings and orientation, not just "Right" or "Left", but how far "Right" or "Left". One shortcoming of this product, aside from its lack of support since 2017, is the categorization of news sources and other pages. When developing our tool, ensuring correct categorization of the entire political spectrum will be absolutely necessary.

The next product tested from the Web Store was *BotCheck.me*, a bot and spam detection extension. After following the steps to install, we activated the extension by navigating to Twitter and searching "#Election2020". Once a tweet was found that contained our search hashtags and would be deemed a "Bot" we clicked on the check button the extension adds to the website. It opens a small popup element within the Twitter window and shows the user the key to their labeling system (if a given profile has been scanned by BotCheck it alerts the user) (Figure 8). Next, it renders a score in the same dialog box element (Figure 9).

Figure 8

Botcheck Popup Element

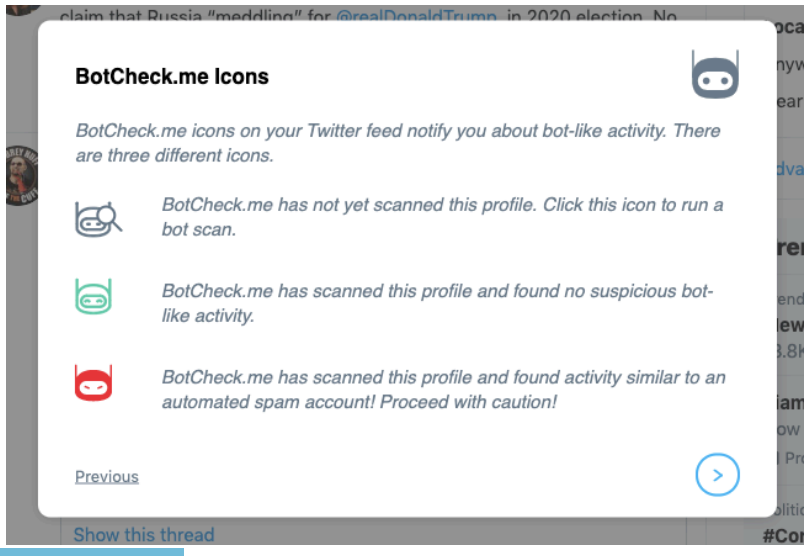
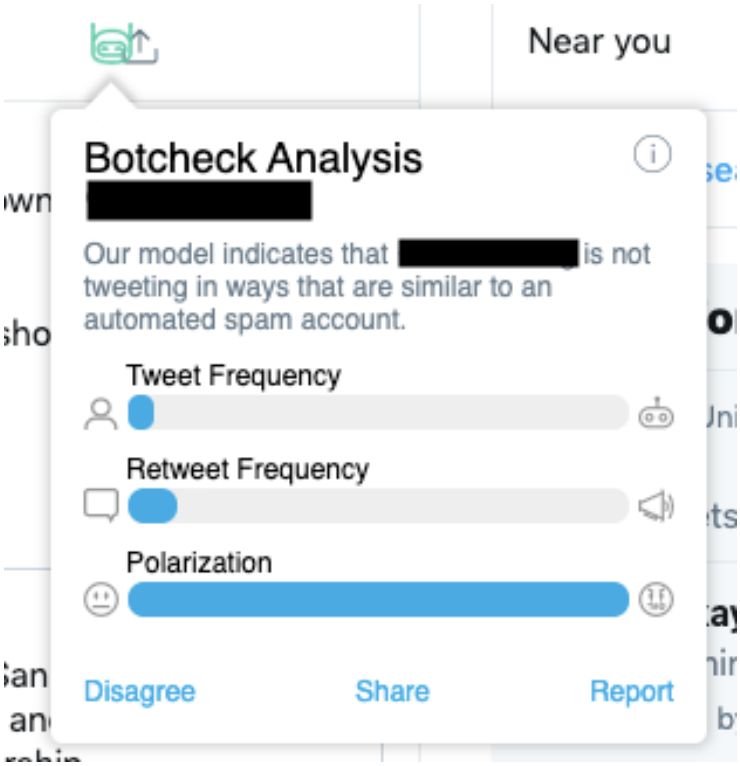


