Dissertation

Understanding the behaviour and influence of automated social agents

Syed Zafar ul Hussan Gilani

Selwyn College

University of Cambridge

Computer Laboratory

Email: Zafar.Gilani@cl.cam.ac.uk

Principal investigator: Prof. Jon Crowcroft

This dissertation is submitted on 24/8/2018 as a requirement for the degree of Doctor of Philosophy

Abstract

Online social networks (OSNs) have seen a remarkable rise in the presence of automated social agents, or *social bots*. Social bots are the new computing viral, that are surreptitious and clever. What facilitates the creation of social agents is the massive human user-base and business-supportive operating model of social networks. These automated agents are injected by agencies, brands, individuals, and corporations to serve their work and purpose; utilising them for news and emergency communication, marketing, social activism, political campaigning, and even spam and spreading malicious content. Their influence was recently substantiated by coordinated social hacking and computational political propaganda. The thesis of my dissertation argues that automated agents exercise a profound impact on OSNs that transforms into an array of influence on our society and systems. However, latent or veiled, these agents can be successfully detected through measurement, feature extraction and finely tuned supervised learning models. The various types of automated agents can be further unravelled through unsupervised machine learning and natural language processing, to formally inform the populace of their existence and impact.

Declaration

I declare that this Dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except where specified in the text. All of the technical work in this dissertation such as code, tests, deployment, experiments, and results have been completed by myself. All of the writing work done in collaboration has been completed by myself as the lead, while collaborating authors only guiding, rewriting or reviewing my text along the way. This report is not substantially the same as any that I have submitted for a degree or diploma or other qualification at any other university. This report does not exceed the prescribed limit of 60,000 words[1].

Ethical considerations of my research. All of the datasets that we collected, stored and consumed strictly adheres to the practices and guidelines of the Computer Laboratory at the University of Cambridge. We followed the ethical considerations and procedures outlined by the University of Cambridge Institutional Review Board (IRB) for all mentioned annotation tasks. Datasets collected through the Twitter Streaming API also follows the Twitter policy[2]. In order to protect individual privacy we follow Twitter's data usage policy: all data kept in an encrypted storage, not redistributed, no personal or sensitive information is used, and we only analyse aggregated statistics. Moreover, the results (classification, categorisation, tagging) are only indicative and informative, not disruptive or decisive. The Twitter bot deployed for studying Web bots was approved by the IRB. The bot was well within the ethical boundaries, since it was non-invasive and non-engaging.

[1] `ps2ascii thesis.pdf | wc -w` calculates that this document is approx. 47,069 words.
[2] https://dev.twitter.com/overview/terms/agreement-and-policy

Informing the user of our research. Our shortener service tnyURL.uk (and later renewed to tnyURL.co.uk) homepage notifies the users about the purpose of our research. We inform them that we are collecting the data triggered by their activity on our tweets or URLs, but while maintaining strict anonymity and privacy so we or anyone else cannot discern the identity of the Twitter user who clicked on our tweet or URL. Furthermore, we only collect information such as timestamp and User Agent string of the web browser a Twitter user is utilising to access our tweet or URL. Informed user consent was not required as the Twitter bot we deployed was non-invasive, neither engaged in direct communication with other Twitter users nor tried to identify individual users by any means.

Funding and fellowship. This work was partially funded by the Marie Curie ITN METRICS research grant EC607728 and the EPSRC Global Challenges Research Fund research grant EP/R512783/1.

Syed Zafar ul Hussan Gilani
21/8/2018

Acknowledgements

There is a long list of people I would like to thank. I will begin with Jon Crowcroft (my principal investigator for his support and encouragement), Arjuna Sathiaseelan (for his support during the first year of my PhD), Fahad Satti (for his assistance in making possible the *elusive* human annotation task for binary classification), Jatinder Singh (his priceless and kind support for actively helping me in going through the painstaking process of securing additional funding for 6 months, that enabled me to finish off my doctoral work and dissertation).

As it can be appreciated computing research is hardly ever a solitary exercise. I have thus been fortunate to have the helping hands of Reza Farahbakhsh[3] for Chapter 4 and § 6.5; Gareth Tyson[4] for Chapters 4 and 6 (under submission); Ekaterina Kochmar[5] for Chapter 5 and initial bot detection algorithm; and Mario Almeida[6] for § 3.3 and initial bot development. Perhaps the most common contribution of my colleagues has been text and reviews. Despite their priceless contributions, I have led the work and have been the first author in all of the publications related to the social bot work. The list of my personal contributions include, but are not limited to: (*i*) research questions and drafts, (*ii*) framework design and conceptualisation, (*iii*) writing and implementation of software, code and tools, (*iv*) system deployment, maintenance and troubleshooting, (*v*) analysis, results, figures and descriptive text, and (*vi*) paper/chapter drafting, compiling and submitting.

I would like to thank Andrew Moore and Cecilia Mascolo for their helpful comments during first and second year reports. I would also like to thank Andrew Rice and Nishanth Sastry for their all-important feedback during the viva

[3]Dr. R. Farahbakhsh, Institut Mines Telecom Paris – `reza.farahbakhsh@it-sudparis.eu`
[4]Dr. G. Tyson, Queen Mary University of London – `g.tyson@qmul.ac.uk`
[5]Dr. E. Kochmar, University of Cambridge – `ekaterina.kochmar@cl.cam.ac.uk`
[6]M. Almeida, UPC Barcelona – `mario.almeida@est.fib.upc.edu`

– it included some of the best suggestions I received during my writing and corrections. The feedback helped me improve the arguments and increase the rigour of findings through additional considerations.

And finally, my family - father, mother, my two sisters, my niece, and my wife. Without them nothing would have been possible. My father, who taught me that true pride is in education, critical thinking and reasoning, sociopolitical narrative and discourse, arts and culture, and not in money or materialism. My mother, who showed me what sacrifice, encouragement and unconditional love is, to learn to soak all the sorrows and emanate calmness in commotion. My two sisters, who took pride in calling me an elder brother, and being my closest friends when I needed them. My niece, who is the greatest pleasure of our lives, a simple look at her makes me so happy, she is the loveliest kid I know and no less than my own daughter. And finally, my wife, the true meaning of a big heart, a wonderful partner, gentle and kind, immensely loving, such wonderful personality, such wonderful sense of humour, my closest friend to discuss my biggest fears and my greatest failures, with whom I shared whatever I had been through, with whom I share my life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Motivation

"To err is human, but to really foul things up you need a computer" are the famous words of Paul R. Ehrlich. Biologist by training, he is best known for his warnings about consequential changes to population, food, computers, *etc.* And some of these warnings are not entirely ill-founded. One could argue existential threats often have humble beginnings, nurtured by the goodwill of scientific discovery and invention to achieve a better and sustainable human condition.

Humankind, social and political in nature, has adapted to the environment and created technology to vanquish problems that arise from limited physical capabilities of humans, such as: speed, efficiency (we need to eat and sleep to rejuvenate), availability and consistency. The age of automation brought mechanical robots and later software robots, that were designed to augment physical capabilities of humans, as well as process vast volume of transactions to deliver products and services to customers, and process a large array of datasets for informative analytics and internal audits.

Software robots, or *bots*, were software adaptation of mechanical robots. There could be a many (probably uncountable) types of software robots, such as system daemons, computer viruses, Web crawlers, indexers, content curators, malicious spiders, virtual assistants and even chat bots. Automated social agents, or *social bots* (as we better know them), are one such extension of this technology. A social bot is a type of automated software robot that controls and operates a social media account. Unlike, a regular automated software robot, a social bot may likely exist surreptitiously on a social network while maintaining a profile and activities that are akin to a real person. While it is a common belief that

most social bots (and even software robots) are malicious, not all bots are created equal. Domain experts would even argue that social bots are unethical – especially if they have a latent existence.

The existence of social bots depends on the social network platform and whether the platform allows automated actions or not. Social bots may have started as friendly and benign hobbies, but were quickly adapted as digital *workers* to serve their human *masters* in a number of different settings on social network platforms. These include but are not limited to news, emergency communication, political campaigning, social activism, targeted social marketing, spamming, *etc.* *Bots*, one may argue, have quickly become a phenomenon of their own atop the social network phenomena.

This brings us to the potential usage of social bots for sociopolitical campaigning and spreading fake news. While online social networks (OSNs) were first effectively used by Barack Obama during the 2008 U.S. presidential election to reach out to masses and propagate his campaign, it is speculated bots first truly made an impact through proliferation during the UK's EU Referendum – since then better known as Brexit (see § 2.1.9). The trend has not reversed since then. It has been found and is now a subject of an FBI inquiry[1] pending thorough investigation and subsequent decision that the 2016 U.S. presidential election was marred by Trump-Russia collusion[2] throughout the campaign. The resources used during the campaign involved online social media, targeted marketing services and bots. Bots have also been found to infiltrate the 2017 French presidential election and the Venezuelan politics (see § 2.1.9). The affect, as the reader can well imagine, is both profound and unprecedented.

Realising the importance of investigating social bots, part of this work develops a generically designed modular platform that is built through measurement and research. The platform delivers the basis for measuring and characterising bots through exploratory data science, detecting bots through supervised machine learning, and categorising bots to discern types using unsupervised machine learning, as well as collecting and experimenting with data from the Web that is otherwise not available from Twitter.

[1]FBI inquiry into 2016 U.S. presidential election (last accessed 16 June 2018) – `https://www.nytimes.com/2017/12/30/us/politics/how-fbi-russia-investigation-began-george-papadopoulos.html`

[2]Trump-Russia inquiry indictment (last accessed 16 June 2018) – `http://www.bbc.co.uk/news/world-us-canada-43095881`

Terms and definitions: For the purposes of research carried out in this dissertation I set forth a few definitions of terms I will be using throughout this dissertation. Conceptually, a 'bot' is an entity that simulates human activity through imitation of actions and behaviour. Operationally, this translates to a 'bot' being in control of any social media account that *consistently* involves automation during the observed period, *e.g.* use of the Twitter API or other third party tools, performing actions such as automated likes, tweets, retweets, *etc.* For the purposes of this dissertation the following four terms mean the same thing: *bots*, *social bots*, *automated agents*, and *automated social agents*. A *tweet* is defined as an original status and not a retweet. A *retweet* is a tweet in which the text is prefixed with 'RT'. A *status* is either a tweet or a retweet, and therefore total statuses are the sum of tweets and retweets. *Content* on Twitter is limited to whatever is contained within a tweet: text, URL, image, and video. A *favourite* or *like* is the activity of liking a status. A *mention* is the act of quoting a Twitter handle of a Twitter user in a status. We define a *bot type* or *category* as a grouping of similar accounts together that exhibit similar behavioural characteristics (features), tweeting about similar topics, and exhibiting similar sentiments.

Thesis statement: Automated social agents exercise an influence (social and otherwise) upon human online social populace. Surreptitious or otherwise, these agents can be successfully detected through carefully executed measurement, feature extraction and finely tuned supervised machine learning models. We can further decompose the social bot population into various types or categories using unsupervised machine learning methods to formally inform the populace of their existence and impact.

Goals and objectives: The goals and objectives of research encompassed in this dissertation are manifold and require concrete steps that are measurable and time-bounded. To investigate automated entities in online social network, a flexible and modular framework is required that utilises methods and techniques from data science and machine learning. This requires understanding the functionality of the framework such that it is able to continuously collect large datasets and process these for analyses. The framework should also be generic within the bounds of the domain, enabling researchers to explore a wide range of domain-specific problems. In addition to the design of the framework, a methodology for creating a ground-truth dataset will also be required (for training machine learn-

ing algorithms). A thorough study of behavioural and network properties would be required to differentiate bots from humans. This will be done by extracting principal features that are most representative of bots.

The second goal will be to use the outcomes of the first goal to extend the framework by implementing an automated supervised learning method that reliably classifies bots and humans. This will also require evaluating the bot classifier against current state of the art using the collected and manually annotated datasets.

The third goal is to use the outcomes of the first and second goals to extend the framework further by implementing an automated bot typification tool using an unsupervised learning method which categorises bots into algorithmically learned categories. A classified bot dataset will be created using the work fulfilled in the second goal. In addition to this, tools will be needed for topic modelling and sentiment analysis to analyse content and sentiment shared by various bot categories.

The final goal of this dissertation will be to demonstrate generalisability of the framework. Firstly, the framework will be extended to study influence of 'Web' bots on social content, to explore bot influence beyond the social networks and onto the Web. Secondly, the framework will be applied to study a problem statement analysing human influence on OSNs.

Contributions: Bots widely exist in OSNs. They contribute a significant amount of activities, both consume and produce content, and even interact with human users. As the analysis on human behaviours is crucial to understanding OSNs, a thorough research on bot demography is equally important. This dissertation contributes the following: (*i*) definition of what is a 'bot' (this chapter), (*ii*) a thorough comparative literature survey and state-of-the-art in this domain (Chapter 2), (*iii*) creating a ground-truth dataset using a manual or *human* annotation task (Chapter 3 and Appendix A), (*iv*) performing a detailed characterisation of bots and humans to extract most representative features and behavioural properties to clearly differentiate automated social agents from humans (Chapter 4), (*v*) using these characterisations I implement a detection algorithm to automatically discern automated social agents from humans (Chapter 5), (*vi*) building bot taxonomies (Chapter 6), (*vii*) perform bot typification to explore and distinguish various bot categories (Chapter 6), (*viii*) exploring bots on the Web (Chapter 6), and (*ix*) contributing characterisation, detection and categori-

sation datasets[3] to the research community.

[3] ***Stweeler*** processed datasets – `http://www.cl.cam.ac.uk/%7Eszuhg2/data.html`

Chapter 2

Background

The World Wide Web (WWW) has seen a massive growth in variety and usage of OSNs. Twitter, with its 313 million active monthly users, is one of the biggest OSNs in the world. The rising population of users on Twitter and its open nature has made it an ideal platform for various kinds of opportunistic pursuits, such as from distributing content (news or spam) to promoting businesses and enterprises (ads, marketing). These opportunistic pursuits are exploited through automated social agents, or *social bots*. *Bots* are automated programs that operate social media accounts via automated control commands and exist in vast quantities in online social networks.

Estimates suggest 51.8% of all Web traffic is thought to be generated by bots[1]. A media analytics company found that 54% of the online ads shown in 2012 and 2013 were viewed by bots rather than humans[2]. In 2014 Twitter itself reported that 13.5 million (5% of the total at the time) of its accounts were either fake, fraudulent or spam[3]. My own work in this dissertation finds that slightly less than half (43.13%) of the Twitter population in the datasets collected are operated by bots or some sort of automation.

Bots are created for a number of purposes, *e.g.* news, marketing, link farming,[4] political infiltration (§ 2.1.9), spamming and spreading malicious content.

[1]Bot traffic report 2016 (last accessed 16 June 2018) – `https://www.incapsula.com/blog/bot-traffic-report-2016.html`

[2]Fake ads traffic (last accessed 16 June 2018) – `http://observer.com/2014/01/fake-traffic-means-real-paydays/`

[3]Twitter's 2014 Q2 SEC filing (last accessed 16 June 2018) – `https://www.adweek.com/digital/twitter-says-over-13-million-accounts-may-be-bots-and-fakes-159458/`

[4]Link farming (last accessed 16 June 2018) – `http://observer.com/2014/01/`

The rise of bots (particularly spambots) on Twitter is substantiated by a number of studies (see § 2.1.4, § 2.1.7–2.1.8) and articles[5]. Despite the phenomenal rise, not all bots are created exclusively for malevolent purposes (*i.e.* spam). There are bots which are benign and benevolent, such as news and emergency communication, art and discovery[6], content aggregation, fun and humour[7], marketing and business promotion, and social activism [71].

This massive rise in bot population on Twitter is not new – bots have existed on Twitter since its inception. The existence of bots on Twitter is owed to a number of reasons: soft inspection during registration (an email address, a CAPTCHA recognition and a phone number are the only requirements), but mostly due to the Twitter API that lets programmers automate actions on Twitter. Studying the bot phenomenon is important in order to understand dynamics of: (*i*) influence on social systems exercised through user (human or bot) behaviour, and (*ii*) human-bot interaction from sociological perspective.

I focus on bots in Twitter primarily because of three reasons: Twitter content is mostly public[8], it allows automation through its APIs[9], and studies below indicate a substantial presence of automated programs on Twitter. Compared to other social networks, such as Facebook or Instagram, Twitter is an *information* social network that exposes most of its content publicly by default. Facebook, therefore, can be thought of as a *pure* social network since it keeps everything enclosed (or private) for a user unless a user chooses to make public a certain piece of content, or a user accepts a 'friend' request from another user (in which case the befriended user can view most of the content). Instagram, from a technical point of view sits between Facebook and Twitter. While Instagram has an API that can be used by third-party apps (for business purposes), it does not allow the API to be used for automation, as directed by its terms of use[10] and platform

---

`fake-traffic-means-real-paydays/`

[5]Bots in press and blogs – `http://www.cl.cam.ac.uk/%7Eszuhg2/docs/papers/bots-discussions.txt`

[6]Art and discovery bots (last accessed 16 June 2018) – `https://qz.com/572763/the-best-twitter-bots-of-2015/`

[7]Fun and humour bots (last accessed 16 June 2018) – `https://qz.com/279139/the-17-best-bots-on-twitter/`

[8]Twitter Public APIs (last accessed 16 June 2018) – `https://developer.twitter.com/en/docs`

[9]Twitter Developer Agreement (last accessed 16 June 2018) – `https://developer.twitter.com/en/developer-terms/agreement-and-policy`

[10]Instagram Terms of Use (last accessed 16 June 2018) – `https://help.instagram.com/`

policy[11]. Secondly, neither Facebook nor Instagram expose any public data via an API, thus leaving data scraping via Web crawlers (that require the input of specific keywords, hashtags, *etc*) as the only option. Facebook and Instagram have maintained an extremely strict policy towards bots and suspend accounts instantly that are found to have unusual activity. Therefore, bots on both the platforms are extremely short-lived (a few hours on average).

In this chapter I will provide a background literature survey of the current state of the art, and shortcomings that I contribute to.

2.1 Literature Survey

Research on social bots has generally focussed on a number of aspects, ranging from user behaviour and social media infiltration to social influence and bot detection schemes. Relevant work can be categorised into a total of thirteen domains discussed below.

2.1.1 Web bots

Though different in nature and purpose to social bots, Web bots mostly serve the needs of search engines and archives by visiting and recording a massive amount of webpages. Though most commonly referred to as 'bots' since the beginning [66], these were also known as 'indexers', 'crawlers', 'worms' or even 'spiders' These bots do not interact directly with humans via a social platform. Despite this, Web bots can create an indirect impact on information being displayed on social platforms to human users. For instance, given the open nature of Twitter, Web bots can contribute to traffic and activity generated that could consequently impact the popularity of content. Hardly any research explores impact of Web bots on social platforms. I explore and measure this impact in Chapter 6.

2.1.2 Chatbots

Chatbots are as old as computers, *e.g.* ELIZA [89], and interact with humans through an interface medium which is usually text. The idea behind a chatbot,

478745558852511

[11]Instagram Platform Policy (last accessed 16 June 2018) – `https://www.instagram.com/about/legal/terms/api/`

or sometimes referred to as 'chatterbot', is to employ natural language processing to process human user's input text to produce a dialogue response [23]. Recent examples include conversational virtual assistants such as Apple Siri, Amazon Alexa, Microsoft Cortana or Google Now. Chatbots have become widely common as conversational virtual assistants and for service provisioning and task management in many instant communication apps (*e.g.* Skype, Slack, Telegram) [29]. Business and corporate sector have employed chatbots to improve experience of their clients and customers.

These types of bots are extremely relevant when it comes to social bot research (especially those that are disguised) because these bots not only interact directly with humans, but can also be used in political scenarios with significant impact. More on this in § 2.1.9.

2.1.3 Game bots

Game bots are usually a type of intelligent software that are designed to play a computer game. These could either be *static* – designed to follow waypoints for each level or terrain map in the game, or *dynamic* – designed to learn the levels or terrain maps by leveraging machine learning. Game bots could be designed for a variety of games, including massively multiplayer online role-playing games (MMORPGs). These bots automate gameplay by mainly imitating *perceptions* and *reactions* in human gameplay traces [82].

Game bots can cause problems for publishers as well as human players. Concerns have been identified that link to collapse of game balance and player dissatisfaction that often leads to discontinuation. To mitigate this, researchers have proposed keystroke detection, game traffic and CAPTCHA tests. Following player dissatisfaction with disrupted gaming experience, many researchers have independently devised similar techniques for non-interactive game bot detection approach.

Chen *et al.* in [17] proposed a manifold learning approach to identify game bots. The method uses actual gameplay data logs to learn the differences between the trajectories of humans and bots. The researchers found that despite bots simulating humans, there are certain types of human behaviour that is not easy to imitate. They used more than 200 dimensions with 3D to 2D dimensional mapping from actual gameplay data logs of Quake 2 to detect and evaluate bot

detection. They found that the scheme achieves an accuracy of up to 98% on a trace of 700 seconds.

In [79], Thawonmas *et al.* propose similar technique using discrepancies between humans and bots in action frequencies and action types in gameplay logs. The researchers propose classifying characters as bots if frequencies for particular actions are higher than human players, and adjusting the classification based on action types. They evaluated their technique on Cabal Online gameplay logs and found that the accuracy is 38–60% for 15–60 mins of detection time.

Similarly, in [35] Gianvecchio *et al.* use human observation proofs to passively monitor input actions that are harder for bots to imitate. The researchers use World of Warcraft gameplay logs to first characterise human and bot behaviour during gameplay. They next develop a neural network that uses human observation proof system for analysing input actions. Using a gameplay log of more than 95 hours, their system is able to identify game bots within 40 seconds.

Although, game bots have a lower relevance to the social bot phenomena explored in this dissertation, I do see a possibility of the two aligning in future when social bots become intelligent enough to pass through game-oriented bot prevention techniques on many Web platforms.

2.1.4 Sybil and fake accounts

Social bots are often considered as an adversary in information security domain. Security experts sometimes use the term 'sybil' to represent these bots that use fabricated identities for a number of purposes. These include spamming, manipulating discussions, spreading malicious links and advertisements, and exploiting personal information extracted from the network.

Cao *et al.* in [15] introduce a tool called SybilRank that uses the social graph to detect 'sybil' or fake accounts. The tool has been evaluated by the researchers on a test dataset from Tuenti – the largest OSN in Spain. SybilRank found that 90% of 200,000 designated suspicious accounts by Tuenti were actually fake. In comparison the manual user-reported system only achieved 5% accuracy.

While 'sybil' or fake accounts are mostly interested in causing users to click a link, astroturf accounts want to create a false view of consensus about a topic or message. Legitimate users may inadvertently help spread the message by being deceived. Therefore, one of the biggest unsolved problems for social networks is

to detect inorganic campaigns from organic ones. Ratkiewicz *et al.* in [69, 70] create a tool called Truthy to bridge the gap by detect astroturfing, smear and misinformation campaigns. Truthy helps analyse meme diffusion through mining and classification of tweet streams of events being posted on Twitter.

2.1.5 Useful social bots

Twitter is full of useful bots that exist in many forms. One of these bots is the @dscovr_epic bot which is an unofficial bot created by Russ Garrett that posts pictures from Earth Polychromatic Imaging Camera (EPIC) on the NASA DSCOVR spacecraft[12]. The bot brings (to its 15,000 Twitter followers) rare Earth and Moon images captured during different time periods.

Two other useful unofficial bots post pictures of exhibits from Museum of Design Zurich (@GestaltungBot) and Metropolitan Museum of Art New York (@MuseumBot). These bots help bring history and knowledge by sharing museum exhibits to their 9,000 Twitter followers.

Twitter bots are not only limited to posting pictures from other sources. @DearAssistant is a Twitter bot that answers questions just like Apple's Siri or Google's Now would. Created by Amit Agarwal, a Google Apps script developer, the bot uses WolframAlpha (a computational knowledge engine) to post replies to questions that are asked of it.

2.1.6 User behaviour

Investigation of user behaviour can reveal traces of activity that can prove immensely valuable in characterising differences between bots and humans. Features that represent frequency of activity, nature of activity, typical behaviour, and how it is posted online are all important in knowing the true nature of an entity.

In [55] authors used follower-to-following ratio on Twitter to classify the users into broadcasters (having significantly more followers than following), acquaintances (congruent follower-to-following ratio), and miscreants and evangelists. In a related work [85] authors use principal component analysis to identify deviations in anomalous user behaviour from normal user behaviour. The authors then apply unsupervised anomaly detection technique to address the problem of de-

[12]NASA DSCOVR – `https://epic.gsfc.nasa.gov/`

tecting subversive promotion techniques via fake and compromised accounts, and collusion networks or bot farms on Facebook. Both of these works perform user classification to detect subversive and attacker strategies in online social settings, but do not focus on automation.

In [12], Boyd *et al.* inspected user behaviour by examining retweets, focussing on *how people tweet*, as well as *why and what people retweet*. The authors found that participants retweet using different styles, and for diverse reasons (*e.g.* for others or for social action). Closely related, in [90] Wu *et al.* study *marionette* users on Weibo[13], created or employed by *puppeteers* or human *masters* either manually or through programs. These marionette users are used to perform specific tasks to earn financial rewards, such as follow certain users or re-share certain posts to increase popularity and visibility. Similar to Twitter, artificially increasing followers and retweet counts leads users to incorrect perception of popularity of posts and search for topical experts. The authors profile such users through analysis of users' posting behaviours and social interactions. The authors apply a probabilistic classification model that uses influence received by a user from its neighbours (such as through likes and re-sharing) to classify a user as either normal or marionette. Their experiments reveal an accuracy of 0.892 for a labelled dataset of 2,000 users.

These are relevant to my own work, as I also study retweets and tweeting patterns through tweet frequency and tweet-retweet distribution. In contrast, my work provides further insights on important differences and striking similarities between bots and humans in terms of *retweet patterns*, *account lifetime*, *account reciprocity*, *content creation*, *content popularity*, *content consumption*, *content propagation* and *entity interaction*. In addition to studying these features above, I also study *sources* or endpoint apps that are used to produce activity on Twitter by humans and bots. These *sources* reveal important information that can be used to differentiate between human activity and bot activity. This forms the basis for a reliable bot detection algorithm.

2.1.7 Social botnets

Botnets are generally considered a threat to cyberspace. Due to the small world properties and reachability advantage of many of the social networks, *bot masters*

[13]Weibo is a Chinese microblogging service, similar to Twitter.

operate their botnets by using social networks as their command-and-control (C&C) centres (also known as soft-infrastructures). Social networks, such as Twitter, give bot masters the ability to control individual bots through API calls. Typically, C&C enables stimulation of botnets [95] and allows quick evolution of strategies to target people and adjust according to the countermeasures.

Botnets and campaigns on social networks, Twitter in particular, are a common phenomena, explaining why Twitter has a dedicated anti-spam team [14] to watch over and mitigate the problem. A number of botnets and campaigns have been discovered, among them are the Naz botnet [65], the KOOBFACE botnet [5], Facebook spam botnet [32], Twitter spam botnets and spam campaigns [42, 81, 19], Twitter cyber criminal botnets [92] and Twitter link farming campaigns [33].

Measuring C&C strategies is important to understand strengths and weaknesses of social botnets. In [54] Kartaltepe *et al.* characterise the social network-based botnet C&Cs. The authors explore application-centric approach of detection and subsequent countermeasures, as compared to network-centric and host-centric approaches. The network-centric approach requires network traffic information including IP addresses, server names, packet content, and the host-centric approach performs an in-depth inspection of the software stack to find malicious processes that use data from network as parameters in system calls. The authors find that the application-centric approach is more effective than the above two approaches while requiring less data and not compromising system performance. The application-centric approach requires a simple detection mechanism that uses a Web service to classify text (from the social network content updater) for determining if a text is suspicious.

2.1.8 Social media infiltration experiments

Some of the research mentioned in this subsection sits at the boundary of what is considered ethical. However, I include these works because of the knowledge they provide.

Researchers have detected and studied as well as created their own social infiltration experiments (or 'honeypots'), that interact with other social network

[14]Twitter anti-spam team – `https://help.twitter.com/en/safety-and-security/report-spam`

users, in the hopes to understand how these honeypots operate.

Similarly, spam now widespread over email has started spoiling the social network experience. In [76] Stringhini *et al.* create 'honey-profiles' on three social networks including Facebook and Twitter, to log traffic and activity. This activity is in the form of friend requests, messages and invitations received from other users of the network. They find that 173 of 3,831 (4.52%) friend requests on Facebook and 361 of 397 (90.93%) follows on Twitter are from spammers. The main reason for such a big difference between Facebook and Twitter is Twitter API's that allows people to interact with the platform via computer programs. Rather, such instances of automated spam accounts on Facebook mainly exist because of the major technical challenge [88] of accurately automating classification of inauthentic or spam accounts. With over 2 billion active monthly users, taking the manual route of identifying such accounts is out of question. Some believe that alleviating the 'bot problem' is as simple as enforcing strict real names[15], thus also triggering the debate of anonymity and privacy on the Internet [13, 45]. Though, sadly this is not true given the existence of legitimate as well as stealthier intelligent bots imitating humans. In fact, the only effective way might be to disallow API interaction with these platforms, *i.e.* making private all public APIs that allow a computer program to execute callable actions.

The researchers in [76] also categorised bots by behaviour in these spam requests. These included: (*i*) spam bots that display content on their profile (least effective strategy for spreading spam), (*ii*) bots that post messages on their own profile (thus only reaching people who befriend or follow them), (*iii*) bots that post messages directly on profiles of people in their friend or follow list (most effective way of spamming as its visible to the friends of that user's profile), and (*iv*) bots that send direct private messages to people in their friend or follow list (only visible to the recipient). The authors were also able to distinguish greedy bots – those that include spam content in every message, and stealthy bots – those that include spam content once in a while. In Chapter 6 I typify bots based on the activity type and frequency in order to annotate latent categories of bots that exist on the Twitter platform.

In [10, 11], Boshmaf *et al.* evaluate vulnerability of Facebook against large-scale infiltration by deploying a social bot network of 102 profiles. They found

[15]How to fix Facebook (last accessed 30 June 2018) – `https://www.nytimes.com/2017/10/31/technology/how-to-fix-facebook-we-asked-9-experts.html`

that 86% of bots infiltrated up to 50 user profiles and 10% bots were able to infiltrate up to 80 user profiles. They also found that a successful infiltration reveals users' private information, and security defences are not sufficient to guard from a stealthy infiltrator. Similarly, in [31] Freitas *et al.* manually evaluate infiltration strategies on Twitter using 120 social bot profiles. They use three metrics to quantify the infiltration of social bots: followers, popularity score, and message-based interaction (other users favouriting, retweeting, replying or mentioning the bot). They found that bots can successfully evade Twitter defences (only 38 out of their 120 bots got suspended over the course of 30 days), and can successfully infiltrate Twitter (20% of the bots had more than a 100 followers). They conclude that infiltration is indeed successful, can affect influence/popularity scores and possibly impact the social network as bots can manipulate trending topics during political and social campaigns.

In [94] Zhang *et al.* create a social botnet for spam distribution by buying 1,000 accounts. The researchers carry out different experiments by designing botnets that simultaneously post tweets, or by creating a 10-ary tree of depth 2 where root bot tweets a post and its descendants retweet at random intervals. The result of these experiments reveal that complete botnets tweeting simultaneously get suspended within 6 hours, whereas only the root bot gets suspended within 6 hours but the descendant bots remain unsuspended. Repeating the second experiment by reallocating a root bot and shuffling the descendants produces the same results, *i.e.* only the root bot gets suspended. The researchers also investigate digital influence of accounts by using third-party Web services such as Klout, Kred, and Retweet Rank, with interesting results. They find that the number of followers impacts Klout influence score the least, whereas Kred and Retweet Rank are most affected. This means that while botnets can increase their Kred and Retweet Rank scores, they are unable to increase Klout influence scores by acquiring fake followers or by purposefully following each other. All three scores are impacted in terms of retweeting since retweets depict influence of an account in the local neighbourhood. However, this makes the influence scores vulnerable to botnets collaborating to retweet each other or any other user. Fake following and purposeful retweeting has been widely studied in political scenarios (see § 2.1.9). Similarly, all three scores are impacted in terms of mentioning which could prove to be exploitable by botnets through collaborative mentioning of each other or any other user.

I do not perform any infiltration experiments, as this is beyond the scope of my research, as well as borderline on ethical grounds. In fact, Facebook has previously faced public backlash because they systematically manipulated user environments to test user reactions [43]. Any such experiment requires obtaining informed user consent, without which it is deemed unethical. However, I use some of the understandings derived from the aforementioned research to study Web bot traffic on Twitter. Studying this Web bot traffic is important to understand as these bots could be *infiltrating* the Twitter platform.

2.1.9 Bots in politics

Bots have been used in political scenarios going as far back as 2012. Top presidential candidates of Mexico started using armies of bots[16] during the 2012 Mexico presidential election to either target opponents via defamation campaigns, or benefit themselves. These campaigns are labelled 'hashtag mischief' by researchers, which are perpetrated by bots with the intention to make these hashtags trend and eventually become a part of Twitter's trending topics.