Figure 9

Botcheck Analysis Popup Element

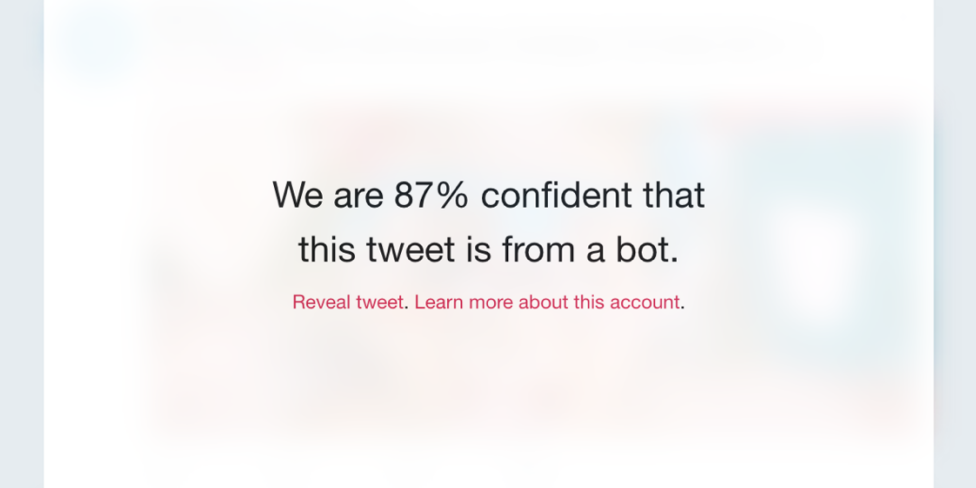


As we tested, we realized their scoring system ranks accounts based on their tweet frequency, retweet frequency, and polarization (BotCheck, 2019). While polarization is not the same as detecting political leaning because it is scoring the emotion regardless of political spectrum. A tool detecting political leaning could find this methodology useful, especially if the user wanted to know how polarizing or emotional levels of their feed and friends.

The fourth product tested was *Botson*, an extension found on the Web Store. Botson claimed to detect bots in real-time on a user's Twitter timeline. While the user scrolls and clicks on a certain tweet, they are notified via a popup window saying the user that posted that particular tweet has a high probability of being a bot (Botson, 2017). Figure 10 is an example of the user interface:

Figure 10

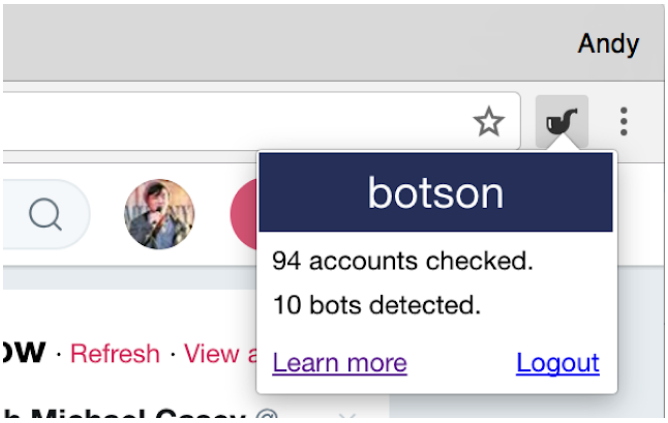
Botson Full-Window Alert Example



The Boston open-source software documentation explains in detail how their product works, which API's they have integrated, and the product limitations. First, they use the Botometer parameters and API to score users appearing on the screen using JavaScript on the back-end of their application. Then, using HTML and JavaScript the tool alerts the user of a possible bot. Users can be alerted in two ways: a window-width popup (Figure 10) or an alert in the corner of the screen (Figure 11).

Figure 11

Botson Popup Element



While the source code and information were easy to find, Chrome no longer supported this extension so it could not be tested.

Lastly we tested Botometer, a popular web interface and public API. Although Botometer is not a browser extension it has a long standing commercial and academic reputation for its accuracy, which is why it is included in this study's testing (Botometer, 2020). The Botometer interface is a simple webpage where the user types a Twitter screenname into the search bar for analysis. Next, the interface allows users to check the account they entered, the friends of that account, or the followers of that account. The user is shown a score from Botometer, gauging how likely they are to be a bot (Figure 12). Users can also click on *details* and that particular account's information will be displayed in the same window (Figure 13).