Another such campaign that year was observed in Russia. Social activists took to Twitter to discuss the 2012 Russian presidential election. Thomas *et al.* in [80] found that a coordinated bot campaign was used to post spam hashtags in order to inundate the political discussion. The bot campaign used 25,860 fraudulent Twitter accounts to inject 440,793 tweets into legitimate conversations. Staggeringly, researchers also found that these fraudulent accounts belong to a larger pool of a million fraudulent accounts, kept dormant during the campaign possibly for future use. Furthermore, the campaign used machines across the globe, 39% of which appeared in IP blacklists, therefore suggesting usage of compromised hosts. Even more staggeringly, 56% of users were found to be located in Russia, whereas only 1% of spam accounts were located within Russia.

Later in 2012 during the U.S. presidential election Mitt Romney mysteriously acquired 116,922 more followers[17] (17% more increase) on 21 July 2012. Researchers uncovered that about 23% of these followers had never tweeted, while

[16]Mexico presidential election campaign 2012 (last accessed 15 June 2018) – `https://www.technologyreview.com/s/428286/twitter-mischief-plagues-mexicos-election/`

[17]Mitt Romney acquires 116K followers in 1 day (last accessed 15 June 2018) – `https://www.cnet.com/news/mitt-romney-suspiciously-gets-116k-twitter-followers-in-one-day/`

a tenth of these followers had been suspended by the time news was published on 6 August 2012. Most astonishingly, a quarter of these followers were less than 3 weeks old, while 80% of these were less than 3 months old. It is believed that followers came from Twitter follower services that sell follower accounts, likes, tweets and retweets.

In [30] Forelle *et al.* uncovered the strategic role of sociopolitical bots. The researchers analysed activity patterns (follow, tweet, retweet) by examining Twitter feeds of prominent Venezuelan politicians from 2015. They find that 10% of all retweets come from bots, where most of bots are used by the Venezuelan opposition. They also find that bots are mostly pretending to be the politicians, leaders, political entities, and government rather than citizens.

By 2016 political bot phenomenon reached its height, taking its shape as masqueraded campaigns during U.K.-E.U. referendum and 2016 U.S. presidential election. Researchers in [48] analyse Twitter data collected for relevant positive, negative and neutral hashtags between 5 June 2016 and 12 June 2016. By collecting more than 1.5 million tweets from more than 313K unique accounts the researchers were able to quantify strategic role of bots from both campaigns Remain (popularly called 'Bremain') and Leave (popularly called 'Brexit'). Firstly, the hashtags associated with 'Leave' campaign dominated hashtags from 'Remain' campaign by as much as 3-6× (341,839 for #voteleave *vs.* 110,653 for #strongerin, respectively). Secondly, different perspectives utilised different levels of automation. For example, the most popular 'Remain' hashtag #strongerin accounted for only 14.6% (186,279) of the tweets out of which only 15.1% (28,075) were generated by automated sources. Whereas, the most popular 'Leave' hashtag, #brexit, accounted for 51.8% (662,745) of the tweets out of which 14.7% (97,431) were generated by automated sources. In fact, 5.7% (13,436) of neutral tweets (18.3% or 234,170 in total) were also generated by automated sources. Thirdly, less than 1% of 313,832 unique accounts generated one-third of the tweets.

Similarly, in [6] Bastos *et al.* uncovered a network of 13,493 Twitter bots that tweeted during the U.K.-E.U. referendum, but disappeared shortly after the U.K. voted for leaving the E.U. The researchers compare normal users with political bots in terms of tweeting behaviour, and retweet proportion and frequency to find strategies of bot usage. The authors made two important discoveries: (*i*) the ability of bots to rapidly generate short-lived retweet cascades containing

user-generated partisan news, and (*ii*) criteria-driven botnets organised to either replicate active users or replicate content posted by other bots.

This was quickly followed by the 2016 U.S. presidential election, where researchers discovered bots behind distortion campaigns in online discussions [8]. They found that 1 out of 5 tweets regarding the elections was posted by a bot, *i.e.* 4 million tweets by 400K bots in the month leading to the election. Twitter's interface does not specify the software platform of the tweet, thus making it difficult for humans to determine whether a tweet is posted by a bot or a human. This meant bots were being retweeted at the same rate as humans. The authors found the bots to be biased, *e.g.* pro-Trump bots were producing supportive and positive content, thus ensuring a false public perception of grassroot support for Trump. In fact, negative campaign by Clinton supporters against the opponent candidate Trump was so unsuccessful that it accrued a 50-50 split of positive and negative responses. Whereas, negative campaign by Trump supporters against Clinton accrued 15.92% more negative responses than positive. Using geo-analysis bots were found to exercise strong support in the Midwest and Southern states, especially Georgia. Whereas, humans were found to exercise influence in most populated states such as California, Texas, Florida, Illinois, New York and Massachusetts. The study also classified top five hashtags for both presidential candidates. It found that among the 7,112 Clinton supporters 590 (8.3%) were bots, whereas among 17,202 Trump supporters 1,867 (10.85%) were bots.

Unfortunately, the covert and unwarranted use of bots in political campaigns had by now become an unimpeded norm. During the 2017 French presidential election Ferrera *et al.* [27] investigated the #MacronLeaks disinformation campaign against the candidate Emmanuel Macron. By collecting 17 million tweets between 27 April 2017 and 7 May 2017 the author discovered 18,324 bots (18%) and 81,054 humans participating in the #MacronLeaks disinformation campaign. The author discovered that some bot accounts were originally created prior to the 2016 U.S. presidential election. These bot accounts first went dormant after November 2016 (upon completion of the U.S. presidential election) and were later recycled for use in #MacronLeaks disinformation campaign in the beginning of May 2017. This further raises the suspicion of existence of social botnet black markets.

2.1.10 Social influence of bots

In [3], authors use a bot on aNobii, a social networking site aimed at readers, to explore the *trust*, *popularity* and *influence* of bots. They show that gaining popularity does not require individualistic user features or actions, but rather simple social probing (*i.e.* bots following and sending messages to users randomly). The authors also found that an account can circumvent trust if it is popular (since popularity translates into influence). Similarly, in [26], Edwards *et al.* highlight a positive view on the existence of bots on social media by studying the differences in perceptions of the quality of communication for a human agent and a bot agent on Twitter. They find that Twitter bots can be viewed as credible, attractive, competent in communication, and interactive. Taking inspiration from this work, I extend exploration to the Twitter platform. However, instead of infiltrating a social network with *honeypot* bot(s), I study the characteristics of existing bots.

Closely related is [86], which develops models to identify users who are *susceptible* to social bots, *i.e.* likely to follow and interact with bots. The authors use a dataset from the Social Bot Challenge 2011[18], and make a number of interesting findings, *e.g.* that users who employ more negation words have a higher susceptibility level. Similarly, users with a higher temporal balance *i.e.* who tweet more often, and those who discuss morbid topics more often tend to have higher percentage of interaction with bots. In my work, I study the characteristics of existing bots in detail and argue that this provides far broader vantage into real bot activities. Hence, unlike studies that focus on the influence of individual bots (*e.g.* the Syrian Civil War [1]), I gain perspective on the wider spectrum of how bots and humans operate, and interact. Additionally, in Chapter 7 I devised a *non-infiltrating* honeypot experiment to study the impact of bots on content popularity.

Mitter *et al.* in [61] explore if social bots can be used to influence link creation between targeted human users. The authors use a dataset from Pacific Social Architecting Corporation 2011 [63] to launch bots for investigating the creation of new social links. The authors find that approximately 12% links are caused by bots mediating suggestions for connecting to target humans. In Chapter 4 I explored the degree of bot and human inter-connectedness and intra-connectedness

[18]I did not use this dataset as it was outdated: Twitter suspends unusual accounts, bots evolve, and so does the technology that drives these entities.

by exploring retweets and quotes, and replies and mentions.

2.1.11 Bot detection

The importance of bot detection on social media has recently gained momentum due to the rapid rise of bots. In [91], Yan *et al.* studied if an automated Turing test such as the CAPTCHA is sufficient to verify that an entity behind a computer is a human or an algorithm. The study concludes that CAPTCHA, apart from being inappropriate for some usability concerns, is insufficient to discern humans from bots. In a comprehensive work [18], Chu *et al.* distinguish and identify Twitter accounts operated by three entities: humans, cyborgs and bots. The authors make this classification by observing the differences among the three entities in terms of tweeting behaviour, tweet content and account properties. Using 1,000 training samples the authors devised a system that classified their subset of the Twitter population into 5:4:1 proportions for human:cyborg:bot, respectively. However, they neither provide an API for evaluation nor share datasets. Comparably, I find that approximately half (43.13% to be exact) of the user accounts in my Twitter datasets are operated by bots.

In another effort DARPA organised a Twitter bot challenge in 2016 [77] to detect influence bots – bots that illicitly shape topical discussions on Twitter to serve the purposes of their masters. DARPA provided 7,038 accounts as ground truth labels that they knew about to the six teams who participated. The report concludes that detection of evolving influence bots requires carefully designed workflow and machine learning does not always work.

Coincidentally, there has been a recent surge in research focused on automating content generation [75] that looks to have been produced by humans. Also, some techniques are focussed on discerning anomalous from normal, spam from non-spam, and fake from original, but they fail to distinguish (or compare) the types of users. I clearly distinguish between my task of agent classification and spam detection. Spam is usually subversive and malicious in nature [76], is often found to be high in volume and frequency, and contains URLs (that point to malicious websites) and spam words [7, 57]. However, as briefed earlier, automation is not exclusively employed for malevolent purposes. There could be many variants of automation due to the usage of APIs and third-party services, and it can often involve direct human intervention (Chapter 3–5). Also, there are

no guarantees that a successfully detected spam account is operated by an agent and not a human – it could be either. This forms a strong basis for detecting automation without any prior judgement.

However, as mentioned most of the techniques neither expose their datasets nor their tools, which makes evaluation tough. To the best of my knowledge there is only one freely available and useable research tool, BotOrNot[19] [22, 83], that detects bots on Twitter. The tool applies a Random Forests classifier and uses 1,000 features divided into six groups to classify accounts as 'bots' or 'humans'. The model is trained using a list of social bots identified in [58] and a dataset from the Twitter Search API of 200 most recent tweets of these bots and 100 most recent tweets mentioning these bots. This yields a dataset of 15K manually verified social bots and 16K human accounts. The authors report a ten-fold cross-validation score of 0.95 AUC.

Apart from using a Random Forests classifier and a more specific feature-set, I use raw historical data to cater for evolution of agents and stealthier agents. I use a dataset partitioned into four popularity bands representing Twitter population at a more granular level, as agents differ according to the popularity and purpose of their creation and presence. I also use 14 novel features from a set of total 22 attributes. Furthermore, I employ account categorisation in the preprocessed and partitioned datasets, and perform ablation tests to identify distinct group of features that are most effective for each popularity band (Chapter 4).

2.1.12 Bot detection avoidance techniques

Social bots created and intended for unapparent purposes, such as human mimicking, sociopolitical campaigning, distortion of online discussion, advertisements and spam, use a number of techniques. These techniques could include the use of any combination of intelligent content retweeting, variable tweeting frequency, manipulation of tweet source endpoint, automated text summarisation, automated text generation, and automated discourse response. Though, social bot detection avoidance has not been particularly studied, one can affiliate specific application of relevant technologies for bootstrapping stealthy social bots.

For instance, many social media integration management apps (*e.g.* Tweet-

[19] BotOrNot is now rebranded as Botometer, but I continue to use BotOrNot to refer to the said.

Deck, Buffer, and HootSuite) provide paid-for value-added services. These services allow users to manage and setup tweet scheduling, intelligent retweeting and adjusting tweeting frequency to maximise reach to their audiences (*i.e.* Twitter followers) through daytime posting of tweets, or tweeting during spikes of social activity. Similarly, social media optimisation apps (*e.g.* SocialFlow) that run their own proprietary URL link proxies help users to amplify the delivery of messages through timing and utilisation of key engagement metrics (such as clicks per tweet, retweets per tweet, followers, *etc*).

Quite a lot of work has been done in creating automated techniques for summarisation, categorisation and generation of text. One of the more popular works by Hovy [47] focused on a series of studies over a number of years on automated generation of multi-sentence texts. The paper argued that the central structural role of textual discourse is determined by communicative intentions. Mainly the work describes the discourse structure relations by focusing on things such as sentence planning and text formatting. Another popular work by Hovy *et al.* [46] focuses on internal workings of a system called SUMMARIST. This system identified topics, structural position of a piece of text, bonus phrases (likely candidates for summary) *vs.* stigma phrases, topic signature and discourse structure identification (text being a hierarchical structure of sentences).

In another work Huang *et al.* [49] propose an integrated solution to construct an abstraction of content that allows users to consume meaningful units of extracted content. The proposed technique integrates different media sources, such as from broadcast news, to generate semantic hierarchical representation of content. The authors perform a two stage process that (*i*) recognises anchorperson from a broadcast news using Gaussian Mixture Models, and (*ii*) news story extraction through text-based discourse tokenisation. They evaluated the technique to find a news classification error rate of less than 10% and anchorperson identification to have an accuracy of 92%.

Researchers have uncovered several ways that enable social botnets to evade detection approaches. For instance, Zhang *et al.* in [94] found that if Twitter bots are placed in a simple 10-ary tree of depth 2 with root to post spam messages and descendants to retweet, only the root bot gets suspended. A simple reallocation of root bot among the descendants can carry the process forward with descendants remaining unsuspended every time.

Ji *et al.* [51] perform a comprehensive study of social bot detection avoidance

mechanisms. First, bots exploit implicit trust on content coming from friends of users, that enables propagating content rapidly through retweets [93]. In fact, malicious URLs also spread faster and cover a wider range [14] while using URL shorteners to hide true URL domain to avoid blacklisting [87]. Second, keeping track of their activities through cookies, use of C&C centres for coordination [95], and having a hierarchical root-descendant setup to avoid large-scale account suspension [94]. Third, bots can imitate activities of humans on OSNs [74] to avoid or lower suspicion level. Fourth, purposefully and randomly delaying an action, *e.g.* tweeting, retweeting, responding, *etc.* The authors also suggest improvements to current detection mechanisms by using information derived from above mentioned behaviours.

2.1.13 Typification of bots

There is hardly any research that explores a general methodology to categorise bots. However, research has focused on topical analysis, such as bots running political campaigns (see § 2.1.9) during 2016 U.S. presidential election for and against the two main candidates, Donald Trump and Hillary Clinton. While it was found that bots vastly followed Donald Trump and positively campaigned for him, usage of pro-Clinton bots were not far behind. I list a few astonishing insights on Trump-Clinton bot campaigning in Chapter 6.

Bots were also found to be involved in a disinformation campaign during 1202 Mexico presidential election, and also against Emmanuel Macron during 2017 French presidential election in support of the far-right candidate Marine Le Pen. See § 2.1.9 for more on bots used in politics.

Mostly, I perform completely new work using Chapters 4–6 to typify bots into various categories, learned automatically by the bot classification algorithm from the characterised dataset.

Chapter 3

Stweeler: Twitter Computation System

In this dissertation I design and develop a framework, Streaming Twitter Computation System (STCS), dubbed ***Stweeler***[1], as one of the major contributions of this research. Throughout the course of this dissertation, from Chapters 4–6, ***Stweeler*** will evolve from being a data collection and characterisation tool to a fully-functional machine learning driven data science framework that enables bot classification and typification. ***Stweeler*** enables (among other things): (*i*) extracting most representative features and behavioural properties to differentiate automated agents from humans, (*ii*) automated supervised learning to discern agents from humans, (*iii*) automated typification of agents to distinguish various categories, (*iv*) topic modelling and sentiment analysis, and (*v*) analysing the influence of Web bots.

3.1 Research Questions

I ask a set of most pressing research questions for bot analyses to understand what the ***Stweeler*** framework should be (§ 3.3) for exploring answers to these questions.

Bots *vs.* humans: The first key aspect is the differences and similarities of bots from humans. What are the key activities of bots compared to humans, when measured through *content generation*, *content popularity* and *content con-*

[1] ***Stweeler***– `https://github.com/zafargilani/stcs/`

sumption. What is the quantity of activity and content generated by bots and humans? What is the degree of similarity between content produced by humans and content produced by bots? Which attracts more attention or which drives popularity and why? Do bots form *critical nodes* and *largest connected components* of the social graph? Can we accurately detect bots using the above knowledge?

Bot engagement, impact and types: The second key aspect is how bots engage with other users of the social media platform. In what different capacities are bots used to disseminate content? Do bots manipulate popularity of content, *i.e.* make topics 'trend'? Would bots impact network systems in future by generating more content than humans do? Can bots be generalised into *categories*? Do bots have preferred *topics*? Do bots represent certain *sentiments* like humans do?

3.2 What is Twitter? Why and how do bots exist on Twitter?

The word *twitter* means 'a call consisting of repeated light tremulous sounds' – similar to chirps from birds. The product name according to Jack Dorsey[2], founder of Twitter, exactly denoted the philosophy of the company, *i.e.* a platform for short bursts of inconsequential meaningless information, where meaning is entirely dependent on the recipient.

The existence of bots on Twitter is owed to three main reasons. First, Twitter identifies itself as an *information social network*, thus clearly focussing on global reach and wide social penetration. This focus meant that the business would generate wide-scale usage, adoption and economics by allowing developers to create thin clients, apps and tools atop the platform. Thus, Twitter provides publicly accessible APIs that enable both organisations and individuals to algorithmically program, control and automate actions on the platform.

Second, organisations and businesses, governments and individuals use Twitter for a multitude of purposes, either organically (via human operators) or inorganically (via automation or bot operators). Using automation legitimately

[2]Jack Dorsey talks about Twitter's founding document (last accessed 14 Jul 2018) – `http://latimesblogs.latimes.com/technology/2009/02/twitter-creator.html`

provides organisations and individuals an accelerated path to attain global reach in quick time while incurring fractional costs.

Third, registering a Twitter account is usually a simple process. Individuals are usually expected to provide an email address, pass a *soft* inspection through CAPTCHA recognition, and more recently a mobile phone number to verify individuals and promote fair usage. However, bypassing or circumventing the mobile phone requirement has been found to be easy due to a number of options, such as virtual mobile networks[3]. Given the realtime global reach, a massive 336 million active monthly user-base[4] and an easy registration process, Twitter inadvertently becomes a great enabler for both illegitimate and *dark* activity such as spam, astroturfing, trolling, social and political campaigning, *etc.*

3.3 *Stweeler* Framework

The ***Stweeler*** analysis framework is laid out in Figure 3.1 as a toolkit that comprises of a number of modular components. The components include: stream collectors (for data collection); stats, decomposition and graphs (for exploratory bot *vs.* human comparison); classifiers (for bot detection); clustering and language processing (for bot typification, topic and sentiment analysis). It accepts raw tweets (usually in JSON format) as inputs (left), processes the inputs using the toolkit (centre) and outputs the analyses (right). The framework contains a tool to run a third-party bot detection tool via a callable API. The framework also presents a bot and a web server for an alternate study on affects of bots on content popularity.

Bot behaviour has been often found to vary from human behaviour [34, 50]. Using the insights derived from Twitter account properties I can perform classification to label users as 'bot' or 'human'. Properties as indicated and measured in [18] include tweets, retweets, follower-friend ratio, URLs posted, and sources or devices used to tweet. These properties are augmented by doing an in-depth study to differentiate between bots and humans, such as account age, media types uploaded, size of media uploaded, account favouriting frequency, favourites

[3]Using Google to bypass Twitter phone verification (last accessed 14 Jul 2018) – `https://woorkup.com/how-to-bypass-the-twitter-phone-verification-for-new-account/`

[4]Twitter active monthly users Q1 2018 (last accessed 14 Jul 2018) – `https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/`

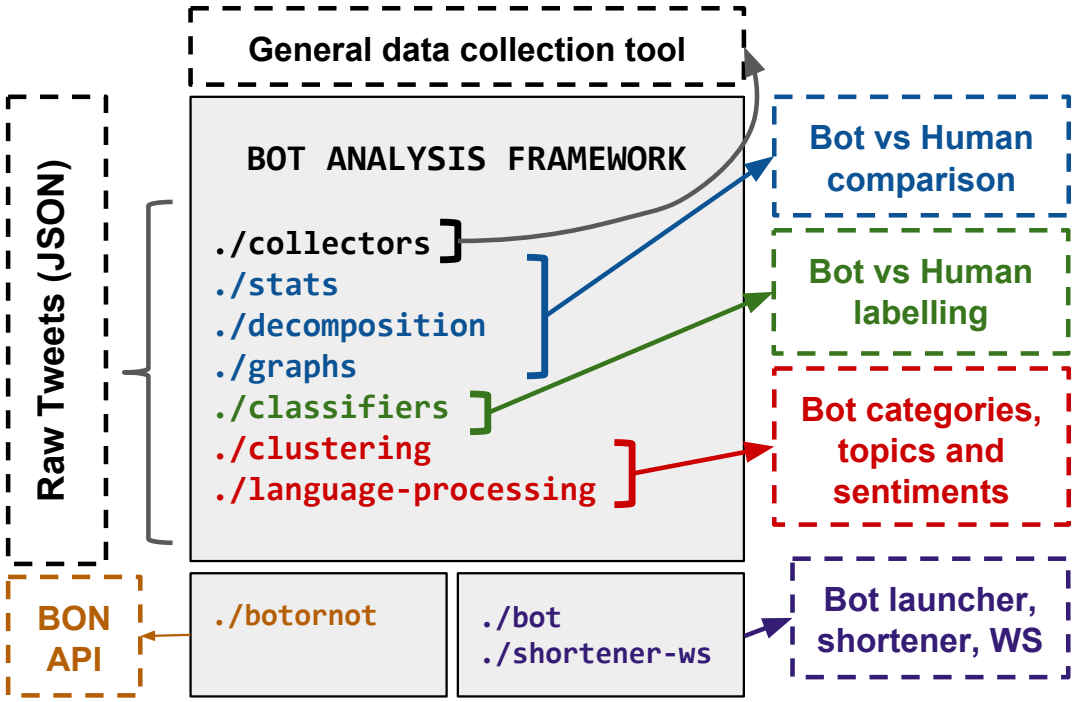


Figure 3.1: ***Stweeler*** analyses framework.

received, retweets received, *etc.* Such a study enables building a bot classification model that can reliably differentiate bots from human users.

The language processing module dissects content based data such as trends, topics, sentiments, keyword, popular hashtags and (if available) geo-coordinates to provide bot impact on Twitter in terms of activity and data volume generated. This will also analyse bot influence on Twitter in terms of followers, and how much the bots morph OSNs and relationship trees. The nature of the bot (content producer or consumer) will determine the nature of the impact. Using text classification and sentiment analysis I could categorise bot types into news, marketing or advertisements, social or political campaigning, spam or suspicious.

3.4 Datasets, Feature Extraction and Annotation Methodology

In this section I describe various datasets: characterisation and detection dataset, creation of human annotated dataset (used in Chapters 4–5), the typification

dataset (used in Chapter 6), and the honeypot dataset (used in Chapter 6).

As part of the aforementioned framework I devise a smart yet simple way[5] of collecting vast amounts of data from Twitter's publicly accessible Streaming API. A generic data collection software is a premium tool in exploratory data science since it enables exploring related aspects of a problem-space. It also mitigates the problems associated with collecting new datasets and ensuring quality and conformance with previous ones, and most importantly solves the issue of archiving historical datasets[6]. I do not filter by any keywords, location or language and collect everything offered by the Streaming API.

3.4.1 Characterisation and Detection dataset

For the purposes of characterisation in Chapter 4, I needed a dataset that could form the basis for establishing a detailed understanding in differences between bots and humans. This dataset is later labeled (§ 3.4.3) to be used as *ground-truth* labels for the purposes of detection by training a classifier in Chapter 5. Using the ***Stweeler*** collector I curated a raw tweet dataset for 30 days in April 2016. This raw dataset is approximately 65 million tweets recorded for approximately 2.9 million unique accounts.

This data contains a range of accounts across the spectrum of popularity (*i.e.* number of followers), from most popular (celebrity status) to least popular (virtually unknown). The purpose, activity and influence of an account differs based on popularity exercised passively (follower count) or actively (through tweets and mentions), as noted in [16]. Hence, I partition profiles into popularity groups to enable a detailed understanding of the dataset. The hypothesis behind dataset partitioning is that popularity intrinsically reveals profile and network attributes. For instance, most credible accounts will have high following, whereas it is much more likely that spam/malicious or *dark* accounts will have lower popularity. In other words, most popular accounts are mostly legitimate, irrespective of being automated or human operated.

The partitions are described as follows:

$\mathbf{G_{10M+}}$− celebrity status: This is the subset of Twitter users with the highest number of followers, *i.e.* >9M followers. These are the most popular

[5] ***Stweeler*** collector – `https://github.com/zafargilani/stcs/blob/master/lib/collector.rb`

[6] The collection service, if kept running, would automatically segregate tweets into daily files.

users, who hold celebrity status and are globally renowned. Popular and credible organisations (*e.g.* CNN, NatGeo) use these accounts for various purposes, which makes them free of spam, thus having high credibility and trustworthiness.

$\mathbf{G_{1M}}$– very popular: This subset of Twitter users is amongst the most popular on the platform, *i.e.* 900K to 1.1M followers. These users are close to celebrity status and global recognition (*e.g.* nytfood, pcgamer).

$\mathbf{G_{100k}}$– mid-level recognition: This subset represents popular accounts with mid-level recognition (*e.g.* Amtrak, CBSNewYork), *i.e.* 90k to 110k followers.

$\mathbf{G_{1k}}$– lower popularity: This subset represents more ordinary users, *i.e.* 0.9k to 1.1k followers. These users (*e.g.* hope_bot, Taiwan_Agent) form a large base and, though they show lower individual or accumulated activity, they do form the all-important tail of the distribution.

I create four partitions as it succinctly covers the entire popularity spectrum, from most to least popular, while clearly differentiating bots and humans. $\mathbf{G_{10M+}}$ and $\mathbf{G_{1M}}$ are similar in their characteristics (see § 4.3) and constitute 0.65% of the total 105k accounts I partitioned in the dataset. $\mathbf{G_{1k}}$ represents the bulk of Twitter, constituting 94.40% of the total partitioned accounts. $\mathbf{G_{100k}}$ bridges the gap between the most popular and least popular groups, constituting 4.93% of the total partitioned accounts. A possible $\mathbf{G_{10k}}$ would be similar to $\mathbf{G_{1k}}$, and a possible $\mathbf{G_{50k}}$ will be similar to $\mathbf{G_{100k}}$.

The dataset[7] is a representative sample as shown in § 5.4.

3.4.2 Feature Extraction

Using tweets from these user profiles I extract all associated metadata and compute values for features (*e.g.* number of tweets). I then use Principal Component Analysis from `scikit-learn` [67] machine learning library[8] to test the relevance and importance of selected features. A set of 22 features across account profile, network and activity reveals σ^2 of almost 100%. This means that this feature-set is representative of most of the variance found in the dataset. The feature-set and associated hypothesis is listed in Table 3.1.

In addition to known features studied in [18, 22] (age, tweets, retweets, favourites, replies and mentions, URL count, follower-friend ratio, *etc*), I also analyse a set of

[7]Datasets can be found here – `http://www.cl.cam.ac.uk/%7Eszuhg2/data.html`

[8]***Stweeler*** PCA – `https://github.com/zafargilani/stcs/blob/master/lib/decomposition/pca.py`

Table 3.1: Features

Feature	**Description and Hypotheses**
Age of account	The age of the Twitter account in days. The assumption is that humans have older accounts.
Favourites-to-tweets ratio	'Favourites' or 'likes' received for all user tweets. I expect humans to receive more 'likes'.
Lists per user	Lists subscribed to. I expect bots to follow more lists for obtaining lists of users to follow.
Followers-to-friends ratio	Previous research [18] shows that humans typically have this ratio close to 1.
User favourites	Tweets 'favourited' by a user. 'Liking' a post suggests an agreement, thus it should point to human-like behaviour.
Likes/favourites per tweet	'Favourites' received by a user. I expect humans to receive more 'likes', owing to content originality and topic diversity.
Retweets per tweet	'Retweets' received by a user. I expect humans to receive more 'retweets', owing to content originality and topic diversity.
User replies	Tweets replied to by a user. I assume humans will engage in conversations with other users, but bots will not.
User tweets	User-generated tweets. Bots should tweet more aggressively, given that they do not experience 'human'-like limitations.
User retweets	User-generated retweets. Aggressive retweeting is a sign of automation [18].
Tweet frequency	Daily tweet frequency of a user. Bots are expected to tweet much more often than humans per day.
URLs count	URLs are used to redirect traffic to elsewhere from Twitter platform. Presence of URLs within tweets suggests automation [18].
Activity source type	A 'source' is the endpoint from where a user posts tweets, denoted as S_n. This is categorised as: browser or web client (S_1), mobile device apps (S_2), social media management apps (S_3), social media scheduling and automation (S_4), marketing and brand promotion (S_5), news content web services (S_6), any other not part of the defined list (S_0). Humans are expected to use S_1, S_2, and S_3; whereas bots are expected to use S_4, S_5, and S_6.
Source count	The number of the endpoints used. I assume humans will use more sources.
CDN content size	Content (pictures and videos, respectively) uploaded on Twitter. Bots should be able to upload more content on Twitter.

eight novel features not explored in past bot research. These are: (*i*) *favourites-to-tweets ratio*, (*ii*) *lists per user*, (*iii*) *likes/favourites per tweet*, (*iv*) *retweets per tweet*, (*v*) *user replies*, (*vi*) *7 activity source identity (or source type) categories*, (*vii*) *source count*, and (*viii*) *CDN content size*. The selection of features is driven by [25].

In addition to the list of features, Table 3.1 also lists hypotheses or assumptions attached to these features. The hypothesis per feature per entity (bot or human) indicates the expectation of how it might perform. Deviation from expected behaviour per feature per entity would define an inclination either towards bot or human behaviour.

3.4.3 Human Annotated dataset

I recruit four human participants to perform a manual data labelling or *human annotation task*[9] to identify bots and humans. Chosen annotators are trained computer scientists and active Twitter users. Each account was reviewed by all participants independently, before being aggregated into a final judgement using a majority count and final collective review (via discussion if needed). Each review was completed using a tool that automatically presents Twitter profiles for reviewing content and URLs posted. This allows the participants to annotate the profile with a classification (bot or human) and add any extra comments.

The participants were asked to check each profile generally but paying special attention to the activity during the month of April 2016. Particular attention to activity in April 2016 was necessary to assure that account activity stays consistent, thus justifying the annotation. For performing reviews the participants were given Twitter profiles as well as summary data. This included information already available on each Twitter profile, such as: account creation date, average tweet frequency, content posted on user Twitter page, account description, whether the user replies to tweets, likes or favourites received and the follower-friend ratio. Availability of this information enabled the participants to find any changes in profiles from April 2016 to other observed time periods.

I also provided participants with a list of the 'sources' used by the account over the month, *e.g.* Twitter app, browser, *etc.* The human workers consider both the number of sources used, and the types of sources used. This is because sources can reveal traces of automation, *e.g.* use of the Twitter API. However, the participants were asked to weigh their best judgement over what the task document described. This would mitigate the possibility of biasing the results along with considerations such as detailed observation, individual judgement and final discussion for unclear or difficult profiles.

Overall, I presented participants with randomised lists that fell into the four popularity groups described in § 3.4.1. Human annotators were instructed to filter out any account that matched the following criteria: an account that does not exhibit activity (*i.e.* no tweet, retweet, reply, and mention), and an account that is suspended. Each account is marked as either human or bot, and final

[9]Details of human annotation task can be found in Appendix A.1 or at `http://www.cl.cam.ac.uk/%7Eszuhg2/docs/papers/human-annotation-task.txt`

ground truth labels are used *iff* majority vote holds between all annotators. This majority vote is the final annotation that is derived from the four annotations. If there is a tie (*i.e.* 2-2 vote split among annotators) it is discussed among the annotators and re-annotated for a majority vote (*i.e.* for final annotation). In total, the volunteers successfully annotated 3,536 active accounts: 1,525 were classified as bots (43.12%), 2,010 as humans and 1 tie.

Though ties are an exception in my dataset (1 out of 3,536), it is important to highlight the importance of properly handling noisy labels. There are three approaches in current research to tackle this problem: (*i*) detecting and correcting incorrect labels [2, 78], (*ii*) weigh the data labels using a loss function according to peripheral information such as noise rates [64, 53, 59], and (*iii*) ignoring or discarding the noisy labels [52, 62, 60, 24]. Incorrect or noisy labels generally tend to mislead learning models [24], especially if they are in high proportions. To handle this I devised a simple solution by having annotators revisit a tie by having an open discussion. The purpose of the discussion is to quickly highlight individual findings, view the account collectively and re-annotate to what the majority decides. This approach provides quality results and does not deviate from the majority vote requirement for an annotation decision (*i.e.* final annotation).

Annotated partitioned groups are described as follows:

$\mathbf{G_{10M+}}$– celebrity status: Out of a total of 102 accounts, 50 were successfully annotated within the given timeframe. Out of these 50 user profiles, 24 are identified as bots and 26 as human accounts.

$\mathbf{G_{1M}}$– very popular: Out of a total of 893 such accounts in my dataset, 746 accounts were successfully annotated within the given timeframe. Out of these 746 user profiles, 295 are identified as bots, 450 as human accounts, and 1 tie. This tie is annotated in the dataset as a 'human' as majority of the annotators after a discussion were convinced of this account being operated by a human.

$\mathbf{G_{100k}}$– mid-level recognition: Out of 9691 such accounts, a total of 1,447 were successfully annotated within the given timeframe. Out of these 1,447 user profiles, 707 are identified as bots, and 740 as human accounts.