Figure 12

Botometer User Score

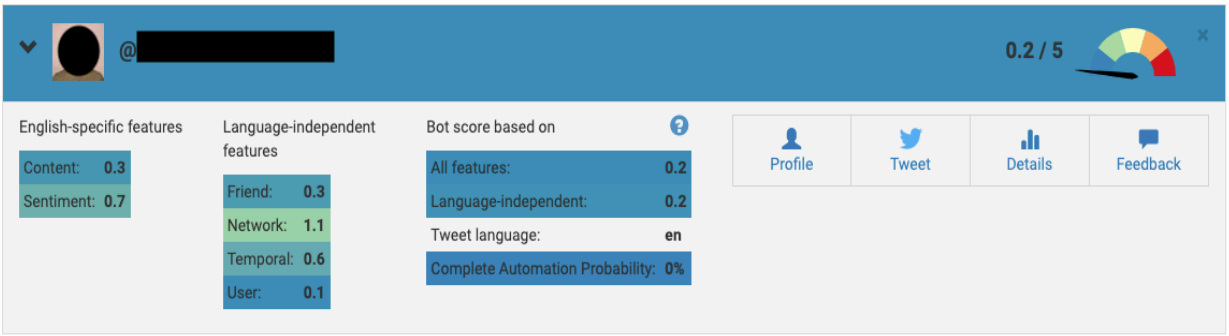


Figure 13

Botometer User Score Detail

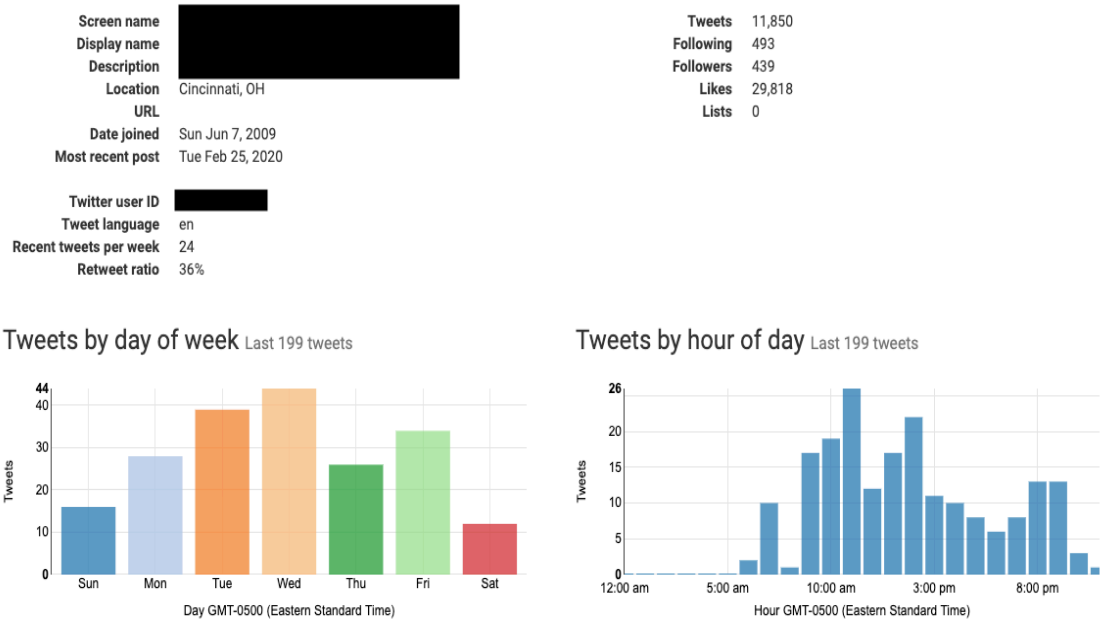


Table 4 and Table 5 are the final comparison analysis. Starting by comparing the information provided (*RQ1*) and the interface for these tools. Next, the system requirements were compared in Table 5 (*RQ2*). It is important to note how these tools operate programmatically, which or if they use an external API, so this work can arcuately determine the correct requirements. It is worth noting that tools that detect political leanings were not tested or ranked differently than tools that detect bots because our final requirements are for a tool that will do both.

Table 4

Tool Overview Comparison

Name	Type of interface	Bot Detection?	Political Leaning Detection?	Alert Type	Still Supported?
Bot Sentinel	Chrome Extension; adds button to profile page	Yes	No	Popup	Yes
PolitEcho	Chrome Extension; separate webpage (does not alter the rendering of social feed)	No	Yes	Interactive graph	Yes
BotCheck.me	Chrome Extension; adds button to Twitter page	Yes	No – although does check for polarization	Popup window	Yes
Botson	Chrome Extension	Yes	No	Popup window	No
Botometer	Webpage and API	Yes	No	Interactive chart	Yes

Table 5

Tool Programming Comparison

Name	Known API Plugin	Back-End	Front-End
Bot Sentinel	No	JavaScript	HTML
PolitEcho	No	JavaScript	HTML / Java Script
BotCheck.me	No	JavaScript	JavaScript
Botson	Yes – Botometer	JavaScript	HTML
Botometer	Yes – Botometer	R, Python	NodeJS, HTML

Research Question I: What are the Information Requirements Necessary to Detect Bots and Political Leanings of Users on Twitter in Real-Time by Comparison Analysis?