$\mathbf{G_{1k}}$– lower popularity: Out of 795,861 such accounts, only 1,293 accounts were annotated successfully within the given time. Out of these 1,293 user profiles, 499 are identified as bots and 794 as human accounts.

Summary of the annotated data is provided in Table 3.2.

Table 3.2: Summary of Twitter dataset post-annotation.

Group	#Bot accts	#Human accts	#Bot statuses	#Human statuses
$\mathbf{G_{10M+}}$	24	26	71,303	79,033
$\mathbf{G_{1M}}$	295	450	145,568	157,949
$\mathbf{G_{100k}}$	707	740	148,015	82,562
$\mathbf{G_{1k}}$	499	794	24,328	13,351
Total	1,525	2,010	389,214	332,895

3.4.4 Typification dataset

For the purposes of exploring bot categories in Chapter 6, I collect a completely new dataset using Streaming API for 30 days in December 2016. The total data collected was approximately 65 million tweets, with information recorded on approximately 3 million unique accounts. This dataset is different to the one described in § 3.4.1–3.4.3 and used in Chapters 4–5. The reason for this change is to obtain a larger dataset for an in-depth exploration of types of bots. Moreover, this mitigates the problem of using past datasets that might contain suspended, deactivated and deleted Twitter accounts.

I initially collect data on 3 million accounts, out of which the ***Stweeler*** bot classifier identifies 11,379 as humans and 11,102 as bots. This reduction occurs due to two main reasons: (*i*) filtering inputs, such as removing suspended accounts and accounts with no tweets, and (*ii*) time constraints – the ***Stweeler*** bot classifier was kept running for a week for a sizeable dataset, though theoretically it could exhaustively process the raw dataset for an input of any number of days. Next, the dataset is cleaned by removing all empty lines from these tweet files, and removing all those bot users which had produced less than two tweets. I remove low activity bots to achieve higher accuracy during the clustering task. Some of these low activity accounts are classified as bots because of the activity source endpoint they use (such as automated services), having very low account reciprocity (0 followers or 0 friends) and lack of an original tweet.

Next I augment the dataset in § 3.4.4 to detect languages and translate non-English text to English text in order to label categories more conveniently and accurately[10]. I only use the most popular languages on Twitter to capture maximum data without compromising performance, *i.e.* English (34%), Japanese (16%), Spanish (12%), Portuguese (6%), Arabic (6%), French (2%), and Turkish

[10]There is scarcity of reliable and accurate non-English topic modelling tools, thus applying a limit to translate non-English corpus to English.

(2%). To detect the language I employ Python's `langdetect` library, and to accurately translate I use Python's `textblob` library for improved results. Though `textblob` can also be used to detect text language, it is much slower compared to `langdetect` since it uses the massive `nltk` corpora database. The `langdetect` on the other hand utilises Google's language detection database. Execution performance aside, both of the toolkits provide high accuracy for language detection and manipulation. Figure 3.2 shows the original text and Table 3.3 shows the translated text from one such example.

RT @AJArabic: الأمم المتحدة تتهم #حزب_الله بعرقلة
تنفيذ اتفاق إخلاء المحاصرين من أحياء شرق #حلب

Figure 3.2: Accuracy of language detection (`langdetect`) and translation (`textblob`) libraries: Original text.

Table 3.3: Accuracy of language detection (`langdetect`) and translation (`textblob`) libraries: Translated text.

Conversion type	`ar` → `en`
Translated text	RT @AJArabic: UN accuses Hezbollah of obstructing implementation of evacuation agreement

The accuracy of these libraries for complex phrases might be a topic for further discussion. However, for the purposes of generating topic models from text corpus of bot categories the accuracy is acceptable. Table 3.4 shows the summary of the dataset used for typification in Chapter 6.

Table 3.4: Summary of Twitter bot dataset (Dec 2016) for typification.

Stat	**Count (%age of total dataset)**
Extracted bot accts	11,102 (100%)
# Extracted statuses	951,481 (100%)
Processed bot accts	5,088 (45.83%)
# Processed statuses	715,081 (75.15%)
# Translated statuses	446,378 (46.92%)

3.4.5 Honeypot dataset

To explore the impact of Web bots on popularity of content posted on Twitter in Chapter 6, I perform a honeypot experiment by deploying a Twitter bot.

Table 3.5 shows statistics for data collected by my Web server from 21-11-2015 to 08-01-2017. My Twitter bot account received more than 223,000 clicks, out of which more than 44.91% had been performed by bots. Surprisingly, the volume of activity of Twitter bots (53.90% of total statuses) and Web bots (44.91% of total clicks) on Twitter is very similar. Details of the experiment and the results of the analyses are presented in § 6.5.

Table 3.5: Click logs dataset – statistics.

Fact	**Figures**
Timeframe	From 21-11-2015 to 08-01-2017
Total clicks	223,062
Clicks by bots	100,194 **(44.91%)**
Unique visitors	2,563
Unique recurring bots	113 **(4.08%)**

3.5 *Stweeler* Dashboard

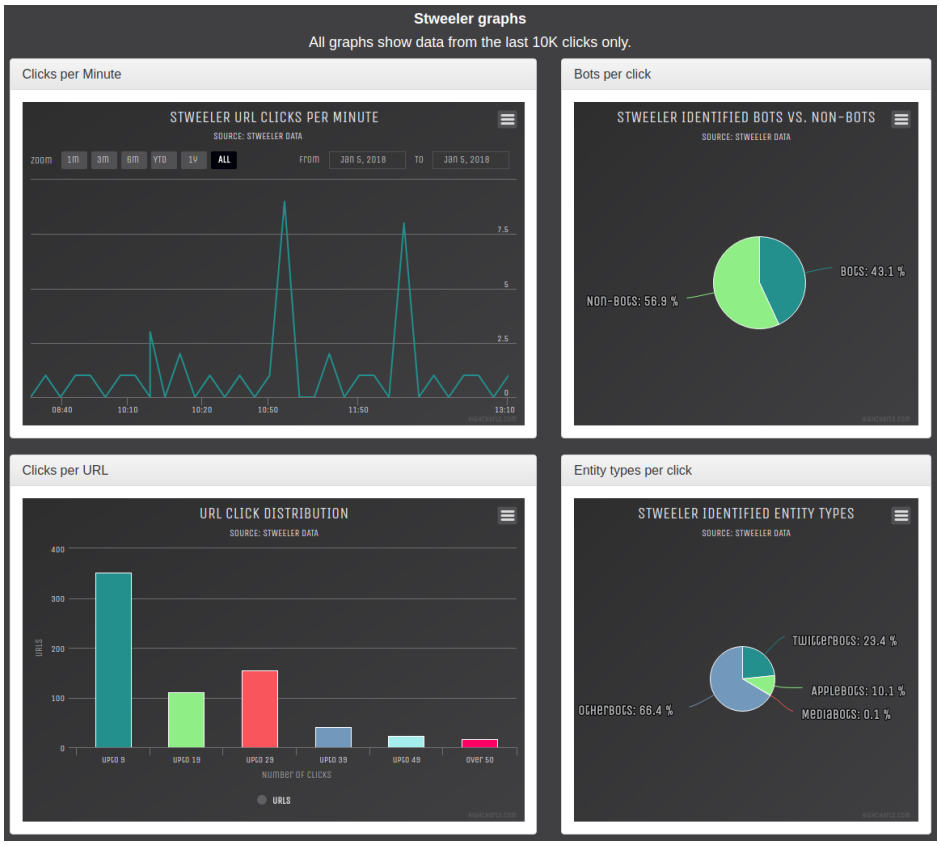


Figure 3.3: ***Stweeler*** dashboard.

This work also contributes a live non-invasive non-engaging Twitter bot and a dashboard from the live clicks dataset collected using the bot (Chapter 6). Using a live Web server I deployed a Twitter bot for a honeypot experiment that captures live clicks by other bots that interact with open Twitter content. These bots could be Twitter bots or wider Web bots (such as content curators, crawlers and spiders). The Web server has a dashboard[11] to display analytics around the clicks dataset (Figure 3.3). More can be found about the bot and how live clicks dataset is used in Chapter 6.

3.6 Takeaways

In this chapter I presented a list of important questions, explained how and why bots exist on Twitter, and presented the ***Stweeler*** framework as an effective tool to study the bot presence. I aim to answer most of the raised questions by using the ***Stweeler*** analyses framework to build a comprehensive understanding of the bot population on Twitter. In the chapters that follow, I perform a detailed characterisation of bots and humans (Chapter 4), using these characterisations I implement a detection algorithm (Chapter 5), perform bot typification to explore and understand types of bots (Chapter 6) and finally conclude in Chapter 7.

[11] ***Stweeler*** dashboard – `http://svr-szuhg2-web.cl.cam.ac.uk/graph/graphs`

Chapter 4

Measuring and Characterising Social bots

In the previous chapter I listed the contributions of this dissertation. In this chapter I utilise ***Stweeler*** to extract a wide spectrum of features. I study these features in detail for the purposes of an in-depth comparative analyses on the usage and impact of bots and humans on Twitter. In order to accomplish this I collect a large-scale Twitter dataset and define various features based on tweet metadata using ***Stweeler***. The human annotation task (§ 3.4.3) is used to assign 'bot' and 'human' ground-truth labels to the dataset. The annotations are compared against a state-of-the-art bot detection tool for evaluation (I build my own bot detection tool in Chapter 5). I then ask a series of questions to discern important behavioural characteristics of bots and humans using features within and among these popularity groups. From the comparative analysis I draw differences and interesting similarities between the two entities, thus paving the way for reliable detection of bots in Chapter 5. Moreover, this enables exploring influence and categories, and extends the ***Stweeler*** platform so it can be used for studying automated political infiltration and advertisement campaigns.

4.1 Introduction

The rise of bots constitutes a radial shift in the nature of content production, which has traditionally been the realm of human creativity (or at least intervention). Although there have been past studies on bots (see § 2.1), this chapter

is particularly focused on exploring their role in the wider social ecosystem, and how their behavioural characteristics differ from humans. This is driven by many factors. The limited cognitive ability of bots clearly plays a major role, however, it is also driven by their diverse range of purposes, ranging from curating news to answering customer queries. This raises a number of interesting questions regarding how these bots operate, interact and affect online content production: What are the typical behaviours of humans and bots, in terms of their own activities as well as the reactions of others to them? What interactions between humans and bots occur? How do bots affect the overall social activities? How do bots affect the overall social activities, and what would the impact of their removal be? The understanding of these questions can have deep implications in many fields such as social media analysis and systems engineering.

Beyond the social implications, the combined popularity of social media and online bots may mean that a significant portion of *network traffic* can be attributed to bots. This conjecture is not without support: according to an estimate 51.8% of all Web traffic is generated by bots[1]. This, however, constitutes a radical shift from traditional views on web traffic bringing about both new research questions and engineering opportunities. For example, can we measure the amount of traffic produced by bots? This is of importance for future network engineering, as preliminary evidence suggests that substantial amount of network congestion is caused by (low priority) bots.

Contributions of this chapter: To answer the above questions, I have performed a large-scale measurement and analysis campaign on Twitter (§ 4.2). I analyse the most descriptive features from the dataset, as outlined in a social capitalist study [25], including six which have not been used in the past to study bots. This chapter offers a new and fundamental understanding of the characteristics of bots *vs.* humans, observing a number of clear differences (§ 4.3). For example, humans generate far more novel content, while bots rely more on retweeting. I also observe less intuitive trends, such as the propensity of bots to tweet more URLs, and upload bulkier media (*e.g.* images). There are divergent trends between different popularity groups (based on follower counts), with, for example, popular celebrities utilising bot-like tools to manage their fanbase. I further analyse the social interconnectedness of bots and humans to characterise how they influence

[1]Bot traffic report 2016 (last accessed 16 June 2018) – `https://www.incapsula.com/blog/bot-traffic-report-2016.html`

the wider Twittersphere. Observation reveals that although human contributions are generally considered more important via typical features (*e.g.* number of likes, retweets), bots manage to sustain significant influence over content production and propagation. My experiments confirm that the removal of bots from Twitter could have serious ramifications for information dissemination and content production on the social network. Specifically, by simulating content dissemination I find that bots are involved in 54.59% of all information flows (defined as the transfer of information from one user to another user). I also seek to discover: (*i*) the amount of data traffic bots generate on Twitter, and (*ii*) the nature of this traffic in terms of media type, *i.e.* URL, photo (JPG/JPEG), animated image (GIF), and video (MP4). This chapter also sheds light on the possibilities of how this ever-increasing bot traffic might affect networked systems and their properties. As well as providing a powerful underpinning for social bot detection (Chapter 5), this chapter makes contributions to the wider field of social content automation. Such understanding is critical for future studies of social media, which are often skewed by the presence of bots.

4.2 Methodology

I build upon my work ***Stweeler***[2] for data collection, pre-processing, feature extraction, bot classification through human annotation, and analysis.

4.2.1 Data Collection and Feature Extraction

Every single action performed on Twitter by a user is recorded as a *tweet* (status), whether a tweet, retweet, reply or mention. I collect data on bot and human behaviour for 30 days in April 2016 from the Twitter Streaming API. This data contains a range of accounts in terms of their popularity (*i.e.* number of followers). Hence, I extract and partition user accounts into four popularity groups to enable a deeper understanding. Please see § 3.4.1 for full details about the dataset used in this Chapter. Features I consider in this study are defined in Table 3.1, details of which are explained in § 3.4.2. The feature-set along with the correlation among different popularity groups is shown in Figure 4.1.

[2] ***Stweeler***– `https://github.com/zafargilani/stcs`

4.2.2 Bot Classification via Human Annotation Task

To compare bots with humans, it is necessary to identify which accounts are operated by bots. I experimented with the updated release of BotOrNot [22, 83], a state of the art bot detection tool (to the best of my knowledge this is the only available online bot detection tool). However, inspection of the results indicated high inaccuracy with different thresholds (40% to 60%) to label an account as 'bot'. Cresci *et al.* in [21] reported similar inaccuracy measures. I cannot say for certain why BotOrNot was very inaccurate due to the internal workings (code) being kept inaccessible by its authors. However, there were three reasons in my understanding that explained why BotOrNot performed below average: (*i*) it works live and therefore can only access a subset of tweets (thus missing the complete picture), (*ii*) it is trained on old data, (*iii*) claims to use far too many features (the authors claim to use a 1,000 or more features).

Hence, I chose to take a manual approach to establish a highly reliable set of classifications, that would serve the exploratory purpose of this chapter, as well serve as *ground-truth* labels for bot detection (Chapter 5). The dataset created via this manual approach can be found in § 3.4.3. Details of the human annotation task can be found in Appendix A.1. In total, I found 43.13% bots in my Twitter dataset, responsible for 53.90% statuses.

For context, I cross validated by comparing the agreement of final annotations by the human workers to the BotOrNot annotation. The average inter-annotator agreement compares the pairs of labels by each human annotator to capture the percentage of accounts for which all four annotators unanimously agree. The average agreement is measured as a percentage of agreement: 0% shows lack of agreement and 100% shows perfect agreement. The human annotation task shows very high unanimous agreement between human annotators for each popularity group: $\mathbf{G_{10M+}}$ (96.00%), $\mathbf{G_{1M}}$ (86.32%), $\mathbf{G_{100k}}$ (80.66%), and $\mathbf{G_{1k}}$ (93.35%). Whereas, BotOrNot shows lower than average agreement with the final labels assigned by the human annotators: $\mathbf{G_{10M+}}$ (46.00%), $\mathbf{G_{1M}}$ (58.58%), $\mathbf{G_{100k}}$ (42.98%), and $\mathbf{G_{1k}}$ (44.00%). Since, BotOrNot yields a lower accuracy, I chose to use the dataset of accounts that were manually annotated. I perform a more thorough comparison with BotOrNot in Chapter 5 while designing my own bot detection tool.

4.2.3 Media Extraction and Processing

Table 4.1: Types of bot traffic uploaded by Twitter users.

Type	Description
URL & schemes	URL hosts and URI schemes (4,849 http and 289,074 https instances). These are extracted from the `[text]` tweet attribute. 162,492 URLs by bots and 131,431 by humans.
photos (JPG/JPEG)	A photos is extracted from the URL in `[media_url_https]` attribute. In total 23.31 GB of photo data is uploaded by 3,536 bots and humans in one month.
animated images (GIF)	Though these are animated photos, Twitter saves the first image in the sequence as a photo, and the animated sequence as a video under the `[video_info]` attribute. In total 2.92 GB of animated image data is uploaded.
videos (MP4)	Video files accompany a photo which is extracted by Twitter from one of the frames of the video. A video is pointed to by the URL in `[video_info][url]` attribute. In total 16.08 GB of video data is uploaded.

As well as text, users are allowed to tweet content such as video and images. These are identified by metadata within Twitter data. Table 4.1 summarises the types of media content I observed. The dataset is the same as defined in Table 3.2. For each tweet created, I extract the media and URLs. Importantly, Twitter automatically creates different resolutions of photos and videos, as well as generating images from animated sequences or videos to accompany static display with each dynamic media. Note that I *only* consider the media originally uploaded by users. This is pointed to by `[sizes][large]`. I do not consider media created or uploaded by Twitter itself as part of my dataset.

4.3 Which manners maketh the Bot?

The purpose of this study is to discover the key account characteristics that are typical (or atypical) of bots and humans. Recall that I take a broad perspective on what a 'bot' is, *i.e.* any account that *consistently* involves automation over the observed period, but may involve human intervention. This definition is justified by the purpose of automation, *i.e.* humans act as *bot managers*, whereas bots are *workers*. To explore this, I use this data (§ 4.2) to empirically characterise bots (dashed lines in figures) and humans (solid lines in figures). To begin, I simply compute the correlation between each feature for bots and humans, in order to highlight similarities and differences. Figure 4.1 presents the results as a heatmap (where perfect correlation is 1.0). Notice that most features exhibit

very poor correlations (0.0 to 0.35), indicating significant discrepancies between bot and human behaviour. I spend the remainder of this chapter exploring these differences in depth.

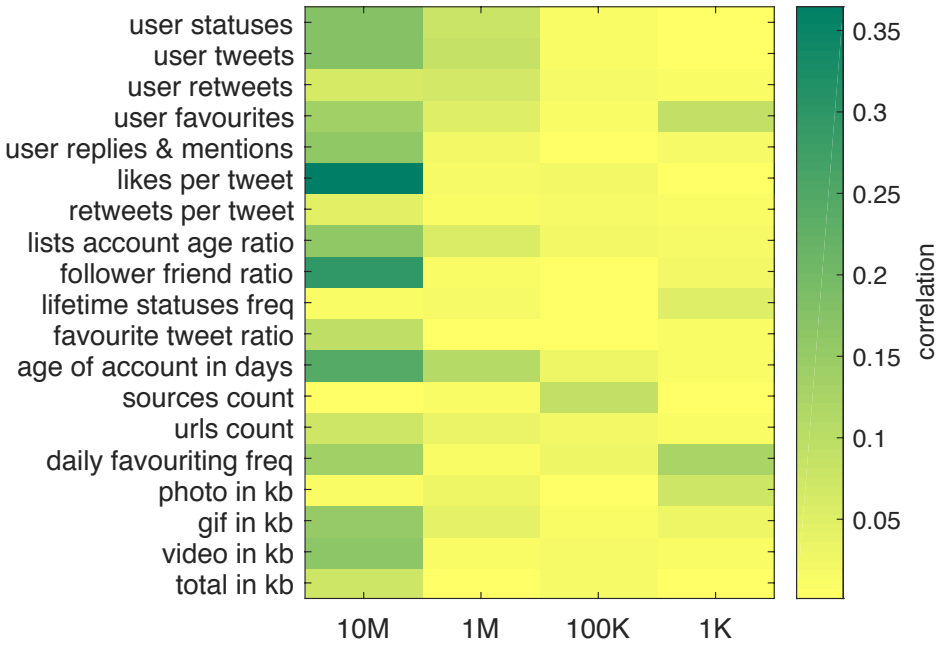
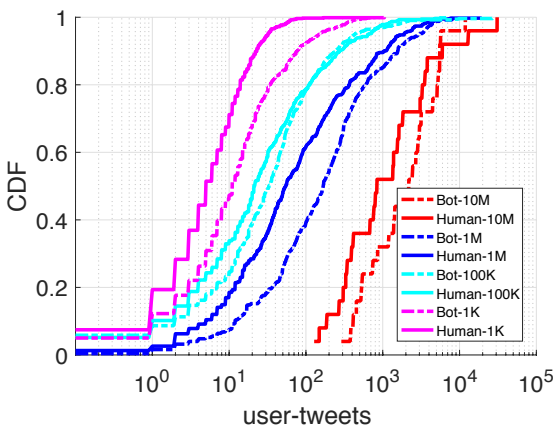


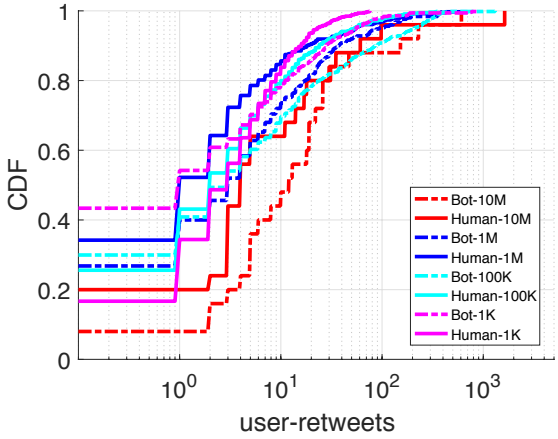
Figure 4.1: Spearman's rank correlation coefficient (ρ) between bots and humans per measured feature. The figure shows none (0.0) to weak correlation (0.35) across all features, indicating clear distinction between the two entities.

4.3.1 Content Generation

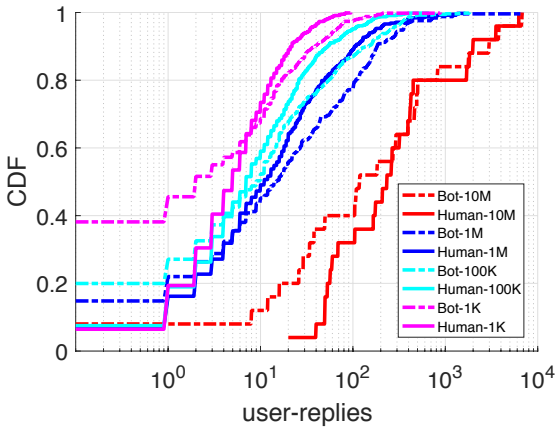
I begin by asking *if bots generate more content on Twitter than humans?* Intuitively, one might imagine bots to be capable of generating more content, however, creativity could be a major bottleneck. I initially consider two forms of content creation: a *tweet*, which is an original status written by the account, and a *retweet*, which is repetition of an existing status. As briefed earlier the term *status* refers to the sum of both tweets and retweets. First, I inspect the amount of content shared by computing the number of statuses (*i.e.* tweets + retweets) generated by each account across the 30 days. As anticipated, humans post statuses less frequently than bots (monthly average of 192 for humans *vs.* 303 for bots), in all popularity groups except $\mathbf{G_{10M+}}$, where surprisingly humans post slightly more than bots. The sheer bulk of statuses generated by $\mathbf{G_{10M+}}$ (on average 2,852



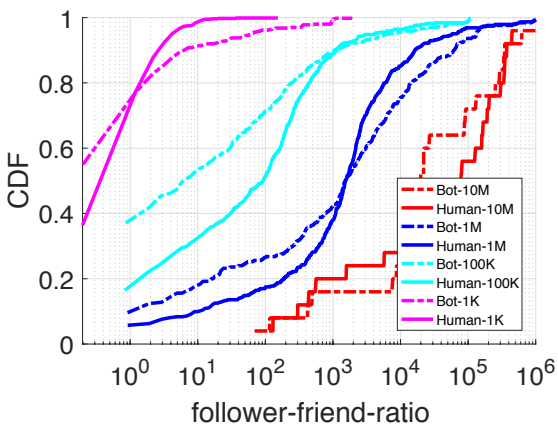
(a) Number of tweets issued by a user.



(b) Number of retweets issued by a user.



(c) Replies and mentions posted by a user.



(d) Follower-friend ratio of a user.

Figure 4.2: Content Creation: Tweets issued, Retweets issued, Replies and Mentions, Follower-friend ratio.

for bots, 3,161 for humans in a month) is likely to acquire popularity and new followers. Overall, bots constitute 51.85% of all statuses in this dataset, even though they are only 43.13% of the accounts.

An obvious follow-up is *what do accounts tweet?* This is particularly pertinent as bots are often reputed to lack original content. To explore this, I inspect the number of *tweets vs. retweets* performed by each account. Figures 4.2a and 4.2b present the empirical distributions of tweets and retweets, respectively, over the 30 days. It is observed that the retweet distribution is rather different to tweets. Bots in $\mathbf{G_{1M}}$, $\mathbf{G_{100k}}$ and $\mathbf{G_{1k}}$ are far more aggressive in their retweeting; on average, bots generate 2.20× more retweets than humans. The only exception to this trend is $\mathbf{G_{10M+}}$ where humans retweet 1.54× more often than bots. This is likely driven

by the large number of tweets generated by celebrity users. Typically, humans do generate *new* tweets more often, while bots rely more heavily on retweeting existing content. Generally, humans post 18 tweets for every retweet, whereas bots post 13 tweets for every retweet in all popularity groups except $\mathbf{G_{10M+}}$ (where both entities show similar trends).

Whereas tweets and retweets do not require one-to-one interaction, a further type of messaging on Twitter, via *replies*, does require one-to-one interaction. These are tweets that are created in response to a prior tweet (using the @ notation). Figure 4.2c presents the distribution of the number of replies issued by each account. I anticipate that bots post more replies and mentions given their automated capacity to do so. However, in $\mathbf{G_{10M+}}$ both bots and humans post a high number of replies, and bots post only marginally more than celebrities. While bot-masters in $\mathbf{G_{10M+}}$ deploy *chatbots* to address simple user queries, celebrities reply in order to engage with their fanbase. It is also possible that celebrities employ managers as well as automation and scheduling tools (§ 4.3.5) for such a purpose. Bots in the remaining popularity groups respond twice as frequently as their human counterparts. Again, this is driven by the ease by which bots can automatically generate replies: only the most dedicated human users can compete.

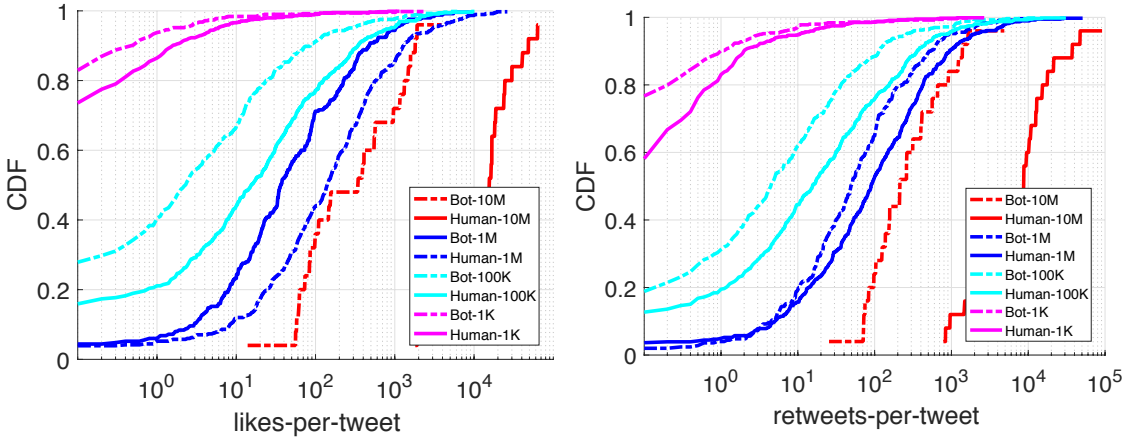
4.3.2 Content Popularity

The previous section has explored the amount of content generated by accounts, however, this does not preclude such content from being of a low quality. To investigate this, I compute standard popularity features for each user group.

First, I inspect the *number of favourites* or *likes* received for tweets generated by the accounts. This is a reasonable proxy for tweet quality, where the assumption is that bots will considerably lag behind humans. Figure 4.3a presents the empirical distribution of the number of favourites or likes received for all the tweets generated by the profiles in each group. As expected a significant discrepancy can be observed. Humans receive *far* more favourites per tweet than bots across all popularity groups except $\mathbf{G_{1k}}$. Close inspection revealed that bots in $\mathbf{G_{1k}}$ are typically part of larger *social botnets* that try to promote each other systematically for purposes as outlined in § 4.1. In contrast, human accounts are limited to their social peers and do not usually indulge in the 'influence' race.

For $\mathbf{G_{10M+}}$, $\mathbf{G_{1M}}$ and $\mathbf{G_{100k}}$ popularity groups, humans receive an average of 27×, 3× and 2× more favourites per tweet than bots, respectively. $\mathbf{G_{1k}}$ bots are an exception that receive 1.5× more favourites per tweet than humans. These findings suggest that: (*i*) the term *popularity* may not be ideally defined by the number of followers, (*ii*) human content gathers greater engagement due to its personalised attributes.

A further *stronger* sign of content quality is another user retweeting content. This is potentially an even stronger signal of endorsement, as a retweet will explicitly be listed on a user's wall. Humans are expected to receive retweets manifold as compared to bots. Humans consistently receive more retweets for all popularity groups $\mathbf{G_{10M+}}$: 24-to-1, $\mathbf{G_{1M}}$ and $\mathbf{G_{100k}}$: 2-to-1, except $\mathbf{G_{1k}}$: 1-to-1. This difference, shown in Figure 4.3b, is indicative of the fanbase loyalty, which is vastly higher for individual celebrities than reputable organisations. In other words, the *quality* of human content appears to be much higher. I then inspect *who* performs the retweets, *i.e.* do bots tend to retweet other bots or humans? We find that bots retweeting bots is over 3× greater than bots retweeting humans. Similarly, humans retweeting humans is over 2× greater than humans retweeting bots. Overall, bots are retweeted 1.5× more often than humans. This indicates a form of *homophily* and assortativity.



(a) Likes per tweet received by a user.

(b) Retweets per tweet received by a user.

Figure 4.3: Content Popularity: Likes per tweet, Retweets per tweet.

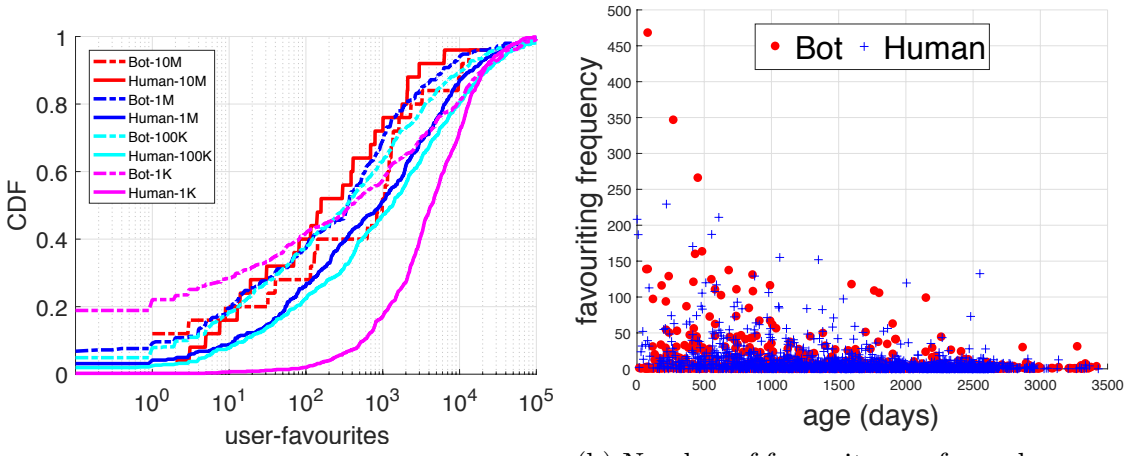
4.3.3 Content Consumption

Whereas the previous features have been based on content produced *by* the accounts under study, my dataset also includes the consumption preferences of the accounts themselves. Hence, I ask *how often do bots 'favourite' content from other users and how do they compare to humans?* Intuitively, bots would be able to perform far more likes than humans (who are physically constrained). Figure 4.4a shows the empirical distribution of the number of likes performed by each account. It can be seen that, actually, for most popularity groups ($\mathbf{G_{1M}}$, $\mathbf{G_{100k}}$, $\mathbf{G_{1k}}$), humans favourite tweets more often than bots (on average 8,251 for humans *vs.* 5,445 for bots across the entire account lifetimes). Linking into the previous discussion, it therefore seems that bots rely more heavily on retweeting to interact with content. In some cases, the difference is significant; *e.g.* humans in $\mathbf{G_{1M}}$ and $\mathbf{G_{100k}}$ place twice as many likes as bots do. $\mathbf{G_{10M+}}$, however, has an average of 1,816 likes by humans compared to 2,921 by bots.

There could be several reasons for this trend: (*i*) humans appreciate what they like, (*ii*) bots are workers for their human managers and serve a purpose (*e.g.* promotion via tweets), (*iii*) humans have an incentive to like other tweets, potentially as a social practice (with friends) or in the hope of receiving likes in return [72]. To explore these strategies further, Figure 4.4b plots the number of favourites performed by an account *vs.* the age of the account. Firstly, bots are as old as humans: the oldest bot account is 3,437 days old *vs.* 3,429 days for the oldest human account. Secondly and more importantly, it can be seen that more recent (*i.e.* more modern) bots are significantly more aggressive in liking other tweets. Older bots, instead, use this feature less frequently; deeper inspection suggests this is driven by the trustworthy nature of older bots, which are largely run by major organisations.

4.3.4 Account Reciprocity

As well as content popularity, I also measure reciprocity (*i.e.* friendship). Twitter classifies two kinds of relationships: reciprocal follower-relationship *i.e.* when two accounts follow each other, and non-reciprocal relationship *i.e.* an account has many followers who are not followed in return (this is often the case for celebrities). This is measured via the *Follower-Friend Ratio*. Figure 4.2d shows empirical distribution of the *Follower-Friend Ratio* for each group of accounts. Humans



(a) Tweets favourited (liked) by a user.

(b) Number of favourites performed *vs.* age of the account.

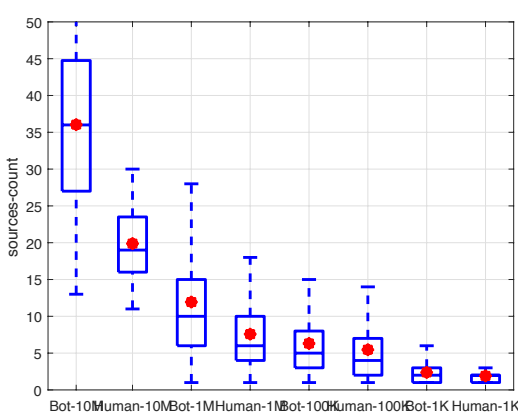
Figure 4.4: Content Consumption: Likes performed, Favouriting behaviour.

display higher levels of friendship ($\mathbf{G_{10M+}}$: 4.4×, $\mathbf{G_{1M}}$ and $\mathbf{G_{100k}}$: 1.33×, $\mathbf{G_{1k}}$: 15×) and thus a lower *Follower-Friend Ratio* than bots.

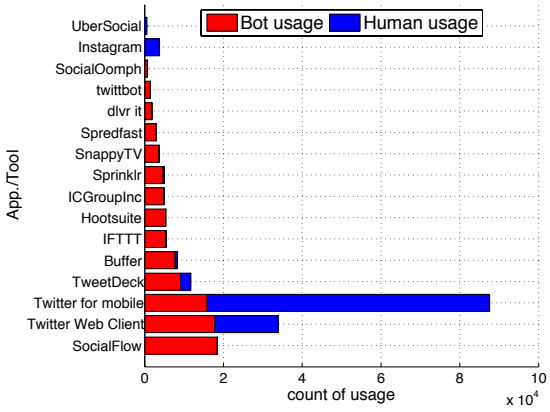
Previous research [18] argues that humans typically have a ratio close to 1, however, my analysis contradicts this assumption. For celebrities, very popular and mid-level recognition accounts this ratio is in the order of thousands-to-1, irrespective of whether an account is a bot or a human ($\mathbf{G_{10M+}}$: 629,011-to-1 for bots *vs.* 144,612-to-1 for humans, $\mathbf{G_{1M}}$: 33,062-to-1 for bots *vs.* 24,623-to-1 for humans, $\mathbf{G_{100k}}$: 2,906-to-1 for bots *vs.* 2,328-to-1 for humans). In fact, even the ratios for low popularity accounts are not 1, but consistently greater ($\mathbf{G_{1k}}$: 30-to-1 for bots *vs.* 2-to-1 for humans). This is caused by the human propensity to follow celebrity accounts (who may not follow in return), as well as the propensity of bots to indiscriminately follow large numbers of other accounts (largely in the hope of being followed in return).

4.3.5 Tweet Generation Sources

In this subsection I inspect the tools used by bots and humans to interact with Twitter. This is possible because each tweet is tagged with the *source* that generated it; this might be the website, third-party app or tools that employ the Twitter API. Figure 4.5a presents the number of sources used by human and bot accounts of varying popularities. Bots are expected to use a single source (*i.e.* an API or own tool) for tweeting. Surprisingly, it can be seen that bots actually



(a) Activity sources used by a user (Red dot is μ of the distribution).



(b) Bar chart of accounts that use each type of Twitter source.

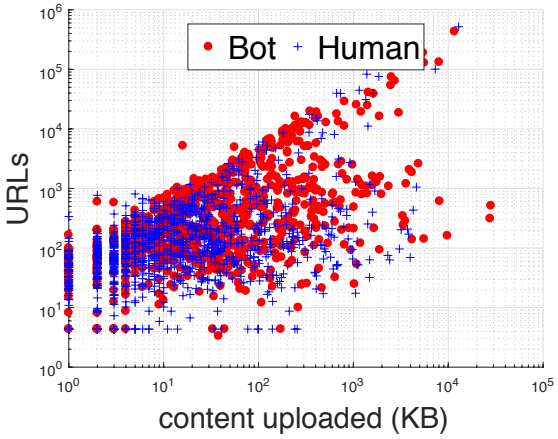
Figure 4.5: Tweet Sources: Count of Activity Sources, Type of Activity Sources.

inject tweets using more sources than humans (see Table 4.2).

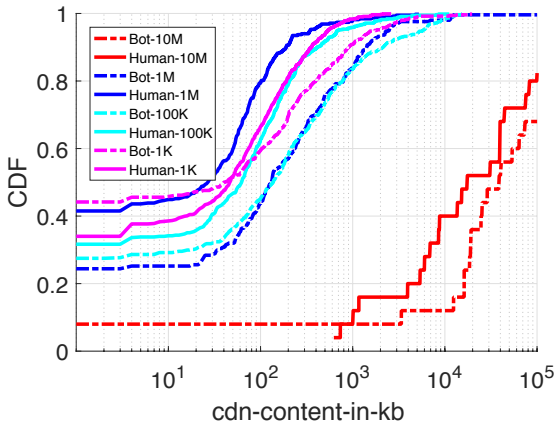
To explore this further, Figure 4.5b presents the number of accounts that use each source observed. The expectation is to observe humans utilising multiple sources (such as Web interface, app, third-party tools), expectedly more than bots (that may not always be programmed to switch from an API to third-party service, or vice versa). It can be seen, somewhat contrary to the expectation, bots use a multitude of third-party tools. Bot news services (especially from $\mathbf{G_{10M+}}$) are found to be the heaviest users of social media automation management and scheduling services *(SocialFlow, Hootsuite, Sprinklr, Spredfast)*, as well as a Cloud-based service that helps live video editing and sharing *(SnappyTV)*. Some simpler bots (from $\mathbf{G_{100k}}$ and $\mathbf{G_{1k}}$ groups) use basic automation services *(Dlvr.it, Twittbot)*, as well as services that post tweets by detecting activity on other platforms *(IFTTT)*. A social media dashboard management tool seems to be popular across most groups except $\mathbf{G_{1k}}$*(TweetDeck)*. Interestingly, it can also be seen that bot accounts regularly tweet using Web/mobile clients — pointing to the possibility of a *mix* of automated and human operation. In contrast, 91.77% of humans rely exclusively on the Web/mobile clients. That said, a small number (3.67%) also use a popular social media dashboard management tool *(TweetDeck)*, and automated scheduling services *(Buffer, Sprinklr)*. This is particularly the case for celebrities, who likely use the tools to maintain high activity and follower interaction — this helps explain the capacity of celebrities to so regularly

reply to fans (§ 4.3.1).

4.3.6 Media Upload



(a) URLs *vs.* Content uploaded.

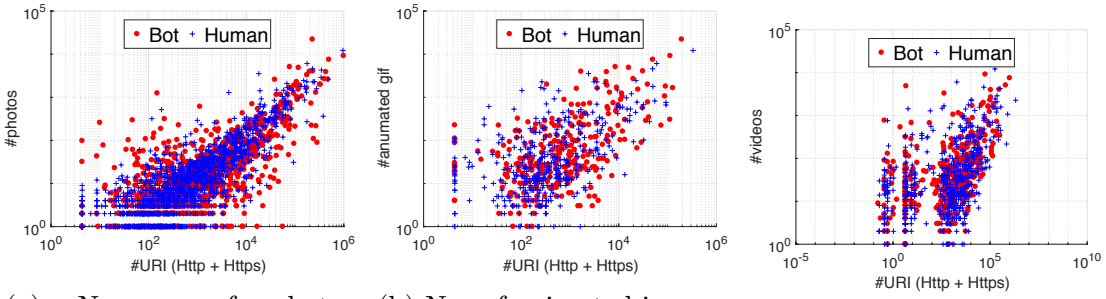


(b) Size of CDN content in KByte uploaded by a user.

Figure 4.6: Content Creation: URLs in tweets, Content uploaded on Twitter.

In this subsection I inspect the actual content of the tweets being generated by the accounts. This is done using two features: number of URLs posted by accounts, and the size of media uploaded, where bots are expected to show their actual impact. Figure 4.6a presents the scatter plot of the number of URLs (y-axis) and content uploaded in KB (x-axis). Bots place far more external URLs in their tweets than humans (see Table 4.2): 162% in $\mathbf{G_{10M+}}$, 206% more in $\mathbf{G_{1M}}$, 333% more in $\mathbf{G_{100k}}$, and 485% more in $\mathbf{G_{1k}}$. Bots are a clear driving force for generating traffic to third party sites, and upload far more content on Twitter than humans. Figure 4.6b presents the distribution of the amount of content uploaded by accounts (*e.g.* photos). Account popularity has a major impact on this feature. Bots in $\mathbf{G_{10M+}}$ have a 102× lead over bots in other popularity groups. That said, humans in $\mathbf{G_{10M+}}$ have a 366× lead over humans in other popularity groups. Overall, bots upload substantially more bytes than humans do (see Table 4.2): 141% in $\mathbf{G_{10M+}}$, 975% more in $\mathbf{G_{1M}}$, 376% more in $\mathbf{G_{100k}}$, and 328% more in $\mathbf{G_{1k}}$. This is due to their ability to automate tasks, while humans are limited by their physical capacity. It is also worth noting that both content upload and URL inclusion trends are quite similar, suggesting that both are used with the same intention, *i.e.* spreading content. Since bots in $\mathbf{G_{10M+}}$ mostly

belong to news media – sharing news headlines is clearly a means of operating their business. This resonates with the well known problem of catering demand for *heavy* users, which is well explored in cellular networks [28]. This potentially has a big impact on the traffic produced as well as the required network capacity. Since the amount of traffic is correlated to the cost and energy [84], identifying the content produced by a bot is a key step to reshaping or optimising the way that service providers should deal with this type of traffic and content.

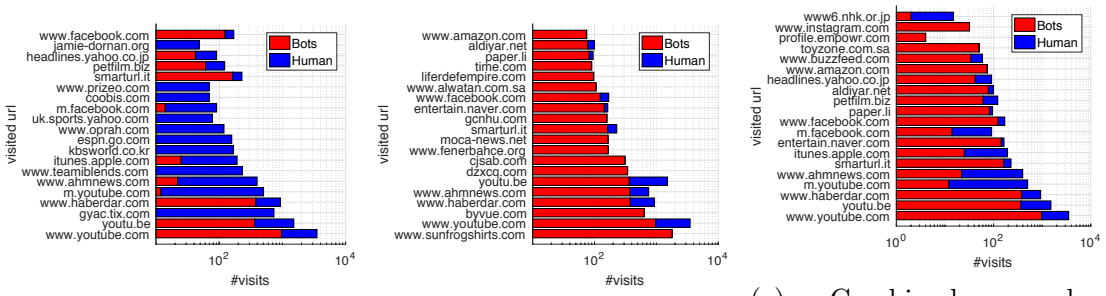


(a) No. of photos (JPG/JPEG) uploaded by bots & humans per URI (http + https).

(b) No. of animated images (GIF) uploaded by bots & humans per URI (http + https).

(c) No. of videos (MP4) uploaded by bots & humans per URI (http + https).

Figure 4.7: Media (photos, animated images, videos) uploaded by bots and humans on Twitter.



(a) Human popular URLs.

(b) Bot popular URLs.

(c) Combined popular URLs.

Figure 4.8: Visiting trends to popular URLs by bots and humans.

I can also inspect the specific types of the media uploaded. The dataset reveals a significant presence of media content generated by bots. Figure 4.7 presents a scatter plot comparing the number of media types uploaded per URI (one URI is a single object). It can be seen that both bots and humans upload significant quantities, however, it is clear that bots contribute the most. In total, bots

account for 55.35% (12.90 GB) of the total photo traffic uploaded on Twitter; 53.58% (1.56 GB) of the total animated image traffic uploaded; and 40.32% (6.48 GB) of the total video traffic uploaded on Twitter. This is despite the fact that they only constitute 43.13% of the accounts under study and contribute 53.90% of the total tweets collected. When combined, bots account for a total of 49.52% (20.95 GB) traffic uploaded on Twitter.

It is also worth noting that many bot accounts post URLs. In fact, 55.28% of all URLs are posted by bots, despite the fact that bots only make up 43.13% of the accounts. This is important because these have the potential to trigger further traffic generated amongst the accounts that view the tweets. To explore this, Figure 4.8 presents the most popular domains posted by bots and humans. Significant differences can be observed. For example, whereas humans tend to post mobile sites (*e.g.* `m.youtube.com`, `m.facebook.com`), bots rather post the desktop version (*e.g.* `youtube.com`, `facebook.com`). We can observe a range of websites exclusively posted by humans, *e.g.* `espn.com` and `oprah.com`. One can also see a few URLs posted by bots, but never by humans. These differences highlighted the differing goals of bots and humans when posting content, with more well-known websites dominating the human dataset. For example, the most regularly posted URL in my bot dataset is `sunfrogshirt.com`, which is actually a website for purchasing bespoke t-shirts. This highlights a common purpose of media posting on Twitter: spam and marketing. Note that bots infiltrate human popular URLs more often than humans infiltrate bot popular URLs. This shows that bots can reach further due to their automated ability and can considerably impact systems in unusual ways.

4.4 A World without Bots?

The previous section has discussed the characteristics that make bots and humans different. However, one of the most important things on Twitter is its social graph, *i.e.* the interconnections between users. Hence, in this section, I will briefly inspect the *social impact* or *influence* that bots have on Twitter, as well as the impact of removing them. In this context, *influence* is defined as the capacity or the ability to drive an action, *e.g.* sharing an item (whether text, photo or video) on social media that induces or generates a response. Graphs throughout

this section are created using Gephi[3].

4.4.1 How Influential are Bots?

I begin by inspecting the *social influence* that bots and humans exercise on Twitter. *Influence* (sometimes referred to as *induction*) is the phenomenon where actions of an individual are affected by other individuals through social interaction. I therefore construct a graph of direct interactions, whereby vertices are users (bots or humans), and edges represent interactions, *i.e.* retweeted statuses, quoted statuses, replies, or mentions. As previous research shows [4], influence in OSNs is directional and position-dependent (*i.e.* position in the social network). Therefore, *influence* of a user (vertex) in this context is the sum of direct interactions (edges) it has been engaged in by other users (vertices). Note that in order to engage in direct interaction, at least one user has to retweet, quote, reply or mention the other user. Furthermore, each interaction could have two perspectives from a user's viewpoint: (*i*) *influencer interaction* when a user belonging to one of these popularity groups exercises influence over another user, (*ii*) *influenced interaction* when a user is influenced by one of the users in these popularity groups.

To answer *how influential bots are*, I present interaction graphs that depict retweeted statuses, quoted statuses, replies, and mentions of bots and humans by their followers. I use two popularity groups: users with 10M and 100k followers, and the users who are involved in the direct interaction, *i.e. influenced interaction*. I do not present results for the 1M and 1k popularity groups as they show similar graphs and properties to 10M and 100k groups, respectively. I use directed edges for the interaction graphs, where an edge is directed from the *influencer* to the *influenced*.

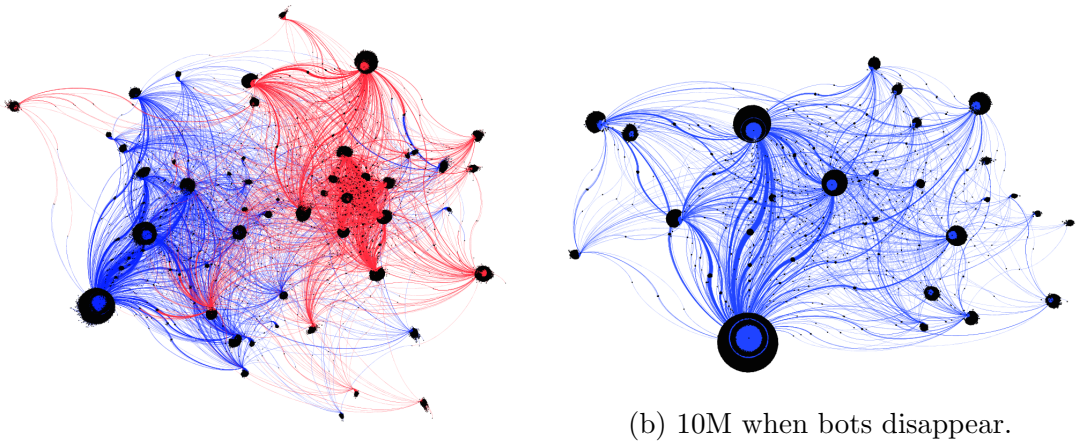
The mean degree for the 10M popularity group is very similar for both bots (1.18) and humans (1.176). This shows that both humans and bots are tightly intra-connected within their respective assortative neighbourhoods: the assortative intra-connectedness is stronger than diversified inter-connectedness. I also find that bots (4.025) have almost 2× the mean degree than humans (2.164) for the 100k popularity group. This shows that bots have accumulated a large influence both within their assortative as well as diversified neighbourhoods. This is

[3]Gephi – `http://gephi.github.io`

partly driven by the more aggressive tweeting activity of the bots under-study.

4.4.2 What happens if Bots disappear?

The above confirms that bots have significant influence in Twitter. Thus, an obvious question is *what would happen if all bots were blocked or removed from Twitter?* This may shed light on the overall impact (positive or negative) that bots have, as has been topically studied for UK-EU referendum [48] and 2016 US Presidential Election [8]. If bots produce high amounts of content (tweets, URLs, content size), then their existence should be critical for intermediary connections (or form *centrality vertices* that sit on critical paths). Such central nodes typically sustain the graph structure. Moreover, if bots are responsible for affecting content popularity (favouriting, retweeting, quoting), then they should be among the critical *super-vertices*. We will look at behaviours between retweeting and quoting graphs as well as replying and mentioning graphs.



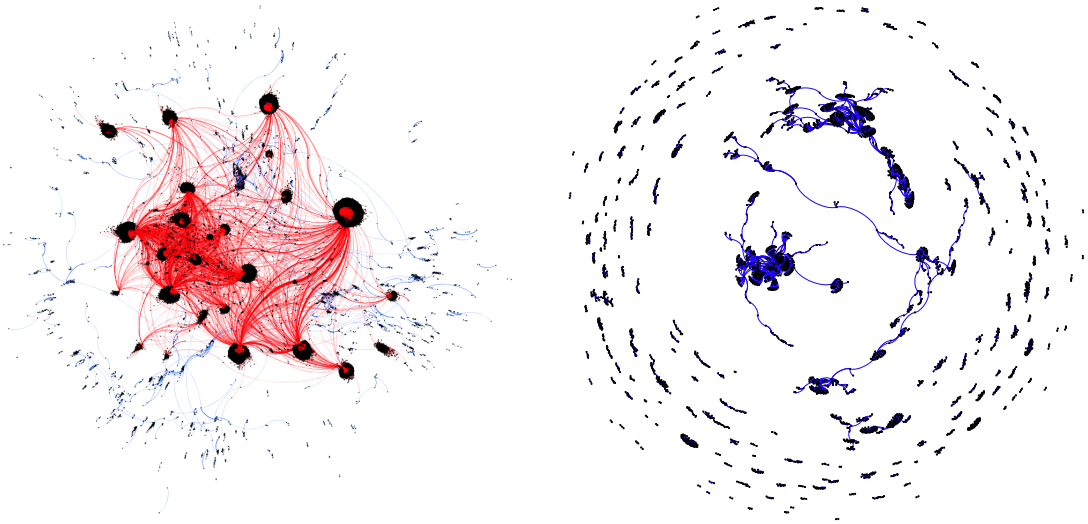
(a) 10M bots and humans.

(b) 10M when bots disappear.

Figure 4.9: Bots *vs.* Humans - graphs for retweets and quotes of 10M popularity group. Black dots are vertices, edges represent an *interaction*. Red edges represent bots and Blue edges represent humans.

Figure 4.9 presents the influence graph for the 10M group for retweets and quotes. The density of edges (due to retweeting and quoting) for both bots (Red) and humans (Blue) emphasises the influence of these vertices within their network. Notice the two separate sub-graphs appearing for bots and humans, which confirms most of the connections are between similar entities, *i.e.* bots

following other bots, and humans following other humans. Despite two separate sub-graphs, vertices of both entity types are connected to each other too, *i.e.* bots following humans, and humans following bots. This shows that *intra*-influence is stronger than *inter*-influence, *i.e.* bots influencing other bots is stronger than bots influencing humans, and vice versa.



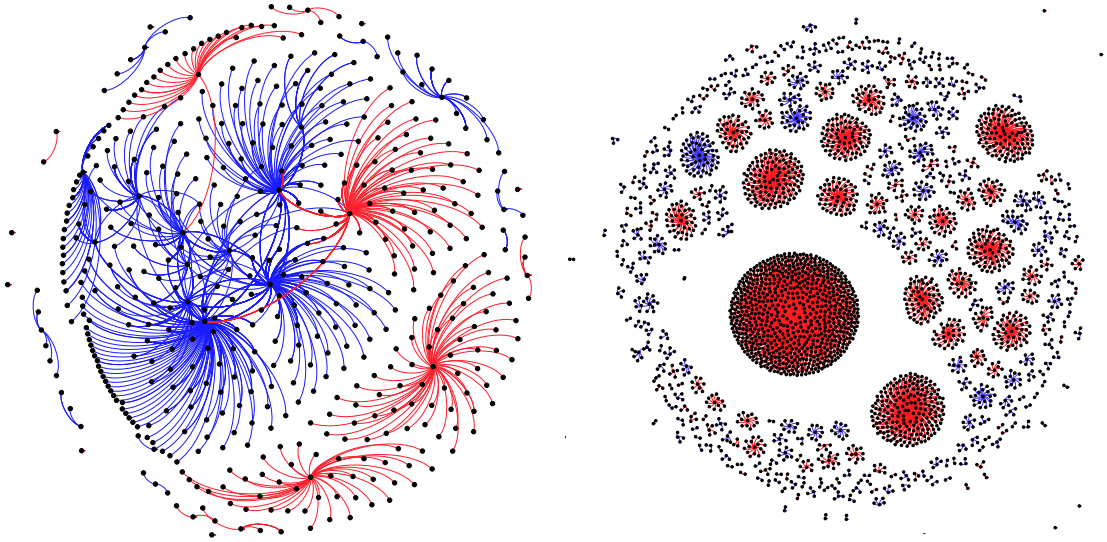
(a) 100k bots and humans.

(b) 100k when bots disappear.

Figure 4.10: Bots *vs.* Humans - graphs for retweets and quotes of 100k popularity group. Black dots are vertices, edges represent a *interaction*. Red edges represent bots and Blue edges represent humans.

Figure 4.10 presents the influence graph from the 100k vertices for retweets and quotes; it exhibits profound differences to the 10M graphs. Inspection reveals that bots are holding the social graph together as they form the medium that connects vertices on the edge of the network. The effects are apparent in Figure 4.10b, which plots the same graph with all bots removed. This indicates that the human part of the 100k retweet graph is only loosely connected, *i.e.* bots play a significant role in influencing and consequently propagating content between humans. Though there are small human communities that seem to be tightly connected, the number of weakly connected components are much higher than strongly connected components.

I also look at replies and mentions for 10M and 100k groups in Figure 4.11, which exhibits substantially different trends to the retweet graph. The density of edges (due to replies and mentions) for both bots (Red) and humans (Blue) shows



(a) 10M bots and humans.

(b) 100k bots and humans.

Figure 4.11: Bots *vs.* Humans - graphs for replies and mentions of 10M and 100k popularity groups. Black dots are vertices, edges represent an *interaction*. Red edges represent bots and Blue edges represent humans.

a range of homophily and interconnectedness between bots and humans. The interconnectedness between bots and humans for 10M and 100k groups ranges from low to very low, respectively. The average degree of interconnectedness in 10M group is 15.4 edges, whereas in 100k group it is 2.7 edges. This observation highlights two important trends within this dataset: (*i*) since replies and mentions are direct one-to-one interactions, strong assortative behaviour is observed in both bots and humans, (*ii*) humans intra-connect more often than bots in 10M group, whereas the trends for 100k group are the exact opposite. This is partly driven by the propensity for automated bots to generate unsophisticated automated responses (*e.g.* spam). It is likely that suspecting humans do not respond to these direct messages by bots, especially those that seem automated or employ *astroturfing*. It is equally likely that naive or simplistic bots are not capable of responding to or engaging in direct messages by unwary humans.

4.5 Takeaways

Bots exercise a profound impact on Twitter. This chapter confirms a number of noteworthy trends: (*i*) bots generally retweet more often, while some humans can exhibit bot-like activity ($\mathbf{G_{10M+}}$); (*ii*) bots can post up to 5× more URLs in their tweets (§ 4.3.1); (*iii*) bots can upload 10× more content with their tweets; (*iv*) humans can receive as much as 27× more likes and 24× more retweets as bots (§ 4.3.2); (*v*) bots retweeting other bots is over 3× more regular than bots retweeting humans, whereas humans retweeting other humans is over 2× greater, indicating homophily (§ 4.3.2); (*vi*) humans favourite others' tweets much more often than bots do, though *newer* bots are far more aggressive in favouriting tweets to replicate human behaviour (§ 4.3.3); (*vii*) humans enjoy higher levels of friendship and usually form reciprocal relationships (§ 4.3.4); (*viii*) bots typically use many different sources for active participation on Twitter (up to 50 or more); and (*ix*) activity sources include basic automation and scheduling services (§ 4.3.5) — used abundantly by bots and seldomly by humans. These findings have been summarised in Table 4.2.

Table 4.2: Feature inclination: $\mathcal{B}$ is more indicative of bots, whereas $\mathcal{H}$ is more indicative of human behaviour, and ◯ is neutral (*i.e.* both exhibit similar behaviour). * represents magnitude of inclination: * is considerable difference, ** is large difference. *signif.* shows statistical significance of each feature as measured by *t-test*.

Feature & value	**Fig.**	**10M+**	**1M**	**100K**	**1K**	***signif.***
More user tweets	4.2a	◯	$\mathcal{B}$*	$\mathcal{B}$*	$\mathcal{B}$*	
Higher user retweets	4.2b	$\mathcal{H}$*	$\mathcal{B}$*	$\mathcal{B}$*	$\mathcal{B}$*	99%
More user replies and mentions	4.2c	◯	$\mathcal{B}$*	$\mathcal{B}$*	$\mathcal{B}$	99%
More URLs in tweets	4.6a	$\mathcal{B}$**	$\mathcal{B}$**	$\mathcal{B}$**	$\mathcal{B}$**	99%
More total content uploaded (KByte)	4.6b	$\mathcal{B}$**	$\mathcal{B}$**	$\mathcal{B}$**	$\mathcal{B}$**	95%
Higher likes received per tweet	4.3a	$\mathcal{H}$**	$\mathcal{H}$**	$\mathcal{H}$**	$\mathcal{B}$	99%
Higher retweets received per tweet	4.3b	$\mathcal{H}$**	$\mathcal{H}$**	$\mathcal{H}$**	$\mathcal{B}$	99%
More tweets favourited (liked)	4.4a	$\mathcal{B}$**	$\mathcal{H}$**	$\mathcal{H}$**	$\mathcal{H}$**	99%
More favourites by *younger* accounts	4.4b	$\mathcal{B}$	$\mathcal{H}$	$\mathcal{B}$	$\mathcal{B}$	
Higher follower-friend ratio	4.2d	$\mathcal{B}$**	$\mathcal{B}$*	$\mathcal{B}$*	$\mathcal{B}$**	
More activity sources	4.5a	$\mathcal{B}$*	$\mathcal{B}$	$\mathcal{B}$	$\mathcal{B}$	99%

I have also shown that bots inject significant proportions of network traffic via the uploading of media (§ 4.3.6). I also found that there were clear differences between the URLs and content posted by bots *vs.* humans. By regularly posting links, I posit that bots trigger further traffic generation amongst their followers. I therefore allude that Twitter, and similar services, should begin to explicitly

factor this within their infrastructural design. Such bots, for example, could be downgraded in terms of Quality of Service priorities, or even have their uploads buffered/delayed until off-peak hours.

In this chapter I performed a measurement study that encompassed feature extraction, an in-depth analysis for differentiating bots from humans, and distinguishing their activities and impact on Twitter. I conclude this chapter by saying that bots have an existential impact on social media, and I believe understanding their activities has inherent scientific value. The scale of their role within Twitter is equal to that of humans and, as such, this Chapter was intended to pave way for a reliable bot detection tool (Chapter 5).

Chapter 5

Detecting Social bots

Chapter 4 utilised ***Stweeler*** to collect a large Twitter dataset, extracted and studied features in-depth to acquire a wide array of attributes that distinguish bots from humans. In this chapter I present a methodology and implement a model for *non-partisan* classification of Twitter users into bots and human users, by refining preprocessing and partitioning of datasets, creating and using a large human annotated dataset as ground truth labels, as well as extracting most relevant feature-sets (via ablation tests) for each popularity group.

To perform accurate classification I use partitioned human annotated dataset (§ 3.4.1–3.4.3) that compensates the differences present due to account popularity. To judge accuracy of the procedure I calculate agreement among human annotators as well as with a bot detection research tool. Treating account categorisation on Twitter as a binary classification problem, I then apply a Random Forests classifier on the dataset. By performing ablation tests I identify most insightful feature-sets for each popularity group. I then apply a Random Forests classifier that achieves an accuracy close to human agreement. Finally, as a concluding step I perform tests to measure the efficacy of the results.

5.1 Introduction

The existence of bots is making a real impact on our daily lives. For instance, Facebook employed automated techniques[1] to populate, curate and tweak its

[1]Facebook trending news module (last accessed 16 June 2018) – `https://www.theguardian.com/technology/2016/aug/29/facebook-fires-trending-topics-team-algorithm`

trending news module which led to disastrous results. The algorithm started populating the trending news feed with false and controversial stories that pushed the questionable content even further. Microsoft's Tay was a bot operating a Twitter account learning to mimic human speech patterns by interacting with other users through tweets and replies. The experiment had to be terminated[2] when Tay was taught hate-speech and racism. This highlights that automated conversation and content dissemination may take an unexpected turn that the users may find offensive and harmful. Recently, an MIT scientist programmed a Twitter bot[3] that tweets like the US president Donald Trump. The bot uses an AI algorithm to learn Trump's style of speech by going through debate transcripts. This exemplifies the other side of the coin – the recent research trend of automating content generation and mimicking people on Twitter.

Contributions of this chapter: The goal of this chapter is to classify Twitter users as bots (that tweet via a scheduling tool or an automated program that uses Twitter API) and human users. This chapter focuses on the following: (*i*) Use of raw historical data (60 million tweets) for attribute collection and account classification (722,109 tweets) to cater for stealthier bots that are harder to discern from humans; (*ii*) A Twitter dataset divided into user popularity groups, further partitioned into lists of bots and humans (for reasons refer to § 5.2) using a human annotation task. This serves as a large ground truth dataset; (*iii*) 14 novel features from a total feature-set of 22 attributes (see § 5.2); (*iv*) Performance evaluation of current state of the art in bot detection by calculating agreement between human annotators and BotOrNot; (*v*) Application of supervised learning approach – Random Forests classifier – for *non-partisan* account categorisation; (*vi*) Identification of a distinct group of features (using ablation tests) that are most informative for classifying bots within each popularity group (see Table 5.7); and (*vii*) Hypotheses (see Table 3.1) verification against my findings using t-tests (see § 5.4).

An implemented research tool that offers an API is BotOrNot [22, 83], that uses six feature-sets and a Random Forests classifier to output bot-likelihood score of a given Twitter account. I carry out a well-defined human annotation task (see

[2]Microsoft's Tay (last accessed 16 June 2018) – `https://www.theguardian.com/technology/2016/mar/24/microsoft-scrambles-limit-pr-damage-over-abusive-ai-bot-tay`

[3]DeepDrumpf (last accessed 16 June 2018) – `http://uk.businessinsider.com/how-donald-trump-talks-2016-9`

§ 5.2) and compare these to the BotOrNot annotations. In the experiments, I have found that BotOrNot produces an average agreement of 48% with human annotators, while the average agreement among human annotators is 89%.

5.2 Methodology

A tweet object[4] is formed of attributes written in JSON structure. ***Stweeler***[5] platform (Chapter 3) is used for collecting data, defining partitions, filtering data, calculating feature values and various other preprocessing tasks. This chapter extends ***Stweeler*** by designing a classification tool for bot detection. Full details about the partitioned dataset can be found in § 3.4.1. Features I consider in this study are defined in Table 3.1, and their details are explained in § 3.4.2. Details about the annotations of the partitioned dataset can be found in § 3.4.3. The annotated partitioned dataset is explored in detail in Chapter 4.