The first research question (*RQ1*) is answered by our analysis of the tools and products in the previous section. From the literature, our testing and comparison, we can see that the Botometer API will ultimately serve as the basis of out bot detection component. Given Botometer's academic and commercial success and reliability (within 98%), we see no reason not to use their services. Due to the transparent nature of their research and open source software, this will be a cost-effective choice as well as an easier way to provide upgrades and produce new releases of our tool as bot detection technology advances. This is an information requirement for *RQ1*.

For the political leanings detection aspect of this tool, we will need to start with a ranking system like PolitEcho. Their system involves ranking verified political figures, news sources, and other pages by a numbering system. In Figure 14, we see a section of their code which ranks these data points in groups: `ul` (more left-leaning), `l` (left leaning), `c` (conservative), `uc` (more conservative).

Figure 14

Political Leanings Datapoints

```
news_dict["5281959998"] = ul; //New York Times
news_dict["19013582168"] = ul; //PBS
news_dict["18468761129"] = ul; //HuffPost
news_dict["56845382910"] = ul; //HuffPost Politics
news_dict["6250307292"] = ul; //Washington Post
news_dict["374111579728"] = ul; //Wasington Post Politics
news_dict["6013004059"] = ul; //The Economist
news_dict["62317591679"] = ul; //Politico
news_dict["273864989376427"] = ul; //MSNBC
news_dict["5550296508"] = ul; //CNN
news_dict["219367258105115"] = ul; //CNN Politics
news_dict["155869377766434"] = ul; //NBC News
news_dict["197311240419563"] = ul; // Late Night with Seth Meyers
news_dict["545775132233909"] = ul; //Late Show with Stephen Colbert
news_dict["1765033567057615"] = ul; //Full Frontal with Samantha Bee
news_dict["479042895558058"] = ul; //Last Week Tonight with John Oliver
news_dict["445821135487302"] = ul; //Inside Amy Schumer
news_dict["908009612563863"] = ul; //NowThis Election
news_dict["223649167822693"] = ul; //Vox
news_dict["10606591490"] = ul; //Time
news_dict["5863113009"] = ul; //LA Times
news_dict["18343191100"] = ul; //Newsweek
news_dict["167115176655082"] = ul; //Vice
news_dict["174742062548592"] = ul; //Mic Media
```

Figure 14 Cont.

```
40
41  news_dict["131459315949"] = l; //CBS News
42  news_dict["19440638720"] = l; //Wired
43  news_dict["20446254070"] = l; //Business Insider
44  news_dict["266790296879"] = l; //Bloomberg
45  news_dict["1481073582140028"] = l; //Bloomberg Politics
46  news_dict["86680728811"] = l; //ABC News
47  news_dict["13652355666"] = l; //USA Today
48  news_dict["21898300328"] = l; //Buzzfeed
49  news_dict["354263831266028"] = l; //Buzzfeed Politics
50  news_dict["29259828486"] = l; //The Atlantic
51
52  news_dict["338028696036"] = c; //Yahoo News
53  news_dict["8304333127"] = c; //Wall Street Journal
54
55  news_dict["15704546335"] = uc; //Fox News
56  news_dict["102533606954"] = uc; //Drudge Report
57  news_dict["95475020353"] = uc; //Breitbart
58  news_dict["80256732576"] = uc; //Info Wars
59  news_dict["136264019722601"] = uc; //Rush Limbaugh Show
60  news_dict["140738092630206"] = uc; //The Blaze
61  news_dict["69813760388"] = uc; //Sean Hannity Show
62  news_dict["36400348187"] = uc; //Glenn Beck Program
63  news_dict["85452072376"] = uc; //NewsMax
64  news_dict["182919686769"] = uc; //DailyCaller
65  news_dict["912274352202712"] = uc; //DailyCaller Politics
66  news_dict["35994014410"] = uc; //The Washington Times
67  news_dict["408250066356"] = uc; //Bill O'Reilly
68  news_dict["123624513983"] = uc; //Western Journalism
69  news_dict["519305544814653"] = uc; //Conservative Tribune
```

Using a JavaScript dictionary will allow the tool to scan for these pages on the user's Twitter friend and follow list, and scoring will be needed. In addition to these pages, we will need to score politically charged hashtags. This is the information requirement for *RQ1*. Using an external R or Microsoft SQL database will probably not be needed because we are not providing an API for public use. Meaning the data needed to score these pages is not as complex as discovering bot accounts and will only be local to this project. Next, we will address our findings and the system requirements for such a tool. This will address research question two (*RQ2*).