Hardly any past work objectively compares other detection or classification tools to their experiments. I use BotOrNot[6] HTTP REST API, which returns a bot-likelihood score for each Twitter account. BotOrNot does not assign labels as 'bot' or 'human', but a 50% threshold (as mentioned on BotOrNot website and confirmed from author publications) is set as the boundary between an account being a human account (*i.e.* $< 50\%$ likelihood) and an account being a bot account ($>= 50\%$ likelihood). I choose 50% threshold in this chapter as logically indicated by BotOrNot authors. Furthermore, the accuracy of BotOrNot across a variable threshold range (40% to 60%) proved to be similar to 50% threshold. Whenever BotOrNot returns a bot-likelihood score of less than 50% the account is labelled as 'human', otherwise assigned a 'bot' label.

The assumption is that the human annotation task produces a dataset annotated with the labels that are the closest approximations of the "ground truth" labels, since the latter are, in general, unavailable (see the discussion in § 5.3). Furthermore, I use the agreement between the human annotators to benchmark the performance of the automated bot classification system.

I then calculate statistics for various features listed in Table 3.1, and use a

[4]Twitter Tweet Object (last accessed 16 June 2018) – `https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/tweet-object`

[5]***Stweeler***– `https://github.com/zafargilani/stcs`

[6]BotOrNot is now rebranded as Botometer, last accessed 16 June 2018 at `https://botometer.iuni.iu.edu/`

Random Forests classifier to perform three sets of experiments. First, I run a 5-fold cross-validation experiment in which I use 4 folds to train and 1 fold to test the classifier in each of the runs, with each fold containing subsets of all popularity groups, and report the results averaged across all 5 runs. Second, I report the results on the data originating with each of the popularity groups in particular. Third, I test how generalisable the features are, and for that I train the classifier using sets of 3 popularity groups and test it on 1 remaining popularity group in each of the runs.

I perform ablation tests: starting with the full feature-set and then remove features one by one in order to detect the minimal optimal feature combination that yields the best results on the task. Features that show up most often in the best performing feature splits in these experiments include *followers-to-friends ratio*, *user retweets*, *tweet frequency* and *URLs count*.

Finally, I obtain the classified datasets as well as the best features and their respective feature splits. Results of the annotation task and bot classification are presented in § 5.3 and § 5.4, respectively.

5.3 Human Annotation Task

The annotation task fulfils two goals: first, it is used to derive the ground truth labels for the machine learning experiments presented in § 5.4. The information provided by the Twitter users on their accounts is not a reliable method to discern an account type. Depending on the goals of a Twitter account operated by an bot, it may or may not self-identify as such: *e.g.* if the goal is to spread false information and malicious content, the bot may pretend to be a human.

Second, human annotation task helps estimate how accurately humans can identify bots on Twitter. This provides a very useful point of comparison for the machine learning experiments presented in § 5.4. The ultimate goal of this chapter is to implement an automated tool for bot classification on Twitter that would perform comparably to humans, but it might be unrealistic to expect it to outperform humans. I will therefore compare the performance of the classifier presented in § 5.4 to the inter-annotator agreement.

For details on human annotations see § 3.4.3. Twitter data within each popularity group has been independently annotated by 4 annotators. Each account

is marked as either human or bot, and final ground truth labels are used (in the following machine learning experiments) *iff* majority vote holds between all annotators. This majority vote is the final annotation that is derived from the four annotations. If there is a tie (*i.e.* 2-2 vote split among annotators) it is discussed among the annotators and re-annotated for a majority vote (*i.e.* for final annotation). Table 5.1 reports the average pairwise inter-annotator agreement across all popularity groups. In addition, I report average annotators' agreement with the final annotation, and average agreement of the annotators with the labels assigned by BotOrNot (BON) [22]. The inter-annotator agreement in Table 5.1 is reported on the scale from 0% to 100%, with 0% showing lack of agreement and 100% being perfect agreement.

Table 5.1: Average inter-annotator agreement (%-age).

Ann	$\mathbf{G_{10M+}}$	$\mathbf{G_{1M}}$	$\mathbf{G_{100k}}$	$\mathbf{G_{1k}}$
An_1	94.50	82.14	73.15	91.32
An_2	95.50	79.46	72.02	89.75
An_3	95.50	75.63	68.32	86.87
An_4	90.50	79.69	70.88	90.72
Avg	95.58	80.65	73.00	90.40
Final	**96.00**	**86.32**	**80.66**	**93.35**
BON	46.00	58.58	42.98	44.00

Table 5.2 reports Cohen's *kappa* (κ) coefficient widely used in annotation experiments for assessing how reliable the annotators' judgements are, or determining "the degree, significance, and sampling stability of their agreement" [20]. This coefficient takes into account the observed agreement between the annotators p_o as well as the agreement that is expected by chance p_c, that is estimated by finding the joint probabilities of the marginals. The κ coefficient is calculated as follows:

$$\kappa = \frac{p_o - p_c}{1 - p_c} \tag{5.1}$$

Following interpretation of κ values provided by [56], it was concluded that the annotators in this experiment achieved moderate ($\kappa \in [0.41 - 0.60]$ for $\mathbf{G_{100k}}$) to substantial ($\kappa \in [0.61 - 0.80]$ for $\mathbf{G_{1k}}$ and $\mathbf{G_{1M}}$) to almost perfect ($\kappa \in [0.81 - 0.99]$ for $\mathbf{G_{10M+}}$) agreement which can be considered reliable in all cases. It is also worth noting that agreement of BotOrNot with human annotators ranges from less than chance[7] ($\kappa < 0.00$ for $\mathbf{G_{1k}}$, $\mathbf{G_{100k}}$ and $\mathbf{G_{10M+}}$) to slight ($\kappa \in [0.01 - 0.20]$

[7]Negative κ shows less than chance agreement.

for $\mathbf{G_{1M}}$) agreement only, which shows that human annotators almost always disagree with the labels assigned by BotOrNot. These evaluation results are similar to what is reported by Cresci *et al.* [21].

Table 5.2: Average Cohen's κ.

Ann	$\mathbf{G_{10M+}}$	$\mathbf{G_{1M}}$	$\mathbf{G_{100k}}$	$\mathbf{G_{1k}}$
An_1	89.00	63.26	46.37	81.68
An_2	90.93	57.90	44.21	77.99
An_3	90.93	50.41	36.69	72.17
An_4	80.86	58.03	41.71	80.14
Avg	85.15	60.27	46.05	79.58
Final	**91.96**	**71.76**	**61.28**	**85.91**
BON	-8.69	01.90	-14.46	-14.70

Interestingly, $\mathbf{G_{100k}}$ shows the highest disagreement. Less particular properties within this group make these accounts similar to each other: *e.g.* the annotators reported that a number of accounts within this group seemed to be initially bot-operated but were personalised later as human users started actively using them, and vice versa. Exploring this further I found that in some cases new users initially made use of third-party apps and services such as SocialFlow, Hootsuite and Sprinklr to post pre-written messages. Reasons for using such services vary for transitioning from human-operated to bot-operated and vice versa, *e.g.* scheduling tweets while being away or passively monitoring, acquiring new followers, experimenting or 'trying out' new apps or services and then discontinuing, initially posting manually but then signing up to solely use third-party services to interface with Twitter, *etc.*

Based on the results of the annotation task I conclude that: (*i*) The annotators mostly agree when they assign labels to the Twitter accounts, and the annotation can be considered reliable for all groups. (*ii*) The annotators label 43.13% accounts as bots. (*iii*) BotOrNot does not perform well on the given data and shows considerably large disagreement with human annotators' votes. (*iv*) I set the human annotation-based benchmark for the machine learning experiments reported in § 5.4 at 87.42, or at the average observed agreement of the annotators with the final labels on the whole dataset spanning all four popularity groups.

5.4 Classifying Bots and Humans

I approach bot classification on Twitter as a binary classification task. Previous research [18] distinguished between bots, humans and *cyborgs* – accounts that are partly operated by humans and also include automation, thus having properties of both bots and humans. However, there is a confusion surrounding when is a cyborg a bot-operated human account and when is it a human-operated bot account? This confusion emanates because operational observation of an account leaves traces of activity that point towards both automated and human actions. In this work, I choose to perform binary classification distinguishing between bots and humans only, because accounts that consistently involve automation (*e.g.* automated tweeting) should be characterised as automated accounts. As noted in § 5.1, the primary goal is to present a thorough methodological mechanism that allows identification of Twitter accounts as bots and humans using supervised classification.

I had a number of choices for the classification task, but two obvious ones: Naive Bayes and Random Forests. Naive Bayes is a simple classification technique based on the Bayes' Theorem with the strong assumption that the predictors (or features) are independent. Naive Bayes uses Bayes' theorem to calculate posterior probability[8] $P(c|x)$ (Equation 5.2) from prior probability of class $P(c)$, a likelihood $P(x|c)$ and a prior probability of the predictor $P(x)$.

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)} \tag{5.2}$$

Naive Bayes assumes that every feature is independent of every other feature, therefore properties corresponding to all of these features. *e.g.* tweeting behaviour and URLs in tweets, would independently contribute to the probability that an entity is a 'bot'. Though, easy to build, scales well for large datasets, and having linear processing times, the model suffers from the drawbacks that it is fragile to overfitting, underperforms for numerical data in favour of categorical data[9], and predictions are recommended to be taken as raw estimations.

Given that the dataset is multivariate, both categorical and numerical, these problems need to be mitigated. Random Decision Trees [44], or Random Forests,

[8]Statistical probability that a hypothesis is true (in this case that an entity is a 'bot') calculated in the light of relevant observations (in this case features).

[9]Categorical data represents non-numerical characteristics, such as binary classes.

are an ensemble learning method that operates by constructing a multitude of decision trees and produces a prediction class that receives the majority vote (mathematical mode of the classes). The idea behind Random Forests is to use a number of average predictors to make a strong final prediction. Therefore, Random Forests are influenced by Adaptive Boosting (AdaBoost), which trains a classifier in the form $F_r(x) = \sum_{t=1}^{T} f_t(x)$, where f_t is a weak learner in a setting of T learners. Each weak leaner then produces a hypothesis $h(x_i)$ for each sample i. A weak learner is selected per iteration of t, assigned a coefficient α_t such that the sum training error E_t (Equation 5.3) of the resulting classifier is minimised.

$$E_t = \sum_i E[F_{t-1}(x_i) + \alpha_t h(x_i)] \tag{5.3}$$

Random Forests are composed of tree bagging and manufacturing forests of similar trees. The bagging procedure involves bagging a training set X with Y responses repeatedly for B times to fit trees to training samples. For $b = 1 \ldots B$: (i) n training examples from (X_b, Y_b) are sampled, and (ii) a classification tree $f_b(X_b, Y_b)$ is trained. After training, predictions for test samples X' can be construed by taking the majority vote (mathematical mode) of the classification trees (Equation 5.4).

$$\hat{f} = \frac{1}{B} mode f_b(X') \tag{5.4}$$

While predictions by single trees are sensitive to noise in training samples, the majority vote mitigates this, thus leading to better model performance in terms of accuracy. Furthermore, the larger the training sample the better the prediction, as the bagging procedure is designed to *de-correlate* the decision trees. Additionally, Random Forests are robust against overfitting and gives better accuracy as the sample size increases.

I apply Random Forests classifier implemented using `scikit-learn`[10] [67] toolkit and 100 decision tree estimators. But first let's define the benchmarks against which the automated account classification system is evaluated. The lower bound is set as the majority class distribution in the data, which for all popularity groups is equal to the proportion of accounts that belong to humans. In other words, if the automated account classification system always "guesses" that

[10]scikit-learn toolkit – `http://scikit-learn.org/`

an account belongs to a human, then it will perform at the majority class baseline level. Next, I use the average observed inter-annotator agreement between each of the annotators and the final annotation, which indicates how well humans perform on this task as it may be unrealistic to expect an automated system to outperform humans (see § 5.3). Finally, I also include the average agreement between the annotators and labels assigned by BotOrNot. Table 5.3 reports these estimates for each of the popularity groups as well as the average across all data points in the whole dataset.

Table 5.3: Dataset benchmarks.

Group	Majority baseline	Human agreement	BON
$\mathbf{G_{10M+}}$	52.00	96.00	46.00
$\mathbf{G_{1M}}$	60.50	86.32	58.58
$\mathbf{G_{100k}}$	51.24	80.66	42.98
$\mathbf{G_{1k}}$	61.41	93.35	44.00
Total	**56.28**	**89.08**	**47.89**

In addition to the dataset benchmarks, I also prove that the sample set of annotations are representative of their population. In this validation experiment I take varying size of training data (to train the classifier model) and test it against a validation sample of 100 annotations. The training data is taken from the human annotated dataset (see § 5.3), and ranges from 1,000 to 3,000 randomly selected annotations. The 100 annotations for validation purposes are also taken from the human annotated dataset, and are not repeated in the training data. I carry out two validation experiments: (*i*) randomised lists that do not have repeated data points among the lists, and (*ii*) randomised lists that may have repeated data points among the lists.

Table 5.4: Validation results.

Training sample size	Acc validation exp (*i*) (%)	Acc validation exp (*ii*) (%)
1,000	81	83
1,500	79	79
2,000	72	78
2,500	79	82
3,000	79	80

Table 5.4 shows that the set of annotations obtained from human annotators is indeed sufficient. For all of the training sample sizes tested, the prediction accuracy of the classifier model remains at acceptable levels, ranging between 72% and 83%, and usually remaining at 80%. The classifier model hits a low

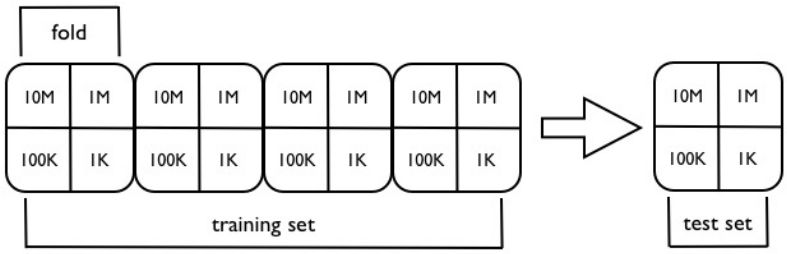


Figure 5.1: Classifying bots by training and testing on all groups with 5-fold cross-validation.

point at 2,000 samples which shows that 2,000 training annotations in validation experiment (i) differed the most from the testing annotations.

Next, I perform three types of machine learning experiments (see § 5.4.1, 5.4.2, and 5.4.3) aimed at detecting how informative and generalisable features, overviewed in § 5.2, are for this task. For each of the experiments, I report accuracy of classification (`Acc`) which shows the proportion of bot and human accounts that the classifier identifies correctly, and precision (`P`), recall (`R`) and $\texttt{F}_1$ measures on the class of bots which show classifier's performance in identifying bots specifically.

5.4.1 Classifying bots by training and testing on all groups with 5-fold cross-validation

In the first experiment, I apply 5-fold cross-validation: I split the data into 5 non-overlapping folds, each containing approximately equal proportion of data points from each of the popularity groups, as well as having similar distribution of human and bot accounts. The classifier is then run over the folds, using each of the 5 folds as a test set once and training the classifier on the other 4 folds for each of the runs. Figure 5.1 illustrates this experiment. The first row (**Total**) of Table 5.5 reports the results obtained with the best-performing feature-sets. This type of test enables determine the general accuracy of the classifier.

Next is to run ablation tests to detect the most optimal feature-set – the minimal feature-set that yields the best accuracy. Ablation tests show that among the total of 22 features that I use in this work 12 features score among the most informative features across all 5 folds in the cross-validation experiment. These include *user replies*, *retweets per tweet*, *tweet frequency*, *age of account*, *followers-to-friends ratio*, *favourites-to-tweet ratio*, *URLs count*, and S_1, S_2, S_3,

S_5, S_0. Note that human annotators also mentioned similar characteristics as strong indicators. A group of 6 other features score well for 4 out of 5 folds. These include *user tweets*, *user retweets*, *user favourites*, *likes/favourites per tweet*, *lists per user*, and S_4. Based on these results and in conjunction with Chapter 4, I conclude that features that represent content propagation (frequently tweeting, retweeting, posting URLs with tweets) and user engagement (following, receiving likes, receiving retweets, subscribing to lists) are overall the strongest predictors of automation.

Interestingly, *activity source* count and *CDN content size* considered in this experiment do not score as frequently among the most discriminative features on the data that combines all popularity groups. The annotators noted that the use of the Twitter API or automated activity source was a strong indicator of automated behaviour on Twitter. This is confirmed by the nature or type of the activity sources (S_1 = browser, S_2 = mobile apps, S_3 = management, S_5 = marketing, and S_0 = all other services), all of which are strong indicators of automation.

5.4.2 Classifying bots by training on all and testing on specific groups with 5-fold cross-validation

In the second experiment, I train my classifier using the same 5 training folds containing data from all popularity groups, but report the results and run the ablation tests on the subsets of the test data that belong to each of the 4 popularity groups separately. Figure 5.2 describes the design of this experiment. In essence, the classifier is trained on the features that describe accounts from all 4 groups, but is then applied to the test data from one particular popularity group.[11] This experiment helps discriminate between the results obtained on the data points originating within different popularity groups. Table 5.5 reports the results.

Note that the performance follows similar trends as I report for the human annotation experiments (see Table 5.1 and Table 5.2): the classifier performs the best on $\mathbf{G_{10M+}}$ and the worst on $\mathbf{G_{100k}}$, whereas I also noted that human annotators reach highest agreement on $\mathbf{G_{10M+}}$ and lowest on $\mathbf{G_{100k}}$. Interestingly, when

[11]Note that the data in the training and test sets is non-overlapping as before: *i.e.* each of the 5 test folds contains a different 20% of the data, with the rest being used for training.

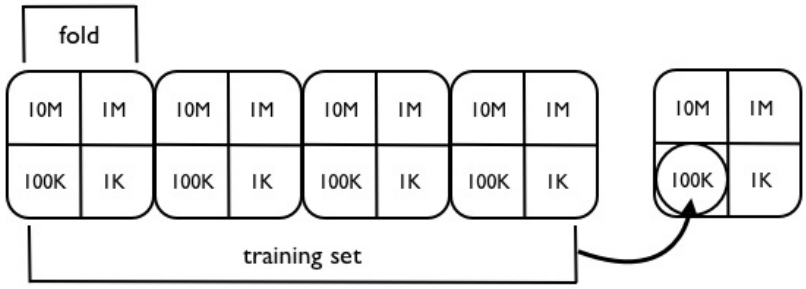


Figure 5.2: Classifying bots by training on all and testing on specific groups with 5-fold cross-validation.

Table 5.5: Machine learning experiments results.

Group	Acc	$\mathbf{P}_{bots}$	$\mathbf{R}_{bots}$	$\mathbf{F1}_{bots}$
Total	86.44	85.40	82.20	83.60
$\mathbf{G_{10M+}}$	100.00	100.00	100.00	100.00
$\mathbf{G_{1M}}$	91.76	90.60	88.00	89.40
$\mathbf{G_{100k}}$	85.70	85.60	85.40	85.60
$\mathbf{G_{1k}}$	88.25	87.80	80.80	84.00

I train the classifier on the data from all popularity groups and measure its performance on specific groups, the classifier's accuracy on $\mathbf{G_{10M+}}$, $\mathbf{G_{1M}}$ and $\mathbf{G_{100k}}$ is above human agreement, and closely approaches human agreement on $\mathbf{G_{1k}}$ (see Table 5.5 and Table 5.3). The most informative features include *retweets per tweet*, *lists per user*, *tweet frequency*, *CDN content size*, and S_2, S_4. Note that features such as *age of account*, *follower-to-friend ratio*, *favourites-to-tweet ratio*, and *URLs count* were informative when data is combined from all popularity groups, but are not discriminative when popularity groups are looked at separately. On the contrary, features such as *lists per user*, *CDN content size* and S_4 = automation services, were not informative for combined data but are discriminative upon observing popularity groups separately.

5.4.3 Cross-group experiments

Next I test how well the system generalises across the popularity groups with respect to the features used. For that, for each popularity group I train the classifier on the data from other 3 popularity groups and apply it to the particular group (see Figure 5.3). The experimental design is described in Figure 5.3, and the results are reported in Table 5.6. Precision, *i.e.* how many selected samples

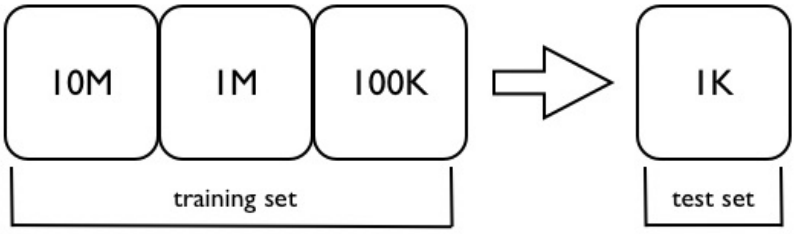


Figure 5.3: Cross-group experiments.

are relevant (usefulness) and Recall, *i.e.* how many relevant samples are selected (completeness), are computed as listed[12]. Similarly, F1 scores[13], *i.e.* harmonic mean of the Precision and Recall scores, are computed to test the accuracy of the test (in this case prediction).

Table 5.6: Cross-group experiments results.

Group	**Acc**	$\mathbf{P}_{bots}$	$\mathbf{R}_{bots}$	$\mathbf{F1}_{bots}$
$\mathbf{G_{10M+}}$	90.00	83.00	100.00	91.00
$\mathbf{G_{1M}}$	86.73	83.00	82.00	83.00
$\mathbf{G_{100k}}$	81.65	82.00	80.00	81.00
$\mathbf{G_{1k}}$	84.17	87.00	70.00	77.00

Note that the classifier performance is consistently high for all groups, reaching the highest for $\mathbf{G_{10M+}}$. This effect might also be due to the size of the training and test sets: the ratio is the highest for $\mathbf{G_{10M+}}$ with 3,486 training and 50 test cases, and the lowest for $\mathbf{G_{100k}}$ with 2,089 training and 1,447 test cases. Nevertheless, note that the performance on all groups is stable, with the accuracy being significantly above the majority class baseline as well as BotOrNot performance (see Table 5.3).

Also note the effect of the training data size on generalisability of the feature-set itself: the largest training set for $\mathbf{G_{10M+}}$ allows the classifier to achieve an accuracy of 90.00% using only 7 features (*user replies*, *follower-to-friend ratio*, *tweet frequency*, *favourites-to-tweet ratio*, and S_4 = automation services, S_5 = marketing, S_6 = news content web services), while the smallest training set for $\mathbf{G_{100k}}$ allows the classifier to achieve an accuracy of 81.65% relying on 16 out of the total of 22 features. The features that are most informative across all the groups include *age of account*, *user replies*, *retweets per tweet*, *tweet frequency*,

[12]Precision and Recall – `https://en.wikipedia.org/wiki/Precision_and_recall`
[13]F1 score – `https://en.wikipedia.org/wiki/F1_score`

favourites-to-tweets ratio, and S_4 = automation services, S_5 = marketing, S_6 = news content web services. It is concluded that this set represents the most generalisable features that are quite independent of the type of account (*i.e.* popularity level). Also note that these features are in general consistent with the features that score well in other experiments, as well as the account properties that human annotators considered important when making their decisions (see § 5.3).

5.4.4 Hypotheses testing

Finally, I check and report whether the features used in this work comply with my original hypotheses. For instance, I had expected that bots tweet more aggressively than humans do and, thus, an average tweet frequency should be significantly higher for bot accounts than for human ones. In the last set of experiments, I apply t-test to the features for the humans and bots within each group and report: (i) whether the difference is statistically significant, and (ii) whether it supports my original hypotheses in terms of the sign of the difference between the means.

Table 5.7 reports the results: I use + where the values for bot accounts are higher than those for human accounts, and - when human accounts have higher values; ** denotes statistical significance at 99% confidence level and * at 95% confidence level.

Table 5.7: Feature significance.

Feature	**10M**	**1M**	**100K**	**1K**	**All**
Age of account	$+^{**}$	+	-	$-^{**}$	-
Favourites-to-tweets ratio	$-^{*}$	+	-	$-^{**}$	-
Lists per user	$-^{*}$	$+^{**}$	$+^{*}$	$+^{**}$	-
Followers-to-friends ratio	+	+	-	$+^{**}$	+
User favourites	+	-	$-^{**}$	-	$-^{**}$
Likes/favourites per tweet	$-^{**}$	N/A	N/A	N/A	$-^{**}$
Retweets per tweet	$-^{**}$	N/A	N/A	N/A	$-^{**}$
User replies	-	+	+	+	$+^{**}$
User tweets	-	$+^{*}$	$+^{**}$	$+^{**}$	+
User retweets	-	$+^{**}$	$+^{**}$	$+^{**}$	$+^{**}$
Tweet frequency	+	$+^{**}$	$+^{**}$	$+^{**}$	$+^{**}$
URLs count	+	+	$+^{**}$	$+^{**}$	$+^{**}$
S_1 = browser	+	+	-	-	-
S_2 = mobile apps	$-^{**}$	$-^{**}$	$-^{**}$	$-^{**}$	$-^{**}$
S_3 = OSN management	$+^{*}$	$+^{**}$	-	-	$+^{**}$
S_4 = automation	$+^{**}$	$+^{**}$	$+^{**}$	$+^{**}$	$+^{**}$
S_5 = marketing	$+^{*}$	$+^{*}$	$+^{**}$	$+^{**}$	$+^{**}$
S_6 = news content	$+^{*}$	+	+	N/A	$+^{*}$
S_0 = all other	$+^{*}$	$+^{**}$	$+^{**}$	$+^{**}$	$+^{**}$
Source count	$+^{**}$	$+^{**}$	$+^{**}$	$+^{**}$	$+^{**}$
CDN content size	+	+	$+^{**}$	$+^{**}$	$+^{*}$

Note that these results are generally in accordance with the assumptions and also corroborate annotators' feedback as well as classification results: *e.g. tweet frequency*, S_2 = mobile apps, S_4 = automation services, S_5 = marketing, S_0 = all other services, and *source count* show the highest statistical significance overall. To summarise, there are several trends worth noting:

- *Age of account* is a good predictor at the extreme ends of the popularity groups. At the same time, within the high popularity groups the bot accounts (*e.g.* those of news agencies) are significantly older than human accounts (*e.g.* those of celebrities). At the lower popularity levels, the difference is exactly the opposite, with the human accounts being significantly older than bot accounts.

- Humans in the high popularity $\mathbf{G_{10M+}}$ follow significantly more lists than bots, while within the other groups bots join significantly more lists.

- Humans in the high popularity $\mathbf{G_{10M+}}$ post more replies, and also tweet and retweet more than bots. Within the other popularity groups the trends change to exactly the opposite.

- The *number of URLs* posted, as well as the *CDN content size*, are higher for bots across all popularity groups, but the difference becomes statistically significant for $\mathbf{G_{100k}}$ and $\mathbf{G_{1k}}$.

- S_2 = mobile app usage is significantly higher for humans than bots in all popularity groups.

- Usage of S_4 = automation services, S_5 = marketing and S_0 = all other services is significantly higher for bots than humans in all popularity groups.

- S_3 = OSN management seems to be employed by bots in $\mathbf{G_{10M+}}$ and $\mathbf{G_{1M}}$, while the opposite is true for $\mathbf{G_{100k}}$ and $\mathbf{G_{1k}}$.

- The number of *source count* is significantly higher for bots in all popularity groups. This shows that within $\mathbf{G_{10M+}}$ and $\mathbf{G_{1M}}$ humans post many URLs as well.

5.5 Takeaways

In this chapter I developed and evaluated a thorough mechanism to reliably classify automated bots and human users on Twitter using a dataset divided into four popularity groups. I used a human annotation task to augment and refine the original ground truth labels (Chapter 4), and verify the annotations using inter-annotator agreement among human annotators and BotOrNot (a bot detection research tool). Using a Random Forests classifier I perform three different machine learning experiments. The classifier yields an accuracy that is on a par with human agreement for all four popularity groups. I report on how different feature splits perform for different experiments and noted that 6 features show the highest statistical significance overall.

Human annotation experiment (§ 5.3) shows that people pay attention to the content of the tweets: *e.g.* human annotators cited the style and pattern of the tweets as strong indicators of bot-operated accounts, and also noted that abundance of promotional and depersonalised content strongly suggested that the account was operated by an automated bot. In this chapter, *URLs count* was used as one of the features to analyse the tweet content, with the higher number of URLs suggesting promotional and depersonalised content. To supplement this, it is possible to explore if bots fall into particular topical divisions and exhibit sentiments that are similar to humans (as also suggested in Chapter 4). In Chapter 6 I address the above and explore bot categories by defining a methodology that employs unsupervised learning to define *unlabelled* bot clusters. Next I label these clusters using distinctive features in order to be able to make sense of the analyses that follows. I then focus on content analysis using topic modelling and sentiment analysis to distinguish between various bot categories.

Chapter 6

Typification of Social bots

In Chapter 5 I explored bot detection that employed supervised learning (classification) to discern bots from humans. However, social bots are not unitary. Instead, bots exist in various shapes and forms, and could range from semi-automated to fully automated entities. This chapter utilises work done in Chapters 4–5 to extend ***Stweeler*** (Chapter 3) for a deeper understanding into the bot phenomenon. In order to explore bot categories I extend ***Stweeler*** to design a set of unsupervised machine learning methods. I evaluate models based on their purpose and output to pick and implement the most suitable method for defining *unlabelled* bot clusters. Next, I label these clusters using distinctive features in order to be able to make sense of the analysis that follows. My focus then shifts towards content analysis using topic modelling and sentiment analysis to distinguish between various bot categories. However, Twitter by default does not offer geolocation information (for privacy purposes) or IP addresses (because of being an application layer service). Network level information is necessary to detect bots that exist on the Web but can impact content popularity and activity on Twitter. I setup and use a bot account on Twitter to collect this supplementary dataset to conduct aforementioned analyses. I conclude with compelling evidence that bots exist in diverse forms and shapes, have diverse existence (on Twitter or off it) while maintaining many similarities but also a large array of differences.

6.1 Introduction

Most recent works have tended to focus on identifying bots and studying their role in particular settings, *e.g.* political infiltration. The limited scope of the latter is largely driven by the difficulty of understanding bot behaviour without *a priori* context to explain their actions. This is particularly challenging at scale simply due to the huge diversity of bots: without knowing approximate intentions (*e.g.* supporting a political candidate, promoting a commercial product) it is near-impossible to explain their actions.

The lack of generalisable tools for categorising "types" of bots has led to a range of ad hoc techniques applied in the above studies. Although sometimes effective, this approach has severe implications on reproducibility and, perhaps more importantly, makes the analysis of new datasets extremely difficult (due to the need to develop new methodologies). Hence, I posit that a generalisable and modular methodology is required to allow *any* researcher to easily (*i*) Identify bots within a social media dataset, and (*ii*) Classify them into "types" of bots for further analysis. I aim to deliver this goal while enforcing two constraints: (*i*) using an *unsupervised* learning approach that is flexible and applicable to various datasets, and (*ii*) simplifying and automating the learning process by removing prerequisites such as a human or manual annotation task to label datasets. Unsupervised learning further helps alleviate the issues of subjectivity, misaligned decision boundary, and pre-annotated classifications; problems common in supervised settings.

Contributions of this chapter: With the above goals in mind, I extend ***Stweeler*** (Chapter 3) – a data collection, measurement, feature extraction, bot detection and analysis framework. To explore bot categories I begin by performing a large-scale measurement and analysis campaign on Twitter (§ 6.2) via ***Stweeler***. Using the ***Stweeler*** bot classifier developed in Chapter 5 bots are detected through classification from the datasets. I then decompose the bots into a set of clusters exhibiting similar traits – I term this process "bot typification". To achieve this, I develop an unsupervised clustering task to create *unlabelled* clusters from features (§ 6.3). These clusters are derived from the quantified behavioural and social properties of the accounts, grouping users based on traits such as retweeting rates, number of followers, *etc* (see Table 6.2). Through a series of topical analyses, I then strive to generate labels for these groups based

on the principle components of discussion within each cluster.

Once the clusters have been defined, I then explore their properties — starting by exploring the innate characteristics of the eight clusters identified (§ 6.3.3). A range of behaviours are observed, with three highly populated clusters made up of bot accounts that follow well known promotional strategies. These include favouriting a large number of tweets (for self promotion), whilst receiving little attention in return (*e.g.* receiving few likes). However, I also discover five outlier clusters, with one containing a maximum of 35 accounts. These tend to contain older bots and more popular bot accounts, sometimes even with celebrity status. For example, one cluster (#5) contains bots with an average of 405 likes per tweet compared to just 20 in another cluster (#0). Although intuitive, this empirically confirms that bots are *not* one shade but, instead, highly diverse with various patterns both in terms of their own behaviour and the reactions of others.

Following this characterisation, I then perform an in-depth analysis into several core aspects of bot activity to understand how it varies across the cluster identified (§ 6.4). I start by evaluating the types of software tools used by bots, as identified via the endpoint metadata contained within this Tweet dataset (§ 6.4.1). This reveals a complex picture, where each cluster typically utilises a range of tools. That said, a few major players are identified – software specifically dedicated to tweet generation and management. Curiously, I also observe that less popular accounts tend to use a mix of toolkits and human intervention (*e.g.* web client). This is also mirrored across some more popular clusters, often driven by a few constituent celebrity accounts (*e.g.* alexburnsNYT).