Research Question II: What are the System Requirements Necessary to Detect Bots and Political Leanings of Users on Twitter in Real-Time by Comparison Analysis?

The second research question is answered by our comparison of the products and tools tested in the previous section. First, given our research in browser usage which revealed a dominance in the market by Google Chrome, we will move forward with the recommendation of using a Google Chrome Extension as the interface for this tool. The first step in answering *RQ2* is explaining the process of publishing an extension on the Web Store. Uploading an extension to the Google Chrome Web Store is a fairly easy process. Google requires developers to have an active Google account and pay a one-time $50.00 fee. Once that developer account has paid their fee, it is free to upload items to the Web Store (Chrome Web Store, 2020).

After the requirements for the Web Store are met, we will discuss the programming component of our analysis. Since our testing and research into the open source files determined that most of these tools are written in JavaScript we will move forward with that requirement. Additionally, we will use React JavaScript (known as *ReactJS*) to tie the back-end scripting to an HTML element that alerts the user. Much like how the Bot Sentinel extension places a button on an accounts profile page to allow the user to check their bot analysis score, we will require a button but will also flag screennames and user ID's that can alert the user in real-time if a known-bot is appearing on their timeline Programmatically, this is using the ReactDOM to edit predetermined HTML elements on a page. Twitter labels their tweets as a `<div>`, ID'ing them as `tweet` – so it is required to target this element to accurately place our alerts.

Results Conclusions

With the system and information requirements in place, we are now able to determine an over encompassing requirements table. Combining both the system and information requirements from *RQ1* and *RQ2* it will help simplify and organize the requirements in one list. Our overall requirements table, Table 6, will address scope, user-interface, structure, programming languages, bot detection, and political leaning detection. As expected, this tool will require a JavaScript back-end (that will also serve a political detection purpose), a ReactJS/HTML front-end, with a Botometer tie-in. This study has also reiterated the required user-interface elements - that do not navigate the user away from their social feed and provide a seamless user experience. These information and system requirements, combined, will bridge the previously mentioned gap between accurate detection and seamless user interface.

Table 6

Requirements Analysis

Element	***Requirement***
Scope	A Chrome browser extension that can accurately detect bots and the political leaning of a user's Twitter timeline
Graphic User Interface (GUI)	Will alter the HTML elements on the user's timeline, adding alerts in-line with user/screen names. Political leaning score will need to be revealed as an HTML popup element, which is near the address bar where extension buttons live in the browser
Structure	The Botometer API will be used in conjunction with a local dictionary, scanning the browser window for Twitter screen/usernames as well as hashtags It will be hosted by Google via Web Store
Languages	JavaScript, ReactJS, and HTML/CSS
Bot Detection	Botometer API
Political Detection	Dictionary based on the open source code used for PolitEcho
Windows Version	Windows 7 or newer*
Mac Version	Intel MacOS Yosemite 10.10 or newer*

**Source: Google Chrome Help Documentation*

Discussion

As discussed in previous sections, the requirements to build such a tool are based off of what is currently available for users and the gaps in offerings. A tool like this does differ from what is currently available to the public, but the requirements are similar to other tools especially when referring to browser extensions. As predicted, most extensions use a JavaScript back-end and tie into an API, locally hosted dictionary, or arrays that the front-end can reference.

Bot Detection

When testing these tools, we found the information requirements for bot detection to be closely aligned with what we had hypothesized. Botometer, a widely used and studied tool that offers a free API would lead researchers and developers to believe that some, if not most bot detection tools would be using their services in one way or another. The tools that did not credit nor specifically mention Botometer (BotCheck.me, Bot Sentinel) use machine learning and training data sets. This is a much more advanced method of bot detection, requiring machine learning knowledge, time to train the program on which accounts are actually bots or genuine, and large data sets. As expected, using Botometer API is easier, cost effective, and can be predicted with a 98% chance of accuracy. When testing various accounts with these tools, Botometer detected more suspected bot accounts and flagged them as such, than the other tools that detected bot-like activity (Appendix B). Interestingly enough, the now defunct Botson is featured on Botometer's website, under *Friends* where they have listed other tools used for bot detection. When answering *RQ1*and *RQ2*, using a simple yet reliable service like the Botometer API will satisfy the information and system requirements.