Next, Latent Dirichlet Allocation is used to identify topics of discussion within each cluster (§ 6.4.2). As the unsupervised learning technique *solely* uses quantified metadata for the clustering process, they are formed independent of the tweet content itself. Hence, I discover that the clusters focus on a range of overlapping topics. Through this I label each cluster with a range of tags, particularly Advertisements & Marketing, Daily Affairs & Lifestyle, International Affairs, News, Politics. I further investigate the content of the tweets by inspecting the sentiment and polarity (positive or negative) of language used within each tweet (§ 6.4.3). Although all clusters broadly exhibit positive sentiment (i.e. > 0) and similar variance (0.0255–0.0572), I find a far greater spread of polarity. For example, it is found that one cluster (#5) has very low average polarity (0.0454), *i.e.* neutral content. This is because the cluster predominantly contains mainstream news

and sports outlets, which post both highly positive and negative content. Finally, I inspect the content links that accounts include in their tweet (*i.e.* URLs). Although, I find many examples of mainstream websites (*e.g.* `youtube.com` is the most popular across most clusters), I also observe various other URLs. These are largely dominated by a few accounts that contribute a disproportionately large number of URLs within each cluster. For example, one cluster (#2) contains links to `elevatedfaith.com` 926 times, just from a single account. The method (this chapter), code/tool[1], and processed datasets[2] are available to the research community for further investigation and future research.

6.2 Preliminaries

In order to define and explore bot categories I build upon ***Stweeler*** (Chapter 3) and use it for data collection, pre-processing, feature extraction and classification tasks. In this section ***Stweeler*** is extended to have bot typification capabilities via clustering and topic modelling (§ 6.3).

6.2.1 Data Collection and Pre-Processing

In order to explore characteristics of various bot types, it is necessary to identify bots from human profiles. Detecting bots is important because the presence of human profiles could skew the results due to similarities. The purpose of clustering is to divide a dataset into equal or unequal chunks on the basis of decided and measured criteria. Bot and human accounts from different subsets of data (*e.g.* similarities between $\mathbf{G_{10M+}}$ humans and bots in Chapter 4) might only exhibit minute differences that could alter the boundaries of clusters, thus forming misrepresenting clusters. Moreover, differences between bots and humans could also cause formation of unnecessary and irrelevant categories containing little or no bots.

Therefore, I use the ***Stweeler*** bot classifier[3] designed in Chapter 5 to distinguish bots from humans for the dataset described in § 3.4.4. I collect a dataset

[1] ***Stweeler***– `https://github.com/zafargilani/stcs`

[2] Datasets – `http://www.cl.cam.ac.uk/~szuhg2/data.html`

[3] ***Stweeler*** bot classifier – `https://github.com/zafargilani/stcs/blob/master/lib/classifiers/rfclassifier.py`

for 30 days in December 2016. Reasons why a new dataset is collected (as opposed to Chapters 4–5) as well as the details on this dataset, language detection and translation, can be found in § 3.4.4. I verify my findings from Chapter 4 in § 6.3.3.

6.3 Typifying Bots: A Methodological Approach

The previous section has described a dataset of tweets, annotated with the bot *vs.* human labels for each account. Next, I further breakdown these accounts into finer-grain classifications that augment the bot label with the *type* of bot. Note that it is not necessary to use ***Stweeler*** for identifying bots; my typification methodology works with any other tools that can extract bot accounts.

6.3.1 Typification Methodology

First, it is necessary to extract "groups" of bot accounts that exhibit similar behavioural traits. This poses two challenges: (*i*) identifying features that typify similar types of bots; and (*ii*) clustering such bots together. The former is particularly difficult to do, as it necessitates a formal definition of bot "types". Although feasible, this comes with a few problems. Firstly, to do this manually, *i.e.* via human annotations, restricts the process to a limited dataset and limited 'freshness'. Secondly, it is likely to suffer from high degrees of subjectivity. In order to remove such subjectivity, I employ an unsupervised learning approach, which can then be analysed *post priori*. The other advantage of an unsupervised task is diminished reliance on training datasets, which would be required during a supervised classification task. Furthermore, this approach is modular, thus a learning model can be replaced with another.

This chapter tests three different clustering approaches for the dataset. A set of features (Table 6.1) for all processed bot accounts is given as input to each of the following clustering algorithms. The feature values are normalised and projected to the clustering method which then predicts the data point per cluster, depending on the algorithm criteria. I initially experimented with the *k-means* clustering approach but found it to be limited given that each data-point is assigned to a cluster whose mean has the least squared Euclidean distance. Therefore, k-means does not capture the differences that might occur between

Table 6.1: Features

Feature	**Description**
Age of account	The age of the Twitter account in days.
Favourites-to-tweets ratio	'Favourites' or 'likes' received for all user tweets.
Lists per user	Lists subscribed to.
Followers-to-friends ratio	Relationship reciprocity.
User favourites	Tweets 'favourited' by a user.
Likes/favourites per tweet	'Favourites' received by a user.
Retweets per tweet	'Retweets' received by a user.
User replies	Tweets replied to by a user.
User tweets	User-generated tweets.
User retweets	Retweeting tweets of other users.
Tweet frequency	Daily tweet frequency of a user.
Activity source type	A 'source' is the endpoint from where a user performs activity on Twitter, as identified in Chapter 4. This categorisation is refined as: browser or web client (S_1), mobile device apps (S_2), social media management apps (S_3), social media scheduling and automation (S_4), social media optimisation and intelligent tweeting (S_5), marketing and brand promotion (S_6), and news content web services (S_7).
Source count	The number of the endpoints used.
URLs count	URLs are used to redirect traffic to elsewhere from Twitter platform.
URL & schemes	URL hosts and URI schemes, extracted from the `[text]` tweet attribute.
photos (JPG/JPEG)	A photos is extracted from the URL in `[media_url_https]` attribute.
animated images (GIF)	Though these are animated photos, Twitter saves the first image in the sequence as a photo, and the animated sequence as a video under the `[video_info]` attribute.
videos (MP4)	Video files accompany a photo which is extracted by Twitter from one of the frames of the video. A video is pointed to by the URL in `[video_info][url]` attribute.

data-points in a multimodal (multivariate) setting. This approach was therefore not suitable.

Next, I experimented with the *Gaussian Mixture Model*, which is applied to multimodal (multivariate) datasets. Gaussian Mixtures instead use Mahalanobis distance, which is a quadratic distance as opposed to a straight line in Euclidean distance. There are, however, two issues when using this model. Firstly, the model cannot learn the number of clusters from the dataset; instead, these have to be provided arbitrarily as an input to the model (which is difficult to know *a priori*). Secondly, the model assumes that the dataset consists of normally distributed dense matrices – this requirement was not met within out data. It was concluded that this approach was also not suitable for this dataset.

6.3.2 Spectral Clustering

Considering the failures with k-means and Gaussian Mixtures, I next experimented with the *Spectral clustering* approach (with *k-means* assignments). Spectral clustering has been widely used in the past for segmenting data points from

a noisy background and image segmentation to identify objects. The algorithm processes normally distributed sparse matrices to group bot accounts into n clusters, where n is learned automatically from the data. This makes it more suitable for this particular purpose. Spectral clustering uses a spectrum, or *eigenvalues*[4], of the affinity matrix to project the data into a low dimension space. This low dimension is the eigenvector (spectral) domain where the data points are easily separable through an assignment method, *e.g. k-means*.

Spectral clustering solves the problem on the affinity graph by cutting the graph into n clusters such that the weight of the edges connecting the clusters (inter-connection) is small compared to the weight of the edges connecting objects inside each cluster (intra-connection). The affinity graph G measures the similarity between data points (or computes the distance) with indices i and j such that $G_{ij} \geq 0$. Cutting the affinity graph is adapted from the normalised cuts problem [73]. This in turn means that since an edge connecting two similar objects on the graph is a function of the gradient (*i.e.* distance), similar objects will be kept together.

Thus, having a distance matrix as affinity matrix for which 0 means identical objects, and high values mean dissimilar objects, the problem can be stated as a weighted *k-means kernel problem* (Equation 6.1).

$$max \sum_{r=1}^{k} \omega_r \sum_{x_i, x_j \in C_r} k(x_i, x_j) \tag{6.1}$$

The weight ω_r is the reciprocal of the number of elements in the cluster, and C_r represents normalised coefficients for each data point for each cluster. The problem can then be vectorised (Equation 6.2) as weighted kernel k-means with n points and k clusters.

$$max_G \; trace(G^T G) \tag{6.2}$$

The *k-means* assignments match finer details of the dataset, though could be unstable and hard to reproduce. Despite this disadvantage, the *k-means* produces finer clusters that match the reality, than the *discretise* assignments that is reproducible and creates clusters of even shapes.

[4] An *eigenvalue* is a non-zero value that only scales by the scalar value and does not change direction when a transformation T is applied to it.

Table 6.2: Clusters produced by Spectral clustering, their comparative tendency *vs.* other clusters for distinctive behavioural properties (**bold** and *italic* signify different tendencies), and descriptive labels.

Cluster	Total bots	Tendency	Distinctive feature (mean value)	Descriptive label
0	3,017	*higher*	favourites performed (14,910)	Young producers
		higher	daily favouriting frequency (26)	
		lower	age (1,105)	
		less	likes per tweet received (20)	
		less	source types used (3)	
		less	URLs posted (37)	
1	1,151	*higher*	favourites performed (11,458)	Young assistants
		higher	daily favouriting frequency (20)	
		lower	age (1,334)	
		less	source types used (4)	
2	809	*higher*	favourites performed (14,600)	Assistants
3	20	**more**	retweets per tweet received (320)	Popular content producers
4	23	**less**	retweets posted (8)	Popular content redirectors
		higher	lists-age ratio (23,043)	
		more	URLs posted (300)	
5	25	*higher*	age (2,357)	Stellar active engagers
		more	tweets posted (1,711)	
		more	replies and mentions posted (404)	
		more	likes per tweet received (405)	
		higher	follower-friend ratio (44,757)	
		more	source types used (19)	
		more	URLs posted (1,151)	
6	35	**more**	retweets posted (60)	Stellar passive engagers
		more	likes per tweet received (661)	
		more	retweets per tweet received (526)	
		higher	follower-friend ratio (33,120)	
		more	source types used (11)	
		more	URLs posted (351)	
7	8	**more**	source types used (12)	Social chameleons

I used Spectral clustering implementation from the `scikit-learn` [67] machine learning library to identify the unlabelled bot clusters. I also identified nine principal components from a list of 24 features (see Table 6.1) that cluster similar accounts together. These include account age, favourites performed, retweets per tweet ratio, follower-friend ratio, number of activity source types used, activity source type, URLs posted as part of tweets, likes received, and retweets received. Note that activity source type is a collection of 7 sub-features (more on that in § 6.4.1). More about feature extraction and exploration can be found in Chapter 3. Findings in Chapters 4–5 helped in refining this list of features (Table 6.2) to achieve an accurate clustered dataset.

However, one persistent shortcoming of Twitter data is that I cannot obtain geolocation information, as Twitter (by default) does not geo-annotate tweets, nor include an IP address which can be used to determine regionality. This would have provided another dimension of features which could have been used

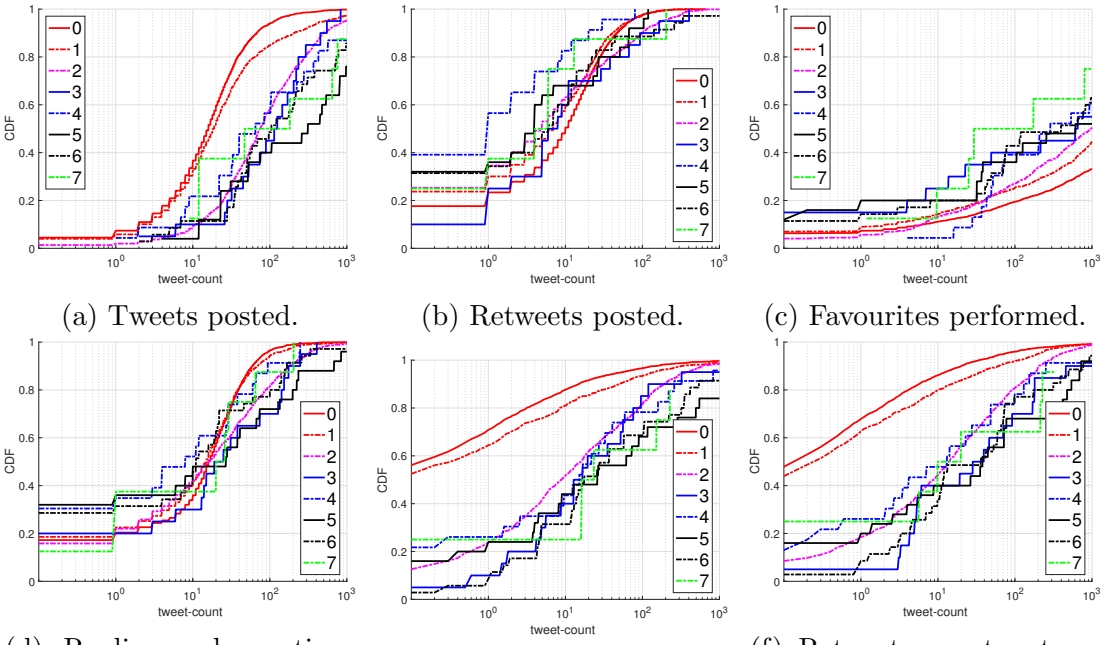
to further refine the clusters, based on account location. However, to experiment with such information I explore other avenues to collect and curate data, such as discussed in § 6.5.

6.3.3 Clustering Results

Table 6.2 presents the clustering results. The process produces eight different clusters which I initially label from 0 to 7. The table lists the number of bots that fall into each group, as well as the characteristics that each group exhibit in regards to features. The characteristics highlighted were identified as the defining factors that resulted in the account being placed in a separate cluster. For example, the largest group is Cluster 0, which tends to contain bots that favourite a large number of tweets, whilst being young, receiving few likes, posting only a few URLs and using just a small number of sources. With these observed characteristics, I then manually label each cluster with a relevant name (see Table 6.2). For instance, in the case of Cluster 0, I term it "Young Producers" as it contains predominantly young accounts that produce a large amount of content. I repeat this for all clusters, selecting names that (in my opinion) best capture their key characteristics. Note that these labels are used for convenience of reference, and do not impact any of the subsequent analysis.

It can be seen that there is high diversity in the cluster sizes. Whereas the majority of accounts are classified as Young Producers, Young Assistants or Assistants, there exists a tail of other accounts that do not have particularly divergent characteristics, *e.g.* Cluster 7 (which is termed as "Social Chameleons"), are bound together exclusively because of number of source types they used. Clusters 3–7 each have 35 or fewer accounts; I find that these clusters tend to contain more "unusual" accounts, which (by definition) have a relatively small number of participants. Most notably, these clusters contain accounts that are both more active and more popular than other clusters. For example, the 25 bots in Cluster 5 post an average of 1,151 URLs compared to just 37 in Cluster 0 (which contains 3,017 bot accounts). Hence, these clusters are of significant interest as they constitute the outliers within my dataset.

To elucidate this, I proceed to explore the exact characteristics of the accounts within each cluster. Figure 6.1–6.2 presents a series of cumulative distribution functions (CDFs) that show the distribution of values across all accounts in each



(a) Tweets posted.

(b) Retweets posted.

(c) Favourites performed.

(d) Replies and mentions posted.

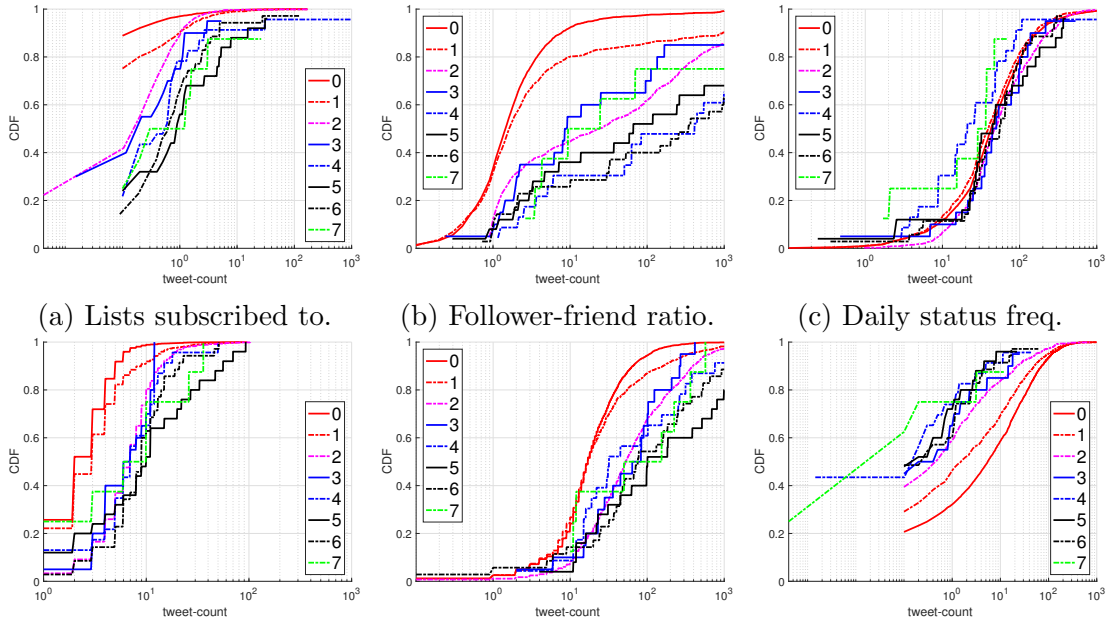
(e) Likes per tweet received.

(f) Retweets per tweet received.

Figure 6.1: Empirical distributions for behavioural activities of bot clusters: 0 (Young producers), 1 (Young assistants), 2 (Assistants), 3 (Popular content producers), 4 (Popular content redirectors), 5 (Stellar active engagers), 6 (Stellar passive engagers), 7 (Social chameleons).

cluster. I present all features considered within the clustering process. Note that Clusters 3–7 have relatively small sample sizes, hence the step-based distributions.

It can be seen that there is a mix of behaviours, with some clustering closely mirroring each other, whilst the remainder diverge significantly. This, for example, can be seen in Figure 6.1a, in which Clusters 0 and 1 generate substantially fewer tweets than other clusters (medians of 32 and 33, respectively vs 65–432). This observation occurs across other features, with Clusters 0 and 1 differing, *e.g.* they tend to favourite more but post fewer tweets. These are what one might term common bots – relatively inactive and unpopular accounts. In contrast, the other clusters exhibit far more unusual characteristics, with high levels of activity across most features. This is most noticeable in terms of tweets, likes, retweets per tweet, follower-friend ratios. The remaining features exhibit roughly equal characteristics across all accounts, with one noticeable difference: favouriting rates. This captures the number of favourites performed by accounts (Figure 6.1c and 6.2f), which Clusters 0 and 1 tend to excel. The median number of



(a) Lists subscribed to. (b) Follower-friend ratio. (c) Daily status freq.

(d) Source count. (e) URL count. (f) Daily favouriting freq.

Figure 6.2: Empirical distributions for behavioural activities of bot clusters: 0 (Young producers), 1 (Young assistants), 2 (Assistants), 3 (Popular content producers), 4 (Popular content redirectors), 5 (Stellar active engagers), 6 (Stellar passive engagers), 7 (Social chameleons).

favourites per day for Cluster 0 and 1 is 4,307 and 1,670, respectively; this can be compared against an overall median of 2,634. This highlights one type of promotion strategy for typical[5] bots, where favourites are used to advertise themselves. Again, I present these distributions to capture the exact characteristics of each cluster, and allow others to contextualise my later analysis. I re-emphasise that using the labels presented (*e.g.* "Young Producers") is a mechanism for discourse, and they do not influence any of the latter analysis.

Before diving deep into the congruent or typical behaviours of each cluster, I verify whether Spectral clustering (*i*) produces representative amount of clusters from the given bot population, and (*ii*) forms same amount of categories rather than new ones. I used two different datasets to find that the same number of categories were formed from both datasets. The first dataset comprised of 9,186 bots from April 2016 and formed a total of eight clusters, although the size of the clusters varied. The second dataset comprised of 5,551 bots from December 2016, that also formed a total of eight clusters. Hence, Spectral clustering proves

[5]Note that 81.92% of all bots in this dataset fall into these two categories.

to be both representative and consistent with the amount of clusters it produces from datasets of similar features.

6.4 Deep Diving into Bot Behaviours

The previous section has presented a methodology to cluster bots into different categories based on various prominent features. Whereas the majority have been clustered into "typical" accounts (*i.e.* those with relatively few followers and low scores across most popularity metrics), I observe a set of outlier clusters containing more unusual bots that exhibit behavioural traits not dissimilar to major human celebrities. This section builds upon these basic characteristics to investigate the deeper behaviour of these bots.

6.4.1 What bot software is used?

I begin by inspecting the bot software used by each account. This is trivial as tweets are accompanied by "source endpoints" which describe the endpoint that created the tweet. Whereas, nearly all (more than 339k tweets, 78.09%) human accounts rely on the official Twitter client (either web or mobile), I observe significant diversity amongst the bot-operated accounts.

To study these, Table 6.3 presents a summary of the different source types I observe, and Figures 6.3 shows the distribution of source type across clusters. It is worth noting that, even though I exclusively include bot accounts, almost 320k tweets (53.83%) from tools involve human usage and intervention (S_1 and S_2), whereas almost 274k tweets (46.17%) are tweeted using automated tools (S_3–S_7). This confirms that many bots are not exclusively automated and, instead, consist of significant human intervention.

In fact, this is further enforced by the human population in the dataset (recall that I detected 11,379 humans as part of ***Stweeler*** bot detection campaign). From the accounts that are detected as humans, approximately 343k tweets (78.90%) of all tweets are generated by tools involving human usage and intervention *vs.* almost 92k tweets (21.10%) by automated tools. This goes a long way in explaining the usual challenges with bot detection – most bots are not exclusively software-based, and most humans are not exclusively using manually operated apps, despite distinctive trends. Inspection of these accounts there-

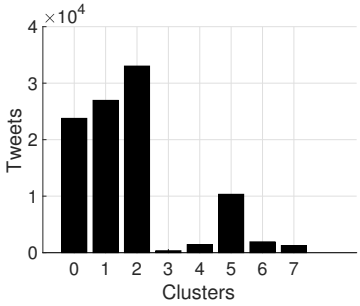
Table 6.3: Types of most prevalent Twitter activity sources for bot clusters.

Source type	**Tool/App**	**Usage description**	**# Tweets**
S_1: Browser or Web client	Twitter Web Client	Human intervention.	98,991
S_2: Mobile device apps	Twitter for iPhone, Twitter for Android, Mobile Web, Facebook, Drudge	Human intervention.	220,176
S_3 Social media management apps	TweetDeck	Social media dashboard management and primitive scheduling.	60,158
S_4: Social media integration, scheduling and automation	Buffer, Hootsuite, SocialOomph, Echobox Social, Postcron, dlvr.it, twittbot.net	Social media integration (Twitter, Facebook, *etc*) and advanced tweet scheduling and automation.	115,663
S_5: Social media optimisation and intelligent tweeting	SocialFlow	Optimise the delivery of messages on Twitter using the commercial Twitter Firehose API and proprietary link proxy (accumulating click data) for large brands and publishers.	34,418
S_6: Social media marketing, brand promotion and customer experience management and analytics for enterprises and businesses	Sprinklr, Spredfast, Sprout Social	Social media marketing, advertising, content management, community management, collaboration, advocacy, monitoring and analytics tools for large brands and agencies.	24,834
S_7: Content web services	SnappyTV.com, IFTTT, Vine	Applets, video editing (*e.g.* creating highlights), video sharing.	38,640

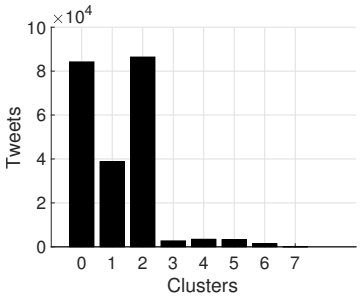
fore reveals a mix of types. Most prominently, I notice that many celebrities (*e.g.* 0220nicole, hughhewitt, saffrontaylor) and organisations (*e.g.* airandspace, TEDTalks, Xbox) with Twitter-facing communications rely on both humans and software to handle significant tweet activity.

As well as revealing human involvement in bot activity, Table 6.3 also presents a number of sources that are automated: S_3–S_7 are all software-based. These include social media integration management and primitive scheduling services (S_3) as well as more advanced tweet scheduling and automation services (S_4). In fact, together S_3 and S_4 form the second largest endpoints for generating tweet activity with almost 176k tweets (29.65%) produced. Beyond these basic tools, I also observe a range of sophisticated and targeted bot platforms. For example, I observe pattern mining bots[6] that learn optimal ways to obtain visibility (S_5), and marketing, monitoring and analytics bots for large brand and enterprises (S_6). The platform provides advertising and marketing products, and monitoring through dashboard services. It is important to note that they account for

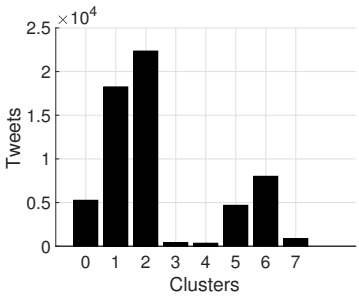
[6]These are based on collecting data from Twitter's commercial Firehose API and accumulating click data through spreading URLs and monitoring clicks.



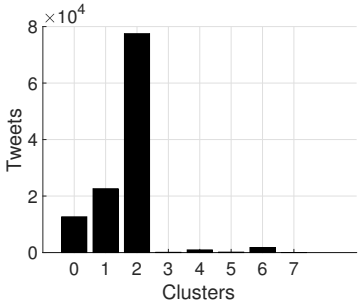
(a) S_1 clusters.



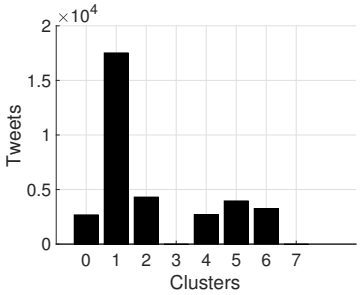
(b) S_2 clusters.



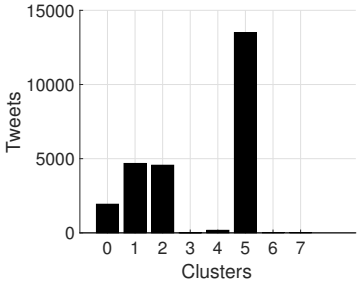
(c) S_3 clusters.



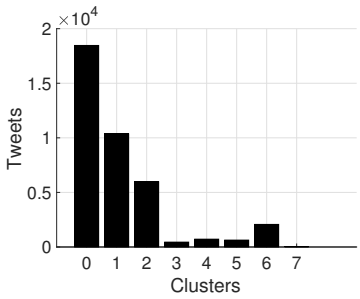
(d) S_4 clusters.



(e) S_5 clusters.



(f) S_6 clusters.



(g) S_7 clusters.

Figure 6.3: Types of most prevalent Twitter activity sources for bot clusters.

less than 60k tweets (10% of my dataset) but are highly optimised: SocialFlow, Sprinklr, Spredfast and Sprout Social (S_6) are specifically designed to optimise tweet activity for large brands (Xbox), agencies (CNN, TIME) and even popular individuals (alexburnsNYT) for maximum visibility and screen time. For example, Xbox retweeted a tweet[7] (originally posted on Friday evening at 2200 hours) every few hours on Saturday to get maximum participants.

The final category of activity source endpoints includes content web services that are purposed to create applets, video editing and sharing on-the-fly by con-

[7]The original tweet can be found here (last accessed 16 June 2018) – `https://twitter.com/xbox/status/809880789437575168`

tent creators (S_7). These services involve a combination of humans (*e.g.* to create highlights of sports events or bulletin news via SnappyTV.com or Vine) and rapid content sharing (*e.g.* through content management and replication such as IFTTT conditional applets). While only around 39k (6.52%) tweets are produced by these services in the dataset, it shows the rapid ability of an information social network to distribute content.

I next inspect how the different clusters exploit each software platform. Figure 6.4 presents the fraction of tweets generated by each source endpoint across the eight clusters. Differences can immediately be seen across the choices made within each cluster. For example, it shows that 100% of tweets injected by Drudge[8] were from accounts in Cluster 1, such as news reporters tweeting for AFP, AJENews, AlArabiya_EGY, AlArabiya, bbcbrasil, FoxSports_br, *etc*; representatives from ELLEfashion; staff from DunkinDonuts, HarvardHealth, *etc*; individuals BobVila, jimcramer, *etc*; and the app itself DRUDGE_REPORT. It is also noticeable that clusters 0 (Young producers), 1 (Young assistants) and 2 (Assistants) use most of the available activity sources, that range from human usage and intervention (left hand side) to completely automated services (right hand side). While clusters 3 (Popular content producers) and 4 (Popular content redirectors) show considerable human usage *vs.* automation, clusters 5 (Stellar active engagers), 6 (Stellar passive engagers) and 7 (Social chameleons) show much higher automation and scheduling *vs.* negligible human usage. This is understandable since content popularity is directly proportional to content novelty and popular trends, that in turn engages human interest. Most bots lack these properties and thus earn much lower popularity levels than human-created content, as noticed previously in Chapter 4.

6.4.2 What topics do bots discuss?

Spectral clustering (§ 6.3.2) produces groups of accounts that exhibit similar traits. Table 6.1 lists traits that are similar among accounts within the same cluster, *e.g.* aggressive tweeting patterns. However, this provides little insight into what different types of bots tweet about. Particularly, I am interested in understanding the context of each bot in terms of its purpose and topics of interest.

[8]Drudge (better known as Drudge Report) is a news aggregator service that allows the user to directly tweet the content being viewed/read.

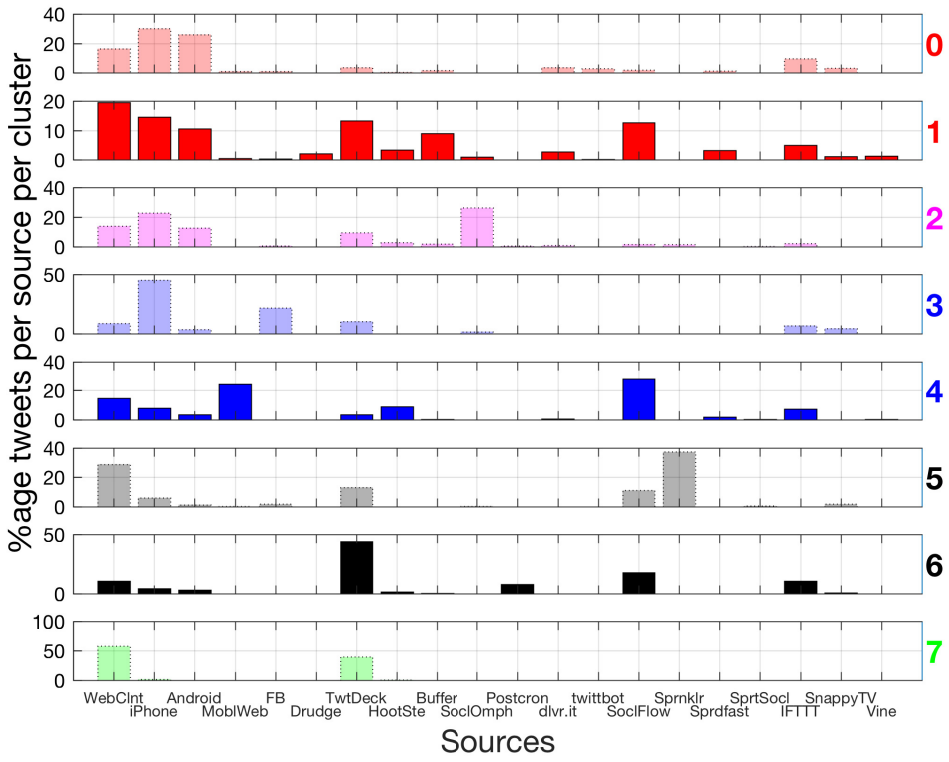


Figure 6.4: Distribution of top 20 activity sources per cluster: percentages are calculated per source per cluster (*i.e.* normalised for different sources in each cluster).

Next, I attempt to explore the topics discussed within each cluster. I hypothesise that certain clusters may have a proclivity towards certain prominent topics. I emphasise, however, that the clusters are derived from the traits listed in Table 6.1, *i.e.* topical similarity was not taken into consideration. Hence, I now explore popular topics discussed within and across clusters.

I start by filtering stop-words and frequently occurring words, such as URL protocol names (to clean the text). I then employ topic-modelling by converting tweets into the most popular topics per bot account. In order to accomplish this I use *Latent Dirichlet Allocation* (LDA). LDA is an unsupervised generative probabilistic model that discovers latent structure in a set of documents by considering each document as a collection of latent topics. Tweets are first broken down into word vectors, and topics are then modelled as a distribution over word co-occurrences. Exact details regarding LDA can be found in [9]. I use the

LDA implementation in `scikit-learn` [67] to generate topic models for the eight clusters.