Political Leaning Detection

The lack of available tools for determining political leaning was an interesting finding. While we did find and test PolitEcho, it is purposed for use with Facebook, not Twitter. During testing, we again found that we had to navigate away from our social media feed to complete their analysis. The information provided by their tool was accurate and informing, it did not integrate well with the user experience of scrolling through a news feed. Determining political leaning, in regard to this tool, will need to be done by writing and keeping an up to date data set. Although this is more time consuming than using a publicly available API, we can regularly update and spot-check for errors – thus ensuring the reliability of our data. Answering *RQ2*, the system requirement for political detection will be an internal dictionary. With regard to *RQ1*, the information requirement is a thorough list, that is accurately ranked, sorted, and updated to provide accurate scoring.

Additional Findings for Discussion

When performing our initial search for tools of this nature, we did not expect our search to narrow down to using exclusively Google Chrome. Then, while searching on the Google Chrome Web Store for the term "politics" the results were mostly (75%) extensions that "hide" political posts from a user's newsfeed; all of the search results were products for use with Facebook. Another interesting finding by our search methodology were the nature of the Twitter extensions available on the Web Store. The searched yielded extensions that can perform tasks like "Bulk auto-follow" which follows every account the user see's in the view window while on Twitter. There were tools that "auto-liked" tweets en masse which could also be used on Instagram's desktop version. Alarmingly enough, we found tools that could be used to post

content/Tweet, follow and like autonomously. Ironically, these are all characteristics of bots which were discussed in the literature.

Finally, we found a seemingly high number of social media plugins on the Web Store that were no longer updated since mid-2017. These extensions were still able to be added to our browser, but the latest release for the majority of the available extensions was 2017. When answering *RQ1* and *RQ2*, it is of upmost importance to keep releasing updated versions of this tool to ensure correct scoring and support for the users.

Conclusions

To summarize, this work observed the recent phenomena of automated troll accounts (bots), their roles in social media and how they enable political discourse. While social media users isolate themselves in their previously discussed echo-chambers, exaggerated and sometimes false content is no longer contest thus believed to be true (Bovet & Makse, 2019). When observing and further testing the available tools that determine whether a user is a bot or genuine user, we found a gap in ease of use and reliability. We also found a large gap in research and product offering that determines the political leanings of a user and or their Twitter feed. Complicated API's that require programming knowledge were one set of tools, the other, more user-friendly tools were outdated or inconvenient.

This work took note of these gaps and set forth system and information requirements that could build such a tool. One that bridges the high accuracy of one tool and the usability of another. What current offerings lack this work took into account. Scope of this tool, as well as a proposed graphic user interface, programming languages, and data-points for scoring were

addressed and set forth as requirements. In the literature, we discussed the rise of automation in social media and how easily one can obtain such technology – even using a Chrome extension to run a bot account. If automation is on the rise for the everyday user, awareness and detection should be as well.

Limitations

This study encountered limitation by means of lack of user testing of these tools. We tested the tools and took detailed notes, and even detailed their user interface but a small focus group could help understand the complete usability picture. Another limitation is the lack of dataset used for testing. A *control group* of data, containing known bots, genuine users, and their political affiliations could be used to test the accuracy and scoring of these tools. While a control set would be ideal, Twitter usually detects and suspends bot accounts themselves. Once the account is removed from the platform, their content is archived and accessing a complete data set from an archine could be difficult. Consequently, a control set would have to be collected, updated, and tested quickly given the short shelf life of these accounts.

Other miscellaneous limitations include the short time to conduct this study, the university semester schedule and curriculum alignment. Ideally, expansions on this work would take longer than the current proposed timeline to complete the Master of Science degree (two semesters) and receive funding for development and focus group design critiques. No funding or grants were requested from the University of Cincinnati or any other industry partners.

Future Works

The growing field of social media and social network analysis will only foster the expansion of this study. While this tool is not in a development phase, naturally the next steps would be to do so. Once the product is completed, it can be hosted on the Google Chrome Web Store and testing could begin. As discussed in the previous section, testing could include user focus groups, which could provide insight into the tool's user-interface and its perception, as well as the usability of this tool. This would require approval from the University Institutional Review Board.

Also, this tool could be used in a study to examine the political nature of bots and genuine users on Twitter. Insight into the demographics, behavioral patterns, and content detected for the users of this product would produce a wealth of information. Long-term, if this product could start determining political leanings and orientations correctly, along with a large enough user base, this work could be the starting point for a public API that other developers and institutions could use.

Reference

Badawy, A., Ferrara, E., & Lerman, K. (2018). Analyzing the digital traces of political manipulation: The 2016 Russian interference Twitter campaign. *Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2018*, 258–265. https://doi.org/10.1109/ASONAM.2018.8508646

Bessi, A., & Ferrara, E. (2016). Social bots distort the 2016 U.S. Presidential election online discussion. *First Monday*, *21*(11). https://doi.org/10.5210/fm.v21i11.7090

Bode, L., & Dalrymple, K. E. (2016). Politics in 140 Characters or Less: Campaign Communication, Network Interaction, and Political Participation on Twitter. *Journal of Political Marketing*, *15*(4), 311–332. https://doi.org/10.1080/15377857.2014.959686

Bovet, A., & Makse, H. A. (2019). Influence of fake news in Twitter during the 2016 US presidential election. *Nature Communications*, *10*(1), 1–14. https://doi.org/10.1038/s41467-018-07761-2

BotCheck.me. (n.d.). Retrieved from http://botcheck.me/

Botson Extension. (n.d.). Retrieved from https://github.com/lambtron/botson-extension

Bot Sentinel Dashboard. (n.d.). Retrieved from https://botsentinel.com/

Cacciatore, M. A., Yeo, S. K., Scheufele, D. A., Xenos, M. A., Brossard, D., & Corley, E. A. (2018). Is Facebook Making Us Dumber? Exploring Social Media Use as a Predictor of Political Knowledge. *Journalism and Mass Communication Quarterly*, *95*(2), 404–424. https://doi.org/10.1177/1077699018770447

Chatfield, A. T., Reddick, C. G., & Brajawidagda, U. (2015). *Tweeting propaganda, radicalization and recruitment*. 239–249. https://doi.org/10.1145/2757401.2757408

Chavoshi, N., Mueen, A., & Hamooni, H. (2016). *Identifying Correlated Bots in Twitter*. *1*(November). https://doi.org/10.1007/978-3-319-47874-6

Chrome Web Store. (n.d.). Retrieved from https://chrome.google.com/webstore/category/extensions

Colleoni, E., Rozza, A., & Arvidsson, A. (2014). Echo Chamber or Public Sphere? Predicting Political Orientation and Measuring Political Homophily in Twitter Using Big Data. *Journal of Communication*, *64*(2), 317–332. https://doi.org/10.1111/jcom.12084

Davis, C. A., Varol, O., Ferrara, E., Flammini, A., & Menczer, F. (2016). *BotOrNot: A System to Evaluate Social Bots*. 4–5. https://doi.org/10.1145/2872518.2889302

Del Vicario, M., Gaito, S., Quattrociocchi, W., Zignani, M., & Zollo, F. (2017). *Public discourse and news consumption on online social media: A quantitative, cross-platform analysis of the Italian Referendum*. Retrieved from http://arxiv.org/abs/1702.06016

Ferrara, E., Varol, O., Davis, C., Menczer, F., & Flammini, A. (2016). The rise of social bots. *Communications of the ACM*, *59*(7), 96–104. https://doi.org/10.1145/2818717

Gain More Twitter Followers. (n.d.). Retrieved December 1, 2019, from https://audiencegain.com/twitter-followers/

Google Chrome - The New Chrome & Most Secure Web Browser. (n.d.). Retrieved from https://www.google.com/chrome/

Grier, C., Thomas, K., Paxson, V., & Zhang, M. (2010). *@spam: The Underground on 140 Characters or Less*. https://doi.org/10.1007/978-94-007-7615-9_11

Gupta, A., Lamba, H. and Kumaraguru, P. $1.00 per RT #BostonMarathon #PrayForBoston: Analyzing fake content on Twitter. eCrime Researchers Summit. IEEE (2013)

Haustein, S., Bowman, T. D., Holmberg, K., Tsou, A., Sugimoto, C. R., & Larivière, V. (2016). Tweets as impact indicators: Examining the implications of automated "bot" accounts on Twitter. *Journal of the Association for Information Science and Technology*, *67*(1), 232–238. https://doi.org/10.1002/asi.23456