Figure 6.5–6.6 presents the topic word cloud for each cluster. For the purposes of comparison, Figure 6.7 shows most popular topics and words tweeted by the 11,379 human Twitter users. To give greater context, I perform a manual review exercise to allocate topic labels to these clusters. Topic labels are only generally suggestive and indicative, not decisive. Therefore, I manually label these eight clusters into any combination of Advertisements & Marketing (**A**), Daily Affairs & Lifestyle (**D**), International Affairs (**I**), News (**N**), Politics (**P**), Online Social Networks (**O**), Sports (**S**), and Television (**T**).

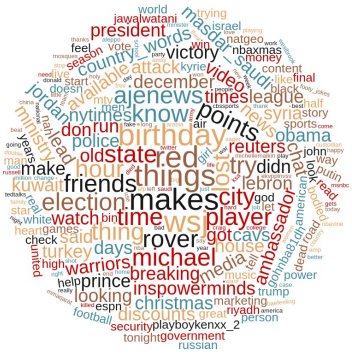
It can be seen that different clusters have a different "skew" towards certain topics. For instance, whereas accounts in Clusters 3–7 (dominos, HPbasketball, RedeGlobo, BBCWorld, MoneyAffairs, BreakingNews, CollingwoodFC, ESPNFC, WDRBNews) have certain very dominant topics of discussion, *e.g.* Basketball, The Economist, Football, *etc*, accounts in Clusters 0–2 (AJArabic, bbcworldfeed, CNNEE, CNNsWorld, NFL, pitchpivot, photo_cj, reddit_top, swissifg, talkvn, teachersdesign, trafficjamnet, whats_live, youkoudan, yalgaarmateen) have a far more egalitarian distribution of topics. This is predominantly driven by the size of these clusters. Whereas Cluster 0 has over 3K accounts, Cluster 7 has just 8 accounts. Despite this, there are clear topics shared across each group, particularly related to politics, *e.g.* US politics. This suggests that each cluster is not dedicated to individual topics but, rather, their behaviour traits are shared across accounts tweeting on a number of issues.

To explore the similarity between the topics, I also compute the topical affinity scores for each cluster against every other cluster. Affinity scores are computed by calculating close matches between pairs of clusters (*e.g.* 0 and 1, 0 and 2, and so on) using Python's `difflib`[9] library. Tiny differences can be observed between same pairs in opposing sequences (*e.g.* 0 and 1, 1 and 0) because the first item of the pair is taken as a base to compare against the second item. When the order of comparison is reversed it changes the comparator cluster (base) and therefore produces the difference in result.

Table 6.4 shows the produced clusters and their affinity scores, where boldface shows the highest topical affinity between two clusters, as well as topic labels per

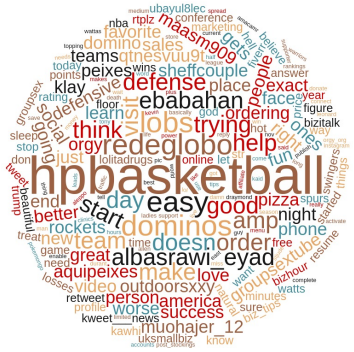
[9]difflib – `https://docs.python.org/2/library/difflib.html`

(a) 0 - Young producers - DNP.



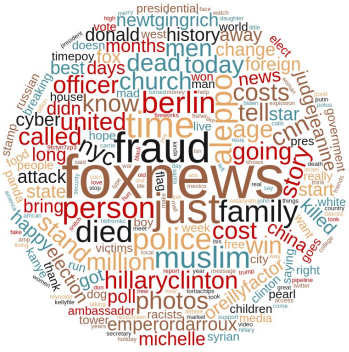
(b) 1 - Young assistants - ANPST.

(c) 2 - Assistants - ADO.



(d) 3 - Popular content producers - DS.

(e) 4 - Popular content redirectors - INP.



(f) 5 - Stellar active engagers - INP.

Figure 6.5: Word Clouds of extracted bot clusters with their statistical labels (Table 6.2) and topic labels: Advertisements & Marketing (**A**), Daily Affairs & Lifestyle (**D**), International Affairs (**I**), News (**N**), Politics (**P**), Online Social Networks (**O**), Sports (**S**), Television (**T**).

cluster. This shows that there is heavy overlap between the topics discussed in different clusters. For the purposes of comparison I also show the affinity scores between the entire human population (11,379 accounts in total) and the eight bot clusters. The bot clusters are strikingly similar to the human population in terms of the popular topics in tweets. The reason of this is that most of the bots are reproducing content which has been posted by humans (either on Twitter or from elsewhere *e.g.* via external URLs). Additionally, this suggests that although there are two very distinct entity populations on Twitter, the topics are highly common among the entities. This strongly indicates that bots are trying to appeal to humans because human action (in the form of a like, retweet, follow, external redirection, influence, bias, manipulation, support, publicity, *etc*) is the end goal

(a) 6 - Stellar passive engagers - ADIT.

(b) 7 - Social chameleons - INPS.

Figure 6.6: Word Clouds of extracted bot clusters with their statistical labels (Table 6.2) and topic labels: Advertisements & Marketing (**A**), Daily Affairs & Lifestyle (**D**), International Affairs (**I**), News (**N**), Politics (**P**), Online Social Networks (**O**), Sports (**S**), Television (**T**).

for most of these entities as noted in Chapter 4.

6.4.3 Do bots exhibit sentiment?

The above has shown that, although clusters tend to have certain dominant topics, there is not a statistically significant trend that *exclusively* limits bots within a cluster to a given set of topics. Next, I expand the content analysis to investigate the sentiments contained within bot tweets. I use the `textblob` API for calculating polarity and subjectivity from all of the text corpora tweeted by the bots in each cluster. Polarity ranges from -1 (negative sentiment) to 1 (positive sentiment), and subjectivity ranges from 0 (very objective) to 1 (very subjective). Table 6.5 shows polarity and subjectivity for the eight bot clusters. I provide both cluster and topic labels.

I observe that Subjectivity scores are fairly even across all clusters (0.4568–0.5386), indicating that all clusters are quite subjective in their generated content, despite the fact that larger clusters have a higher overall variance (*e.g.* Cluster 0 with variance of 0.028 *vs.* Cluster 7 with variance of 0.008, thus a 3.5× difference). Interestingly, Clusters 6 and 7 seem to have mid-range subjectivity, *i.e.* neither completely objective nor subjective. This is owed to two reasons: (*i*) Cluster 6 has only two accounts on either end of the subjectivity spectrum (the very objective primiciasyacom – an Argentinian TV shows portal, and the very subjective

Figure 6.7: Word Cloud of 11,379 human accounts.

Table 6.4: Inter-cluster affinity scores and review labels *vs.* humans. Cluster labels could be any combination of categories: Advertisements & Marketing (**A**), Daily Affairs & Lifestyle (**D**), International Affairs (**I**), News (**N**), Politics (**P**), Online Social Networks (**O**), Sports (**S**), Television (**T**).

Cluster	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0 - Young producers	1	.814	**.782**	**.706**	.728	.796	.682	.784
1 - Young assistants	**.880**	1	.762	.696	.762	**.854**	.700	**.846**
2 - Assistants	.838	.804	1	.674	.736	.744	.656	.782
3 - Popular content producers	.770	.762	.724	1	.712	.712	.656	.746
4 - Popular content redirectors	.800	.788	.694	.696	1	.796	.662	.790
5 - Stellar active engagers	.860	**.840**	.742	.690	**.768**	1	.686	.840
6 - Stellar passive engagers	.810	.744	.710	.608	.748	.784	1	.758
7 - Social chameleons	.846	.772	.742	.668	.710	.818	**.718**	1
Cluster	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Labels	DNP	ANPST	ADO	DS	INP	INP	ADIT	INPS
Humans (pop. 11,379)	**.788**	.746	.746	.712	.684	.752	.662	.752
all clusters *vs.* *Humans*	.730							

VanguardiaSon – Mexican daily information network), while Cluster 7 has none on the ends (rather all between 0.3668–0.6333). This is understandable given the nature of the accounts that mostly relate to Daily Affairs, International Affairs, News, Politics, Sports and Television. However, some particular accounts across all of the other clusters exhibit variance from very objective, *i.e.* 0 (*e.g.* reddotjobs that is operated by reddotjobs.co.uk – a specialist sales recruiter in the UK, or ELLEfashion operated by elle.fr from France tweeting about fashion and products) to very subjective, *i.e.* 1 (*e.g.* DinheiRonaldo that tweeted about Cristiano Ronaldo's net worth roughly 268 times a day from Mar 2015 to Oct 2015, or TheGifLibrary tweeting funny GIFs).

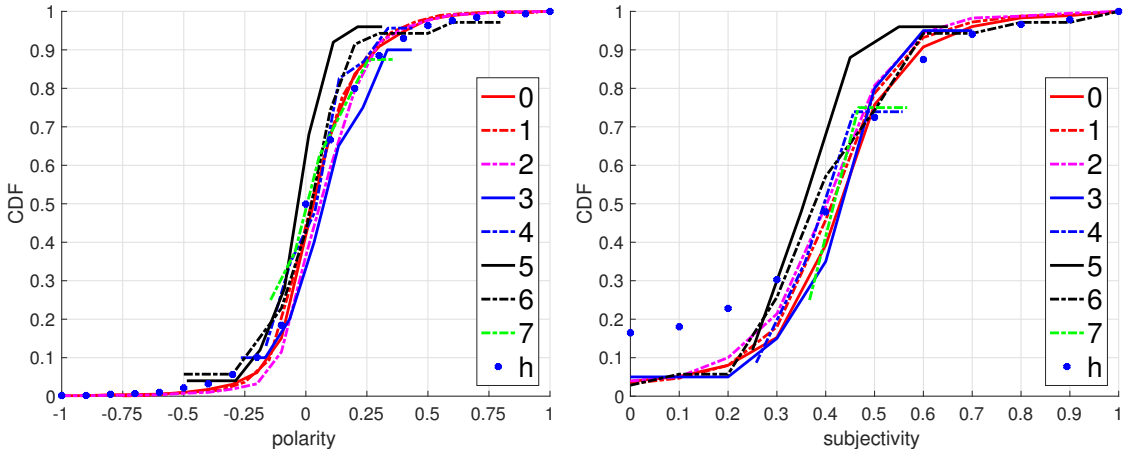
Table 6.5: Average polarity and subjectivity for bot categories and their formulating clusters *vs.* humans.

Bot cluster	**Avg Polarity [-1, 1]**	**Avg Subjectivity [0, 1]**
0 - Young producers - DNP	0.1554	0.5191
1 - Young assistants - ANPST	0.1352	0.4707
2 - Assistants - ADO	0.2059	0.5386
3 - Popular content producers - DS	0.2105	0.5303
4 - Popular content redirectors - INP	0.1310	0.4652
5 - Stellar active engagers - INP	0.0454	0.4568
6 - Stellar passive engagers - ADIT	0.2777	0.5194
7 - Social chameleons - INPS	0.1125	0.4885
Humans (population of 11,379)	0.1266	0.4531

There is a greater spread of sentiment polarity, although *all* clusters broadly exhibit a positive sentiment (*i.e.* > 0) and similar variance (0.0255–0.0572). Quite interestingly, Cluster 5 is the most different overall in terms of polarity, exhibiting low average polarity (0.0454), *i.e.* neutral content. This can be attributed to two reasons: (*i*) most of the accounts in Cluster 5 are operated by (relatively) mainstream news channels (CNN, Fox News, TIME, AlArabiya, MetroTV and NBC's Louisville affiliate wave3news, Q13FOX, franceinter, detikcom), which means these accounts will post content in vast quantities that is both negative and positive; and (*ii*) some of the accounts also belong to sports news (SpheraSports), brands (Starbucks) and Twitteratis running social campaigns (segalink) that will try to post content with positive undertones to keep followers engaged. That said, throughout Clusters 0–7 some particular accounts exhibit variance from very negative sentiment, *i.e.* -1 (*e.g.* CornOppa is a sarcastic account tweeting about topics that typically contain words, such as 'empty' or 'warning', that are usually marked as negative) to very positive sentiment, *i.e.* 1 (*e.g.* LakeNormanRE which is operated by a realty business that tweets listings of attractive properties).

Clinton *vs.* Trump: To ground these results, I next zoom into two pertinent accounts – Hillary Clinton *vs.* Donald Trump – who were being debated in Dec 2016 because of their candidacy in the 2016 US Presidential election. It is now commonly believed that the 2016 US Presidential election was "hacked" through collusion[10] between Trump's campaign team and Russian individuals posing as Americans. In fact, the it has been indicted by the US Department of Justice that the Russian individuals: (*i*) organised and promoted pro-Trump political rallies within the US, (*ii*) posted political messages on social media accounts that

[10]Trump-Russia inquiry indictment (last accessed 16 June 2018) – `http://www.bbc.co.uk/news/world-us-canada-43095881`



(a) Polarity [-1, 1]. (b) Subjectivity [0, 1].

Figure 6.8: Distributions of polarity and subjectivity per bot cluster *vs.* humans.

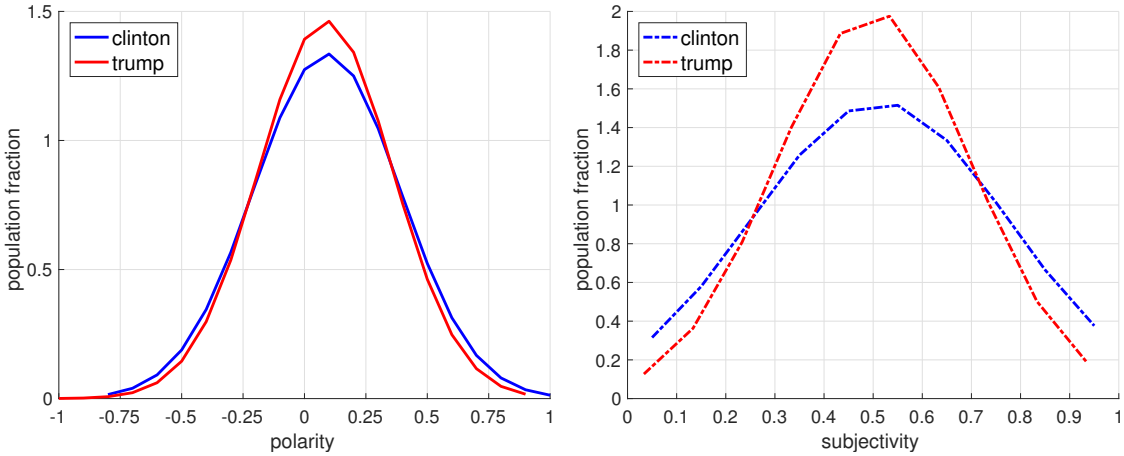
impersonated real US citizens, and (*iii*) promoted information that disparaged Hillary Clinton – the Democrat candidate.

I use the dataset to find if the three of the indictments actually took place, *i.e.* if Donald Trump received more screen time simply because he had received greater promotion, if Donald Trump had received greater social media coverage, and if Hillary Clinton had received infrequent and negative coverage as compared to her Republican rival.

Figure 6.9 presents the distribution of polarity and subjectivity values for all tweets mentioning Clinton or Trump, either as a word, mention or a hashtag. Polarity and subjectivity scores are calculated per account across all clusters, and normalised against total number of tweets posted per account mentioning each topic. Therefore, an account mentioning Clinton in one tweet and Trump in ten tweets will be given normalised weightage. Despite similar distributions, both Clinton and Trump show some differences, such as higher average positive polarity towards Trump, but lower content subjectivity for Clinton (and therefore higher objective argumentation).

However, to find out the sheer volume of traffic produced per topic I look at Table 6.6, which shows polarity scores for Clinton *vs.* Trump tweets. Quite surprisingly, Donald Trump (13,631) received almost 14× more positively inclined tweets than Hillary Clinton (1,005). Even more surprisingly, Hillary Clinton received 796 negative sentiments in tweets than Donald Trump's 538.

To dive deeper I review most renowned news outlets significantly covering



(a) Polarity [-1, 1]. (b) Subjectivity [0, 1].

Figure 6.9: Clinton *vs.* Trump: Normal distributions of polarity and subjectivity.

Table 6.6: Tweet polarity scores for Clinton *vs.* Trump.

+ve Clinton tweets	-ve Clinton tweets	+ve Trump tweets	-ve Trump tweets
1,005	*796*	**13,631**	538

Clinton and Trump during Dec 2016. All of these news outlets are operated through one or more automated sources, with frequent human intervention. Table 6.7 shows the results. At first glance it is obvious that all of the news outlets were providing 6×–42× more coverage to Donald Trump than Hillary Clinton. More surprisingly, most of the news outlets had comparatively more positive coverage towards Donald Trump than Hillary Clinton. In fact, nytimes, Reuters, and TIME were the only news outlets that despite giving Trump more coverage and screen-time, had tweeted more positively towards Clinton. Even more unexpectedly, none of the news outlets had negative sentiment (negative average polarity) towards Trump. This here proves that the three indictments are in fact correct.

6.4.4 What content do bots share?

A major characteristic of bot behaviour is their tendency to share content or redirect traffic to external Web resources via URLs. Whereas the average number of URLs shared by human accounts is 17, it is 22 for bots. In the most extreme case (Cluster 5), the average is 672. This is intuitive as bots are regularly tasked with promoting websites and/or particular viewpoints.

Table 6.7: Polarity scores for Clinton *vs.* Trump by renowned news outlets.

	CNN	**Fox**	**MSNBC**	**nytimes**	**Reuters**	**Economist**	**TIME**	**WSJ**
Clinton tweets	313	277	31	120	12	11	52	10
Clinton polarity	0.0517	*-0.0405*	0.0907	**0.4249**	**0.2**	*-0.2857*	**0.2554**	0.0486
Trump tweets	**1,792**	**3,945**	**331**	**1,730**	**502**	**181**	**567**	**328**
Trump polarity	**0.0773**	**0.1233**	**0.0968**	0.1133	0.1634	**0.1034**	0.1114	**0.1337**

Table 6.8: Shortened URI hosts used for redirection, per bot cluster.

Bot cluster	**URI host**	**# Tweets**
0 - Young producers - DNP	`t.co`	74,583
	`tinyurl.com`	7
1 - Young assistants - ANPST	`t.co`	66,507
	`on.natgeo.com`	5
2 - Assistants - ADO	`t.co`	74,612
	`tinyurl.com`	1
3 - Popular content producers - DS	`t.co`	1,063
4 - Popular content redirectors - INP	`t.co`	4,248
5 - Stellar active engagers - INP	`t.co`	16,804
6 - Stellar passive engagers - ADIT	`t.co`	6,808
7 - Social chameleons - INPS	`t.co`	639
Humans	`t.co`	193,792
	`yfrog.com`	11

I extract all URLs from the bot tweets and find that almost all of the hosts are actually URL shortening services (*e.g.* t.co, tinyurl.com), thus hiding the real URL. Table 6.8 presents the most frequently used URL shorteners for each cluster. Unsurprisingly, the most frequently used URL shortener is Twitter's shortening service `t.co`. The domain `t.co`[11] allows Twitter to automatically shorten a URL whenever a tweet is posted, thus helping Twitter to track and monitor URLs (for spam and malicious content), generate quality signals for insights and conserve the tweet character limit. Little insight can be garnered from this, and therefore I resolve all of the shortened URL to track where they redirect to. Table 6.9 shows the actual URI hosts post-resolution.[12]

Table 6.9 presents a number of popular domains – some well known, others less so. Most prominently, I find YouTube regularly occurring across most clusters. This is particularly the case in Clusters 0, 1, 2, which have large populations with many accounts posting such URLs. I also observe a number of more fringe

[11]Twitter `t.co` (last accessed 16 June 2018) – `https://help.twitter.com/en/using-twitter/url-shortener`

[12]Note that shortened URI hosts and redirected URI hosts are not equatable *i.e.* the sum of shortened URI hosts will not equal the sum of redirected URI hosts because of a number of reasons while parsing the redirected links, such as: suspended URLs, URL resolution expired or deleted, host not found (webpage deleted), *etc.*

URLs being posted, particularly in the smaller clusters. A surprising result is the sheer impact of just a small number of accounts. The nature of the bots means that it is trivial to generate significant numbers of URL tweets, allowing a small number of intense accounts to dominate the cluster. Whereas popular domains (*e.g.* YouTube, Huffington Post) tend to be contributed by many accounts, popular fringe domains are primarily injected by just a few prominent accounts – a clear differentiator from (manual) human behaviour. For example, links to `couponchief.com` were tweeted 595 times in one month (Dec 2016) by just two accounts (Twitter has since flagged it as spam). Although one might imagine more legitimate websites (*e.g.* news) would differ, many other domains are seen achieving high presence through the contributions of just one or two accounts. For example, the second most popular domain in Cluster 1 is `ahmnews.com` with 625 tweets by one account; similarly, in Cluster 4 `reuters.com` is the most popular domain with 30 tweets by one account.

I next zoom into the behaviours of each cluster. I remind the reader that the content of the URLs was *not* used within the initial cluster process. Noticeably different activities are identified with the large (0–2) *vs.* small (3–7) clusters. The large clusters tend to contain a large number of accounts, each generating a relatively small proportion of the URLs. As stated earlier, there is only one commonality shared across most clusters: links to YouTube. In larger clusters, this is driven by a high number of accounts, *e.g.* in Cluster 0, 844 tweets were generated by 72 accounts containing links to YouTube. In contrast, smaller clusters tend to only have a single account that generates a large number of YouTube links. Inspection of the videos reveals that most are music, news, politics, anime and promotional videos (fantasy, religion, ads).

The latter observation generalises across nearly all other domains: their popularity within a cluster is dictated by a tiny number of highly active accounts. This creates an unstable dynamic, where the top domains vary dramatically over time. This is, in part, due to the small population of some clusters, and the extremely aggressive levels of activity seen by a small number of accounts. For example, a single bot (JawalWatani – an Arab news bot with 1.09 million followers) posts 1,337 of 3,105 URLs as part of tweets covering YouTube, Saudi Press Agency, Ahm News and Saudia Today Arabic daily. Similarly, religion is also quite a popular theme in some clusters. For example, elevatedfaith.com (tweeted 926 times by LovLikeJesus from Cluster 2) is a website selling bracelets to promote

Table 6.9: Top most URI hosts post-resolution, per bot cluster (similar URL types are colour-coded), and accounts most typically tweeting a URL (*e.g.* 0_1 is Cluster 0 account 1, and 0_2 is Cluster 0 account 2).

Bot cluster	**URI host**	**URL type**	**# Tweets**	**Accts**
0 - Young producers - DNP	youtube.com	multimedia	844	0_1–0_{72}
	financialsbeat.com	finance	444	0_{73}, 0_{74}
	adnil.site	recruitment	339	0_{75}, 0_{74}
	ryann1200.com	*unknown*	172	0_{55}
	twitter.com	social media	124	0_{76}, 0_{77}
	huffingtonpost.com	news	83	0_{60}, 0_{78}–0_{91}
1 - Young assistants - ANPST	youtube.com	multimedia	716	1_1–1_{26}
	ahmnews.com	news	625	1_4
	hwswworld.com	automation	570	1_{27}
	spa.gov.sa	press	518	1_4
	fenerbahce.org	sports	195	1_{28}
2 - Assistants - ADO	youtube.com	multimedia	1,717	2_1–2_{37}
	elevatedfaith.com	religion	926	2_{38}
	google.co.in	search	769	2_{39}
	couponchief.com	coupons	595	2_{40}, 2_{41}
	amazon.com	e-shopping	258	2_{13}, 2_{33}, 2_{35},
				2_{42}–2_{47}
3 - Popular content producers - DS	youtube.com	multimedia	78	3_1
4 - Popular content redirectors - INP	reuters.com	news	30	4_1
	investors.com	stock market	6	4_1
	hbr.org	business mag	2	4_1
	fortune.com	business mag	1	4_1
5 - Stellar active engagers - INP	moca-news.net	news	293	5_1
	youtube.com	multimedia	38	5_1
	animatetimes.com	*unknown*	35	5_1
	washingtonpost.com	news	2	5_2
6 - Stellar passive engagers - ADIT	politico.com	news	33	6_1
	topstarnews.net	celeb news	22	6_2
	sinembargo.mx	news	12	6_3
	washingtonpost.com	news	10	6_4
Humans	youtube.com	multimedia	2,861	
	90min.com	football	453	
	play.google.com	app store	272	
	prizeo.com	charity	269	
	itunes.apple.com	music store	141	
	facebook.com	OSN	85	

Christianity.

Dynamics are more significant in large clusters, they are even more pronounced in the smaller fringe clusters (4, 5, 6). This is because only a tiny fraction of accounts post large amounts of URLs. For example, *all* domains in Cluster 4 are injected by a single account (josephjett), which is a Popular Content Redirector. It tweets all of 39 URLs to Reuters, Investors, HBR and Fortune. The account is owned by a corporate finance expert and solely uses `dlvr.it` (a social media automation and scheduling app) to post tweets mainly on a number of related themes, including corporate finance, business, and politics.

Similar examples can be highlighted across Cluster 5 – Stellar active engagers, *e.g.* one of the 25 bot accounts (animeseiyu) tweets 410 of 425 URLs to video streaming services (YouTube), Japanese entertainment websites (`kiramune.jp`, `lantis.jp`), and Japanese anime news websites (`moca-news.net`). It is also worth briefly comparing the various bot clusters against the remaining human accounts in my dataset. Again YouTube is the dominant domain, but I also see OSNs (Facebook) and app stores (Google Play and iTunes).

Many of the accounts in Cluster 6 produce URLs as part of tweets to various political and news websites (`politico.com`, `topstarnews.net`, `sinembargo.mx`, `washingtonpost.com`). Cluster 7 does not tweet any URL that I was able to redirect successfully. This was probably because the URLs had either been suspended, expired or deleted.

Next, I collect and use a supplementary dataset to study the impact of Web bots on Twitter content and activity.

6.5 The Social Cost of Web Bots

According to an estimate 51.8% of all Web traffic is generated by bots[13]. In this section, I quantify the impact of Web bots on content popularity and activity on Twitter. Web bots could be of many types, such as crawlers, indexers, content curators and publishers. I show that despite Web bots being smaller in numbers, they exercise a profound impact on content popularity and activity on Twitter.

To quantify the impact of Web bots, I set up a bot account on Twitter and conduct analysis on the dataset of click logs (Table 3.5) collected on the Web server. I then characterise the properties of bots using the click logs dataset, highlighting key properties in terms of impact on URL popularity, revisiting behaviour, and use of IP addresses and Autonomous Systems to launch requests or clicks.

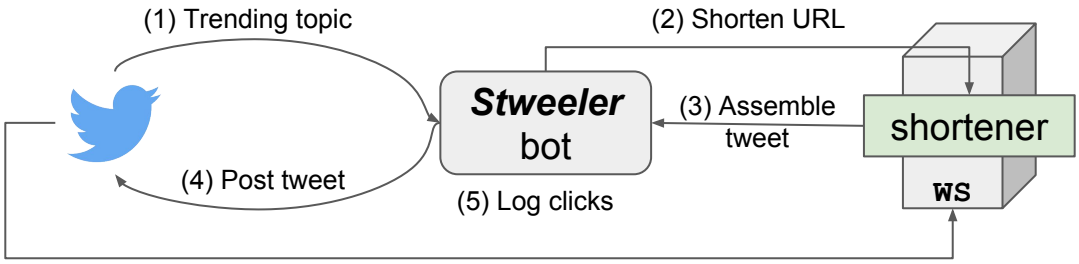


Figure 6.10: How ***Stweeler*** bot works.

6.5.1 Setting up a bot account

I extend ***Stweeler*** (Chapter 3) to collect click logs dataset (Table 3.5) from my web server powered by the Twitter bot. The honeypot bot[14] (Figure 6.10) operates as follows: (*i*) The bot fetches a popular 'job' related tweet from the Twitter Streaming API. It then disassembles the text and URL in the tweet. (*ii*) The URL is then fetched into the web server (WS). The WS runs a shortener module that shortens the URL into a reserved domain name. The shortener is needed to enable redirecting click traffic to the WS in order to collect click logs. (*iii*) The bot reassembles the tweet using the text and shortened URL. (*iv*) The tweet is then posted to my bot's Twitter account. In essence, the Twitter bot and WS performs a simple 'tweet manipulation' to avoid retweeting, which would otherwise prevent the click logs dataset from being obtained. (*v*) Finally, whenever a user (Twitter user or from the Web) clicks on a tweet(s) or URL(s), the WS records the click. Table 6.10 shows the type of information that is collected. Note that in order to respect the ethical boundaries of social media research, I **only** collect publicly available data about users and hash sensitive information such as IP addresses.

Table 6.10: Data collected through click logging.

Data attribute	**Description**
Click timestamp	Date and time of click, local to my web server.
Tweet ID	Tweet ID which received a click.
Hashed IP address	Hashed IP address of the machine that clicked the URL in the tweet identified by Tweet ID.
AS number	Obtained using the IP addresses from CAIDA.
User agent string	This records the `HTTP_USER_AGENT` string of the user clicking the URL in the tweet identified by Tweet ID.

[13]Bot traffic report 2016 (last accessed 16 June 2018) – `https://www.incapsula.com/blog/bot-traffic-report-2016.html`

[14]Details of honeypot experiment can also be found in Appendix A.2.

6.5.2 Bot detection

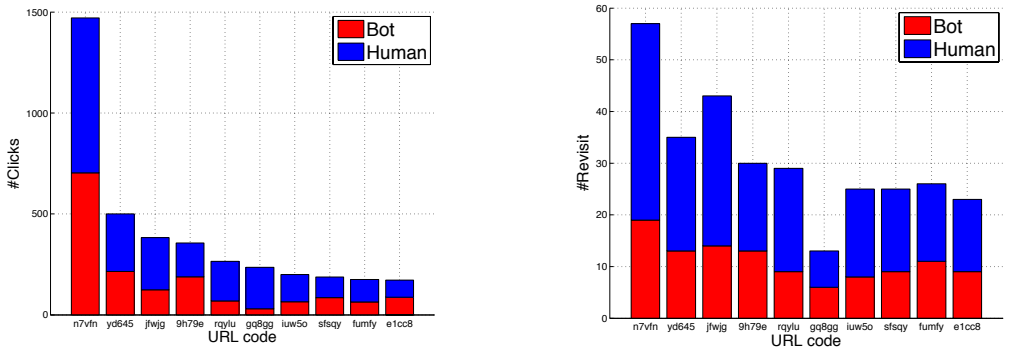
For the purposes of this particular study I implemented a simple bot detection method. I used the two most relevant features from the click logs dataset, *i.e.* (*i*) click frequency, and (*ii*) User agent strings. I use a different technique to Chapter 5 because bots on the Web are different to bots on Twitter, thus presenting a completely different dataset (§ 3.4.5) and activity profile. Since these bots do not exist on the Twitter platform, they do not present the vast array of attributes available from Twitter data. The information these bots generally expose is outlined in Table 6.10.

My Twitter bot account receives more than 223,000 clicks from 21-11-2015 to 08-01-2017. Out of these 223,000 clicks more than 44.91% have been produced by some sort of automated agent or a bot. I use a simple two-step bot detection method by analysing (*i*) frequency of clicks, and (*ii*) User agent strings. I employ time series analysis that takes into account the frequency of clicks by a single Twitter user account. As shown in [18] higher tweet frequency is indicative of automated behaviour. I then perform User agent string analysis, which reveals properties such as a URL containing description of the tool responsible for performing clicks on my URLs. Moreover, I find that there are a total of 2,563 unique visitors, out of which only 113 are unique bots that have a recurring presence. These facts are summarised in Table 3.5.

6.5.3 Characterisation

Next I highlight important behavioural properties of bots and humans. These include click activity, revisiting a previously visited URL, and the use of IP addresses and Autonomous Systems (AS) to launch requests to the deployed web server. Note that a tweet might have one or more URLs, however each request translates to one click on one URL. Since one request is triggered by one click, therefore they are equivalent in this chapter.

Surprisingly, from my click logs dataset only 4.08% of the visitors to my tweets or URLs are Web bots but are responsible for almost half of the clicks (44.91%). In contrast, from my Twitter dataset I found 43.13% accounts were operated by bots which were responsible for 53.90% statuses. However, bots in my click logs dataset account for a large chunk of the traffic produced on and contributed to the Twitter CDN and the Web. This finding points to interesting implications since

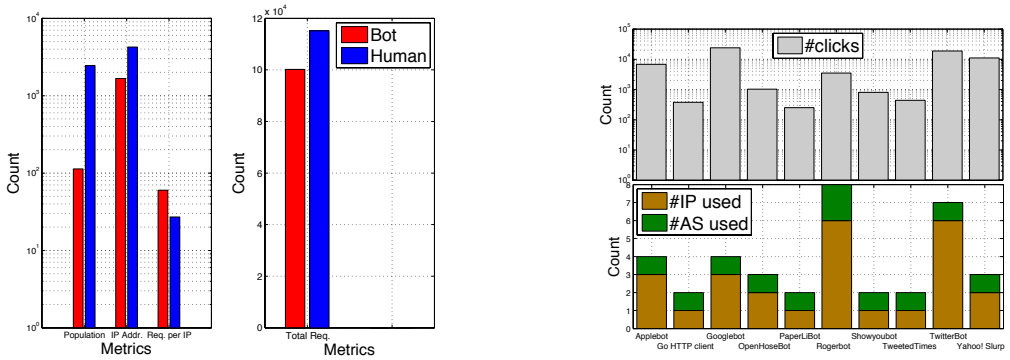


(a) Clicks on top most popular URLs.