Howard, P. N., & Kollanyi, B. (2016). *Bots, #StrongerIn, and #Brexit: Computational Propaganda during the UK-EU Referendum*. Retrieved from https://t.co/ysoqi9MoQ0

Justwan, F., Baumgaertner, B., Carlisle, J. E., Clark, A. K., & Clark, M. (2018). Social media echo chambers and satisfaction with democracy among Democrats and Republicans in the aftermath of the 2016 US elections. *Journal of Elections, Public Opinion and Parties*, *28*(4), 424–442. https://doi.org/10.1080/17457289.2018.1434784

Llewellyn, C., Cram, L., Favero, A., & Hill, R. L. (2018). *For Whom the Bell Trolls: Troll Behaviour in the Twitter Brexit Debate*. Retrieved from http://www.parliament.uk/documents/

Maynard, D., & Funk, A. (2011). Automatic detection of political opinions in tweets. *CEUR Workshop Proceedings*, *718*, 81–92.

Pennacchiotti, M., & Popsecu, A.-M. (2010). A Machine Learning Approach to Twitter User Classification. *Fifth International AAAAI Conference on Weblogs and Social Media*. https://doi.org/10.1097/00002060-199609000-00002

Pew Research - Trends in party affiliation among demographic groups. (2019, December 31). Retrieved February 24, 2020, from https://www.people-press.org/2018/03/20/1-trends-in-party-affiliation-among-demographic-groups/

Pickett, G., Worden, K., & Wilborn, A. (2019). *Virginia Tech Online Role-related User Classification on Twitter*.

PolitEcho - Is your news feed a bubble? (n.d.). Retrieved from https://politecho.org/

Realtime Tweets Overview - Twitter Developers. (n.d.). Retrieved January 13, 2020, from https://developer.twitter.com/en/docs/tweets/filter-realtime/overview

Search Tweets Overview - Twitter Developers. (n.d.). Retrieved November 20, 2019, from https://developer.twitter.com/en/docs/tweets/search/overview

Stewart, L. G., Arif, A., & Starbird, K. (2018). Examining Trolls and Polarization with a Retweet Network. *Proceedings of WSDM Workshop on Misinformation and Misbehavior Mining on the Web (MIS2)*, 6. https://doi.org/https://doi.org/10.475/123_4

Appendix A

Term	*Definition*
Tweet	The content created by users on Twitter. Users are limited to 280 character posts, referred to as Tweets
Follow	Following is a user is subscribing to that user's updates and Tweets
Followers	Followers are the users that
Re-Tweet	One user sharing another account's Tweets to their followers
Username / Screenname	The name a user chooses to use on the platform; stylized with the '@' character. Example *@UofCincy*
Timeline, Social Media Feed	The default view of Twitter for a user, which displays Tweets in a mostly chronological order
Hashtag	A way for a user to tag their content and make their tweets searchable; stylized with the '#' character. Example #GoBearcats

Appendix B

Username(@)	BotCheck.me	Botometer	BotSentinel	Consistent Results?
@realDonaldTrump	Bot = No Polarize = No	Bot = No	Rating = Normal	No
@BarackObama	Bot = No Polarize = No	Bot = No	Rating = Normal	Yes
@Nike	Bot = No Polarize = No	Bot = No	Rating = Normal	Yes
@AP	Bot = No Polarize=75%+	Bot = No	Rating = Normal	Yes
@TeaPainUSA	Bot = No Polarize = Yes	Bot = No	Rating = Normal	Yes
@BreitbartNews	Bot = No Polarize = No	Bot = No	Rating = Normal + Untrustworthy	No
@TheMAGANewsNet	Bot = Possibly Polarize = N/A	Bot = Yes	Rating = Normal	No
@NoisyInfamous	Bot = Yes Polarize = Yes	Bot = 50%	Rating = Alarming	Yes
@WhiteIsTheFury	Bot = Yes Polarize = Yes	Bot = No	Rating = Normal	No
@KeyToGenocide	Bot = No Polarize = No	Bot = Yes	Rating = Alarming	No
@JoshCantSwim	Bot = No Polarize = Yes	Bot = No	Rating = Normal	Yes
@_SJPeace_	Bot = No Polarize = No	Bot = No	Rating = Normal	Yes
@PolishPatriotTM	Not = No Polarize = Yes	Bot = Yes	Rating = Moderate	No
@Vote4Women2020	Bot = 50% Polarize = Yes	Bot = No	Rating = Moderate	No