(b) Revisits on top most popular URLs.

Figure 6.11: Click logs dataset - Clicks, Revisits.

bots not only access these URLs on the Web, but may also repost or retweet these tweets on their Twitter page or elsewhere using the website or platform-specific APIs. This is evident from Figure 6.11.



(a) IP addresses and requests.

(b) IPs and ASs used by bots.

Figure 6.12: Click logs dataset - IPs and requests, IPs and ASs used by bots.

Figure 6.11a shows the number of clicks received by top 10 most popular URLs that my bot posted on its Twitter page. The URL code is the shortened suffix that replaces the original URL. The most popular URL for bots (`n7vfn`) advertises a UI/UX job in Sunnyvale CA, and the least popular URL for bots (`gq8gg`) advertises a job in Nairobi. The top 10 list would change by at least 3 URLs if bots had not existed, thus clearly showing that bots cause the rise in URL popularity.

Revisits are more typical for humans than bots, as observed in Figure 6.11b. This is because these bots usually follow tweet streams which always flow for-

wards, thus requiring additional functionality for fetching historic profile. Moreover, some of the bots in my click logs dataset are actually content crawlers that maintain databases to avoid performing repeated activity.

Figure 6.12a shows the distribution of IP addresses used by bots vs IP addresses used by humans. 113 bots use 1,667 unique IP addresses to generate a total of 100,194 requests. On the other hand 2,450 humans use 4,258 unique IP addresses to generate a total of 115,137 requests. Human activity per IP address is considerably lower (27 requests per IP) than bots (60 requests per IP).

Lastly, Figure 6.12b shows the distribution of number of unique IP addresses and Autonomous Systems (AS) used by the top 10 most active bots (rank based on User agent string analysis), along with their click activity. The top most active bots detected from my click logs dataset tend to be Twitter bots that make use of the Twitter API to perform actions (Twitterbot = 18,828 clicks), web crawlers and indexers (Googlebot = 15,790, Yahoo! Slurp = 11,022, Applebot = 6,755), and content curators and publishers (PaperLiBot = 249, TweetedTimes = 437). There is a possibility that Twitter might also inject its own bots for account profiling, spam detection, monitoring and reporting, by using its BotMaker software.

Typically, the top most active bots use multiple static IP addresses from within a single AS, possibly to parallelise tasks. Interestingly, this possibility is further supported by the fact that all except one AS (25 of 26) are designated as type 'Content' (content hosting and distribution system), while only one is designated as type 'Transit/Access' (connecting networks through itself). Furthermore, in the dataset for the top 10 most active bots, there was one exception of an unusually aggressive (but benign) bot called Rogerbot, a web crawler for a marketing firm, that used 6 IPs from 2 ASes to register 3,485 clicks.

6.6 Takeaways

Social bots are not unitary. In this chapter I explored the various shapes and forms of social bots, that exist as semi-automated and fully automated social entities. Using the ***Stweeler*** bot classifier (Chapter 5) I detect bots from the datasets. I then decomposed the bots into a set of clusters exhibiting similar traits. To achieve this, I developed an unsupervised clustering task to create *un-*

labelled clusters from features. I observe a range of behaviours, with three highly populated clusters made up of bot accounts that follow well-known promotional strategies. I also found a range of software services, tools and apps specifically dedicated to generate tweet content and Twitter account management. Curiously, it is observed that less popular accounts utilised a mix of apps and human intervention (*e.g.* Web clients). This empirically confirmed that bots are *not* one type, but are highly diverse with various patterns both in terms of their own behaviour and the reactions of others.

Through a series of topical analyses, I then generated labels for these groups based on the principle components of discussion within each cluster. I found that the clusters focus on a range of overlapping topics, particularly: Advertisements & Marketing, Daily Affairs & Lifestyle, International Affairs, News, Politics. I further investigated the content of the tweets through polarity and subjectivity of language used within each tweet. Although all clusters broadly exhibited positive sentiment (i.e. > 0) and similar variance (0.0255–0.0572), a greater spread of polarity was found that ranged from very low (0.0454), *i.e.* neutral content to medium high (0.2777), *i.e.* definitely positive content.

Finally, I inspected the content links that accounts include in their tweet (*i.e.* URLs). Although, examples of mainstream websites are found (*e.g.* `youtube.com` is the most popular across most clusters), various other URLs are also observed. These are largely dominated by a few accounts that contribute a disproportionately large number of URLs within each cluster. For example, one cluster (#2) contains links to `elevatedfaith.com` 926 times, just from a single account.

However, bots that exist outside the Twitter ecosystem can too impact content popularity and activity on Twitter. To study this I extended ***Stweeler*** to implement a honeypot experiment to provide empirical evidence that the impact on Twitter is not restricted to social bots on Twitter. Rather, bots on and off Twitter form part of the larger automated *agents of influence* ecosystem, whose reach and impact spreads across the Web. I showed bots, even from the Web, play a significant role in boosting URL popularity, demonstrate differences in URL revisiting behaviour, and exercise increased usage of IP addresses and ASes to launch requests.

Such a study provides supplementary evidence that bots indeed have many types, and impact the popularity of content on Twitter while existing beyond its boundaries. More generally, by carrying out an exhaustive analysis I find that

bots exist in diverse quantities: from hyper-active content producers to extremely popular passive bots, and from social bots on Twitter to Web bots interacting with Twitter content. If some are found to be tweeting positively about a product or a political candidate, others are found to be sarcastic and negative. Through these studies I have effectively shown generalisability and applicability of the ***Stweeler*** platform to a wider array of domain-specific problems. I am also confident that ***Stweeler*** could be very useful in producing new research in future.

Chapter 7

Final Remarks

Social bots contribute a significant amount of activity on Twitter. They consume and produce content, and interact with human users via Twitter's many functions (retweets, replies, mentions, likes, *etc*). Social bots are function-driven – functions that are defined by their human masters.

During the course of research encompassed within this dissertation, I have largely contributed to methods and tools that enable measuring, detecting and investigating bots in online social networks using tools and techniques from data science and machine learning. I embarked on the mission by first properly defining the problem, outlining the background research (Chapter 2) and introducing a framework (Chapter 3), measuring and characterising bots through exploratory data science (Chapter 4), detecting bots through supervised machine learning (Chapter 5), and categorising bots to discern types using unsupervised machine learning and exploring the Web bots through the use of data curated from the Web (Chapter 6).

7.1 Summary and Conclusions

During the beginning of this dissertation I set out a path as well as a framework that would be extended along the journey of this research. I began in Chapter 1 by introducing the scale of the problem and setting specific, measurable, and attainable goals for this work, as well as outlining major contributions of this dissertation. In Chapter 2–3, I outlined the background work and formally introduced the ***Stweeler*** framework to the reader. Chapter 3 also introduced all

of the datasets used for the purposes of research carried out in Chapters 4–6. In Chapter 4, I found that bots exercise a tremendous impact on Twitter. The work gave me a set of principal features that I could use to formulate an understanding of how bots are different to humans. I found bots to be generally more active, but neither as novel as humans nor as appreciated as humans, in terms of content produced. I also found that humans and bots maintain a certain characteristic *homophily* amongst their kind, despite the lack of any real knowledge of another user being a bot or human. Unsurprisingly, humans formed far more reciprocal relationships than bots. I also argued that bot traffic can impact many aspects of network operations, including traffic engineering, routing, cloud computing, content distribution networks and quality of service.

Chapter 4 paved the way for Chapter 5, in which I used these findings to develop and evaluate a thorough mechanism to reliably classify bots and humans, through a supervised machine learning task. I used a dataset divided into four major popularity groups and found how different feature splits performed for different detection experiments. I found statistically most significant features that could be utilised for accurately detecting bots. My evaluation revealed that the ***Stweeler*** classifier was twice as much accurate than the current state of the art bot detection tool.

These bot activities may lead to dramatic changes in social structures and interactions in the longterm (as the bot population increases). Thus, there is a wide array of problems to explore in future, such as: exploring credibility scores, influence botnets, analysing bot content, and developing accurate detection tools. Credibility of social media accounts and their following could be used as one of the defining features for detecting *dark* bots. I therefore envisage that, in the longterm, the distinction between human and bot research will wane, with greater integration of their activities (*e.g.* greater automation of human accounts).

Using the ***Stweeler*** classifier developed in Chapter 5 I obtained a pre-classified bot dataset in Chapter 6 that enabled a deeper understanding of types of bots. Through unsupervised clustering I was able to divide a singular bot population into a number of types. Then through topic modelling I was able to do content analysis to distinguish what different categories of bots produce as content. Through an exhaustive analysis I found bots that varied from hyper-active content producers to social chameleons. I even found individual bot-operated accounts having *quasi*-celebrity status.

This work opened possibilities for related research in the future. A lot can be learned from topic analysis of the type of lists an account is following: *e.g.* if the main goal of an agent is to expand its reach it can be assumed that the agent account would try to follow many different lists without particular topic coherence. Another line of work could explore the provenance of social *botnets*, and ask if least popular Twitter accounts (having minimum activity) are being used to artificially inflate another account's popularity.

Finally, in Chapter 6 I used ***Stweeler*** for studying bots more generally on the Web. This was accomplished by deploying a honeypot experiment consisting of a *bespoke* bot, a URL shortener and a Web server. It was found that bots can have a substantial affect on Twitter by impacting the popularity of content that is displayed on the platform.

7.2 Future Directions

Though I have covered a wide spectrum of bot phenomenon, there is a list of work outstanding. This dissertation paves the way for more research into this developing phenomenon, as outlined below.

One of the most pressing issues is obtaining and updating the ground-truth datasets for supervised classification. Supervised learning, particularly classification, requires a training sample that is most often created by human annotators. This task is tedious as well as requires a boilerplate involving task description, recruiting annotators, data preprocessing to make it human readable and understandable, ensuring high quality through verification of results. All of this comes at the cost of time and money, and it is impossible to scale or diversify to another dataset. Despite a few drawbacks human annotators typically perform high quality annotations because of two reasons: (*i*) their cognitive ability to relate terms and not be restricted to the set of those terms but use a term that represents all of the given terms e.g. the words "chapters, contents, index" immediately bring the term 'book' to our minds, (*ii*) realise the context beyond the corpus.

Though nearly impossible to accomplish without human or manual participation, perhaps this could be alleviated by extending ***Stweeler*** to automatically verify and flag post-classified datasets for bot and human labels.

Despite the flexibility of unsupervised learning methods, they are prone to in-

accuracy if not applied properly. There is a great opportunity to extend ***Stweeler*** with a combination of semi-supervised (such as [68]) and unsupervised approaches to continue automated labelling of bot categories. This will enable deeper understanding into the *latent* bot categories that we do not know about.

7.3 Last Thoughts

Automation in social systems is a genuinely new direction. Made possible by machine learning and language processing, its power is unprecedented and its affects are profound. The impact factors of *social automation* are hard to measure due to the interdisciplinary knowledge requirements and issues concerning business, ethics, law, sociology and practical computing systems knowledge. In this dissertation I have taken the first few steps to address the implementation requirements that should enable researchers of the future to utilise for understanding this nascent social phenomenon. Nonetheless, the age of *cognisant machines* is here.

Bibliography

[1] Norah Abokhodair, Daisy Yoo, and David W. McDonald. Dissecting a social botnet: Growth, content and influence in twitter. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, CSCW '15, pages 839–851, New York, NY, USA, 2015. ACM.

[2] David W Aha, Dennis Kibler, and Marc K Albert. Instance-based learning algorithms. *Machine learning*, 6(1):37–66, 1991.

[3] Luca Maria Aiello, Martina Deplano, Rossano Schifanella, and Giancarlo Ruffo. People are strange when you're a stranger: Impact and influence of bots on social networks. In *Proceedings of the 6th International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2012.

[4] Aris Anagnostopoulos, Ravi Kumar, and Mohammad Mahdian. Influence and correlation in social networks. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 7–15, New York, NY, USA, 2008. ACM.

[5] Jonell Baltazar, Joey Costoya, and Ryan Flores. The real face of koobface: The largest web 2.0 botnet explained. *Trend Micro Research*, 5(9):10, 2009.

[6] Marco T. Bastos and Dan Mercea. The brexit botnet and user-generated hyperpartisan news. *Social Science Computer Review*, 0(0):0894439317734157, 0.

[7] Fabricio Benevenuto, Gabriel Magno, Tiago Rodrigues, and Virgilio Almeida. Detecting spammers on twitter. In *Collaboration, electronic messaging, anti-abuse and spam conference (CEAS)*, volume 6, page 12, 2010.

[8] Alessandro Bessi and Emilio Ferrara. Social bots distort the 2016 u.s. presidential election online discussion. *First Monday*, 21(11), 2016.

[9] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[10] Yazan Boshmaf, Ildar Muslukhov, Konstantin Beznosov, and Matei Ripeanu. The socialbot network: When bots socialize for fame and money. In *Proceedings of the 27th Annual Computer Security Applications Conference*, ACSAC '11, pages 93–102, New York, NY, USA, 2011. ACM.

[11] Yazan Boshmaf, Ildar Muslukhov, Konstantin Beznosov, and Matei Ripeanu. Design and analysis of a social botnet. *Computer Networks*, 57(2):556 – 578, 2013. Botnet Activity: Analysis, Detection and Shutdown.

[12] D. Boyd, S. Golder, and G. Lotan. Tweet, tweet, retweet: Conversational aspects of retweeting on twitter. In *System Sciences (HICSS), 2010 43rd Hawaii International Conference on*, pages 1–10, Jan 2010.

[13] Danah Boyd. The politics of "real names". *Commun. ACM*, 55(8):29–31, August 2012.

[14] Jian Cao, Qiang Li, Yuede Ji, Yukun He, and Dong Guo. Detection of forwarding-based malicious urls in online social networks. *International Journal of Parallel Programming*, 44(1):163–180, Feb 2016.

[15] Qiang Cao, Michael Sirivianos, Xiaowei Yang, and Tiago Pregueiro. Aiding the detection of fake accounts in large scale social online services. In *Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation*, NSDI'12, pages 15–15, Berkeley, CA, USA, 2012. USENIX Association.

[16] Meeyoung Cha, Hamed Haddadi, Fabricio Benevenuto, and P Krishna Gummadi. Measuring user influence in twitter: The million follower fallacy. *Icwsm*, 10(10-17):30, 2010.

[17] Kuan-Ta Chen, Hsing-Kuo Kenneth Pao, and Hong-Chung Chang. Game bot identification based on manifold learning. In *Proceedings of the 7th ACM SIGCOMM Workshop on Network and system Support for Games*, NetGames '08, pages 21–26, New York, NY, USA, 2008. ACM.

[18] Zi Chu, Steven Gianvecchio, Haining Wang, and Sushil Jajodia. Who is tweeting on twitter: Human, bot, or cyborg? In *Proceedings of the 26th Annual Computer Security Applications Conference*, ACSAC '10, pages 21–30, New York, NY, USA, 2010. ACM.

[19] Zi Chu, Indra Widjaja, and Haining Wang. Detecting social spam campaigns on twitter. In Feng Bao, Pierangela Samarati, and Jianying Zhou, editors, *Applied Cryptography and Network Security*, pages 455–472, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[20] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.

[21] Stefano Cresci, Roberto Di Pietro, Marinella Petrocchi, Angelo Spognardi, and Maurizio Tesconi. The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race. *CoRR*, abs/1701.03017, 2017.

[22] Clayton Allen Davis, Onur Varol, Emilio Ferrara, Alessandro Flammini, and Filippo Menczer. Botornot: A system to evaluate social bots. In *Proceedings of the 25th International Conference Companion on World Wide Web*, WWW '16 Companion, pages 273–274, Republic and Canton of Geneva, Switzerland, 2016. International World Wide Web Conferences Steering Committee.

[23] O. V. Deryugina. Chatterbots. *Scientific and Technical Information Processing*, 37(2):143–147, Apr 2010.

[24] Yifan Ding, Liqiang Wang, Deliang Fan, and Boqing Gong. A semi-supervised two-stage approach to learning from noisy labels. *arXiv preprint arXiv:1802.02679*, 2018.

[25] Nicolas Dugué, Anthony Perez, Maximilien Danisch, Florian Bridoux, Amélie Daviau, Tennessy Kolubako, Simon Munier, and Hugo Durbano. A reliable and evolutive web application to detect social capitalists. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, pages 741–744. ACM, 2015.

[26] Chad Edwards, Autumn Edwards, Patric R. Spence, and Ashleigh K. Shelton. Is that a bot running the social media feed? testing the differences in perceptions of communication quality for a human agent and a bot agent on twitter. *Computers in Human Behavior*, 33:372 – 376, 2014.

[27] Emilio Ferrara. Disinformation and social bot operations in the run up to the 2017 french presidential election. *First Monday*, 22(8), 2017.

[28] Alessandro Finamore, Marco Mellia, Zafar Gilani, Konstantina Papagiannaki, Vijay Erramilli, and Yan Grunenberger. Is there a case for mobile phone content pre-staging? In *Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies*, CoNEXT '13, pages 321–326, New York, NY, USA, 2013. ACM.

[29] Asbjørn Følstad and Petter Bae Brandtzæg. Chatbots and the new world of hci. *Interactions*, 24(4):38–42, June 2017.

[30] Michelle C Forelle, Philip N Howard, Andrés Monroy-Hernández, and Saiph Savage. Political bots and the manipulation of public opinion in venezuela. 2015.

[31] Carlos Freitas, Fabricio Benevenuto, Saptarshi Ghosh, and Adriano Veloso. Reverse engineering socialbot infiltration strategies in twitter. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, ASONAM '15, pages 25–32, New York, NY, USA, 2015. ACM.

[32] Hongyu Gao, Jun Hu, Christo Wilson, Zhichun Li, Yan Chen, and Ben Y. Zhao. Detecting and characterizing social spam campaigns. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, IMC '10, pages 35–47, New York, NY, USA, 2010. ACM.

[33] Saptarshi Ghosh, Bimal Viswanath, Farshad Kooti, Naveen Kumar Sharma, Gautam Korlam, Fabricio Benevenuto, Niloy Ganguly, and Krishna Phani Gummadi. Understanding and combating link farming in the twitter social network. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 61–70, New York, NY, USA, 2012. ACM.

[34] Steven Gianvecchio and Haining Wang. Detecting covert timing channels: An entropy-based approach. In *CCS '07*. ACM, 2007.

[35] Steven Gianvecchio, Zhenyu Wu, Mengjun Xie, and Haining Wang. Battle of botcraft: Fighting bots in online games with human observational proofs. In *Proceedings of the 16th ACM Conference on Computer and Communications Security*, CCS '09, pages 256–268, New York, NY, USA, 2009. ACM.

[36] Zafar Gilani, Jon Crowcroft, Reza Farahbakhsh, and Gareth Tyson. The implications of twitterbot generated data traffic on networked systems. In *Proceedings of the SIGCOMM Posters and Demos*, SIGCOMM Posters and Demos '17, pages 51–53, New York, NY, USA, 2017. ACM.

[37] Zafar Gilani, Reza Farahbakhsh, and Jon Crowcroft. Do bots impact twitter activity? In *Proceedings of the 26th International Conference on World Wide Web Companion*, WWW '17 Companion, pages 781–782, Republic and Canton of Geneva, Switzerland, 2017. International World Wide Web Conferences Steering Committee.

[38] Zafar Gilani, Reza Farahbakhsh, Gareth Tyson, Liang Wang, and Jon Crowcroft. An in-depth characterisation of bots and humans on twitter. *arXiv preprint arXiv:1704.01508*, 2017.

[39] Zafar Gilani, Reza Farahbakhsh, Gareth Tyson, Liang Wang, and Jon Crowcroft. Of bots and humans (on twitter). In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, ASONAM '17, pages 349–354, New York, NY, USA, 2017. ACM.

[40] Zafar Gilani, Ekaterina Kochmar, and Jon Crowcroft. Classification of twitter accounts into automated agents and human users. In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, ASONAM '17, pages 489–496, New York, NY, USA, 2017. ACM.

[41] Zafar Gilani, Liang Wang, Jon Crowcroft, Mario Almeida, and Reza Farahbakhsh. Stweeler: A framework for twitter bot analysis. In *Proceedings of the 25th International Conference Companion on World Wide Web*, WWW '16 Companion, pages 37–38, Republic and Canton of Geneva, Switzerland, 2016. International World Wide Web Conferences Steering Committee.

[42] Chris Grier, Kurt Thomas, Vern Paxson, and Michael Zhang. @spam: The underground on 140 characters or less. In *Proceedings of the 17th ACM Conference on Computer and Communications Security*, CCS '10, pages 27–37, New York, NY, USA, 2010. ACM.

[43] James Grimmelmann. The law and ethics of experiments on social media users. *J. on Telecomm. & High Tech. L.*, 13:219, 2015.

[44] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 1, pages 278–282 vol.1, Aug 1995.

[45] Bernie Hogan. Pseudonyms and the rise of the real-name web. 2012.

[46] Eduard Hovy and Chin-Yew Lin. Automated text summarization and the summarist system. In *Proceedings of a Workshop on Held at Baltimore, Maryland: October 13-15, 1998*, TIPSTER '98, pages 197–214, Stroudsburg, PA, USA, 1998. Association for Computational Linguistics.

[47] Eduard H. Hovy. Automated discourse generation using discourse structure relations. *Artificial Intelligence*, 63(1):341 – 385, 1993.

[48] Philip N Howard and Bence Kollanyi. Bots,# strongerin, and# brexit: Computational propaganda during the uk-eu referendum. *Browser Download This Paper*, 2016.

[49] Qian Huang, Zhu Liu, Aaron Rosenberg, David Gibbon, and Behzad Shahraray. Automated generation of news content hierarchy by integrating audio, video, and text information. In *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, volume 6, pages 3025–3028. IEEE, 1999.

[50] H. Husna, S. Phithakkitnukoon, and R. Dantu. Traffic shaping of spam botnets. In *CCNC 2008*. IEEE, 2008.

[51] Yuede Ji, Yukun He, Xinyang Jiang, Jian Cao, and Qiang Li. Combating the evasion mechanisms of social bots. *Comput. Secur.*, 58(C):230–249, May 2016.

[52] George H John. Robust decision trees: Removing outliers from databases. In *KDD*, pages 174–179, 1995.

[53] Armand Joulin, Laurens van der Maaten, Allan Jabri, and Nicolas Vasilache. Learning visual features from large weakly supervised data. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 67–84, Cham, 2016. Springer International Publishing.

[54] Erhan J. Kartaltepe, Jose Andre Morales, Shouhuai Xu, and Ravi Sandhu. Social network-based botnet command-and-control: Emerging threats and countermeasures. In Jianying Zhou and Moti Yung, editors, *Applied Cryptography and Network Security*, pages 511–528, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[55] Balachander Krishnamurthy, Phillipa Gill, and Martin Arlitt. A few chirps about twitter. In *Proceedings of the First Workshop on Online Social Networks*, WOSN '08, pages 19–24, New York, NY, USA, 2008. ACM.

[56] J Richard Landis and Gary G Koch. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174, 1977.

[57] Kyumin Lee, James Caverlee, and Steve Webb. Uncovering social spammers: Social honeypots + machine learning. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 435–442, New York, NY, USA, 2010. ACM.

[58] Kyumin Lee, Brian David Eoff, and James Caverlee. Seven months with the devils: A long-term study of content polluters on twitter. In *ICWSM*, 2011.

[59] Yuncheng Li, Jianchao Yang, Yale Song, Liangliang Cao, Jiebo Luo, and Li-Jia Li. Learning from noisy labels with distillation. In *ICCV*, pages 1928–1936, 2017.

[60] André L. B. Miranda, Luís Paulo F. Garcia, André C. P. L. F. Carvalho, and Ana C. Lorena. Use of classification algorithms in noise detection and elimination. In Emilio Corchado, Xindong Wu, Erkki Oja, Álvaro Herrero, and Bruno Baruque, editors, *Hybrid Artificial Intelligence Systems*, pages 417–424, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[61] Silvia Mitter, Claudia Wagner, and Markus Strohmaier. Understanding the impact of socialbot attacks in online social networks. *CoRR*, abs/1402.6289, 2014.

[62] Fabrice Muhlenbach, Stéphane Lallich, and Djamel A. Zighed. Identifying and handling mislabelled instances. *Journal of Intelligent Information Systems*, 22(1):89–109, Jan 2004.

[63] Max Nanis, Ian Pearce, and Tim Hwang. Pacsocial: Field test report, 2011.

[64] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 1196–1204. Curran Associates, Inc., 2013.

[65] Jose Nazario and Thorsten Holz. As the net churns: Fast-flux botnet observations. In *Malicious and Unwanted Software, 2008. MALWARE 2008. 3rd International Conference on*, pages 24–31. IEEE, 2008.

[66] Christopher Olston and Marc Najork. Web crawling. *Found. Trends Inf. Retr.*, 4(3):175–246, March 2010.

[67] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[68] Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D. Manning. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 248–256, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[69] Jacob Ratkiewicz, Michael Conover, Mark Meiss, Bruno Gonçalves, Snehal Patil, Alessandro Flammini, and Filippo Menczer. Truthy: Mapping the spread of astroturf in microblog streams. In *Proceedings of the 20th International Conference Companion on World Wide Web*, WWW '11, pages 249–252, New York, NY, USA, 2011. ACM.

[70] Jacob Ratkiewicz, Michael Conover, Mark R. Meiss, Bruno Gonçalves, Snehal Patil, Alessandro Flammini, and Filippo Menczer. Detecting and tracking the spread of astroturf memes in microblog streams. *CoRR*, abs/1011.3768, 2010.

[71] Saiph Savage, Andres Monroy-Hernandez, and Tobias Höllerer. Botivist: Calling volunteers to action using online bots. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, CSCW '16, pages 813–822, New York, NY, USA, 2016. ACM.

[72] Lauren Scissors, Moira Burke, and Steven Wengrovitz. What's in a like?: Attitudes and behaviors around receiving likes on facebook. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, CSCW '16, pages 1501–1510, New York, NY, USA, 2016. ACM.

[73] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[74] Ashutosh Singh. Social networking for botnet command and control. 2012.

[75] Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and William B. Dolan. A neural network approach to context-sensitive generation of conversational responses. In *HLT-NAACL*, pages 196–205. Association for Computational Linguistics, May–June 2015.

[76] Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. Detecting spammers on social networks. In *Proceedings of the 26th Annual Computer Security Applications Conference*, ACSAC '10, pages 1–9, New York, NY, USA, 2010. ACM.

[77] V. S. Subrahmanian, A. Azaria, S. Durst, V. Kagan, A. Galstyan, K. Lerman, L. Zhu, E. Ferrara, A. Flammini, and F. Menczer. The darpa twitter bot challenge. *Computer*, 49(6):38–46, June 2016.

[78] Jiang-wen Sun, Feng-ying Zhao, Chong-jun Wang, and Shi-fu Chen. Identifying and correcting mislabeled training instances. In *Future generation communication and networking (FGCN 2007)*, volume 1, pages 244–250. IEEE, 2007.

[79] Ruck Thawonmas, Yoshitaka Kashifuji, and Kuan-Ta Chen. Detection of mmorpg bots based on behavior analysis. In *Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology*, ACE '08, pages 91–94, New York, NY, USA, 2008. ACM.

[80] Kurt Thomas, Chris Grier, and Vern Paxson. Adapting social spam infrastructure for political censorship. In *LEET*, 2012.

[81] Kurt Thomas, Chris Grier, Dawn Song, and Vern Paxson. Suspended accounts in retrospect: An analysis of twitter spam. In *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, IMC '11, pages 243–258, New York, NY, USA, 2011. ACM.

[82] Christian Thurau, Christian Bauckhage, and Gerhard Sagerer. Learning human-like movement behavior for computer games. In *Proc. Int. Conf. on the Simulation of Adaptive Behavior*, pages 315–323, 2004.

[83] Onur Varol, Emilio Ferrara, Clayton A Davis, Filippo Menczer, and Alessandro Flammini. Online human-bot interactions: Detection, estimation, and characterization. *arXiv preprint arXiv:1703.03107*, 2017.

[84] A. Vishwanath, Jiazhen Zhu, K. Hinton, R. Ayre, and R. S. Tucker. Estimating the energy consumption for packet processing, storage and switching in optical-ip routers. In *OFC/NFOEC, 2013*, pages 1–3, March 2013.

[85] Bimal Viswanath, Muhammad Ahmad Bashir, Mark Crovella, Saikat Guha, Krishna P Gummadi, Balachander Krishnamurthy, and Alan Mislove. Towards detecting anomalous user behavior in online social networks. In *Usenix Security*, volume 14, 2014.

[86] Claudia Wagner, Silvia Mitter, Christian Körner, and Markus Strohmaier. When social bots attack: Modeling susceptibility of users in online social networks. *Making Sense of Microposts (# MSM2012)*, 2, 2012.

[87] De Wang, Shamkant B Navathe, Ling Liu, Danesh Irani, Acar Tamersoy, and Calton Pu. Click traffic analysis of short url spam on twitter. In *Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom), 2013 9th International Conference Conference on*, pages 250–259. IEEE, 2013.

[88] Jen Weedon, William Nuland, and Alex Stamos. Information operations and facebook. *version*, 1:27, 2017.

[89] Joseph Weizenbaum. Eliza—a computer program for the study of natural language communication between man and machine. *Commun. ACM*, 9(1):36–45, January 1966.

[90] Xian Wu, Ziming Feng, Wei Fan, Jing Gao, and Yong Yu. *Detecting Marionette Microblog Users for Improved Information Credibility*, pages 483–498. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[91] Jeff Yan. Bot, cyborg and automated turing test. In *International Workshop on Security Protocols*, pages 190–197. Springer, 2006.

[92] Chao Yang, Robert Harkreader, Jialong Zhang, Seungwon Shin, and Guofei Gu. Analyzing spammers' social networks for fun and profit: A case study of cyber criminal ecosystem on twitter. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 71–80, New York, NY, USA, 2012. ACM.

[93] Louis Yu, Sitaram Asur, and Bernardo Huberman. Dynamics of trends and attention in chinese social media. 2013.

[94] Jinxue Zhang, Rui Zhang, Yanchao Zhang, and Guanhua Yan. On the impact of social botnets for spam distribution and digital-influence manipulation. In *Communications and Network Security (CNS), 2013 IEEE Conference on*, pages 46–54. IEEE, 2013.

[95] Ziming Zhao, Gail-Joon Ahn, and Hongxin Hu. Examining social dynamics for countering botnet attacks. In *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, pages 1–5. IEEE, 2011.

Appendix A

Tasks, Experiments and Ethics Approval

A.1 Human Annotation Task for Binary Classification

The Human (or Manual) Annotation Task adheres to the ethical considerations of the institutional ethics review board at the University of Cambridge Computer Laboratory[1]. This task is only indicative and informative, not disruptive or decisive.

A.1.1 Task Description

We recruited four undergraduate students for the purposes of annotation, who classified the accounts over the period of a month. This was done using a tool that automatically presents Twitter profiles, and allows the recruits to annotate the profile with a classification (bot or human) and add any extra comments. Each account was reviewed by all recruits independently, before being aggregated into a final judgement using a final collective review (via discussion among recruits if needed).

Human annotators were paid accordingly per task successfully performed. Per item payment made to 4 annotators was roughly USD 0.11 (PKR 11) per annotation for 3535 annotations. The task was completed in August 2016. A receipt of

[1] `https://www.cl.cam.ac.uk/local/policy/ethics/`

the payment that confirms the date can be requested via email (szuhg2@cam.ac.uk).

The task is to create a labelled dataset given a list of Twitter accounts (screen names) and list of sources for these accounts. There will be four lists each for Twitter accounts and their associated sources:

1. 10M followers
2. 1M followers
3. 100k followers
4. 1k followers

Note: It is recommended that at least 3-4 people perform this task independently of each other for fairness, cross inspection (inter-annotator agreement), and calculating confidence levels (Cohen's kappa). It is the responsibility of the human worker to make sure these lists are kept segregated.

The following attributes are provided to consider from Twitter profile for labelling an account as either human or bot:

1. date when account was created (not entirely sure if bots could be older than humans)
2. number of tweets, retweets, tweet frequency = number of tweets / age of account in days (if an accounts posts more than 25 tweets, that account has higher chances of being automated)
3. do they reply to tweets? (replying to tweets is an indication of human behaviour)
4. content they post on their Twitter wall (tweeting about certain topics only is a sign of automation)
5. number of favourited tweets (higher number is associated with human behaviour)
6. ratio of followers / friends (higher ratio is associated with human behaviour)
7. account profile description and picture (natural looking description and personal picture is a sign of human behaviour)
8. number of URLs posted in tweets (more URLs in tweets point towards automated behaviour)
9. size of content uploaded (more content points towards automated behaviour)

The other important piece of information is to consider sources used by a Twitter account to post content on Twitter. Sources information to consider:

1. number of sources used (higher number is associated with human behaviour)
2. types of sources (humans tend to use Twitter app from their devices such as smartphones and tablets, third party applications, Web interface; whereas, automated accounts might be using API, scheduling tools, automating tools, *etc.*).)

Note:

1. Known feature apps: echofon.com, snappytv.com, periscope.tv
2. Account sharing & scheduling: tweetdeck
3. Automation and scheduling: buffer.com, socialflow.com, hootsuite.com, sprinklr.com, spredfast.com, twuffer.com, sendible.com
4. Smart automation & scheduling: ifttt.com, dlvr.it

The worker might need to perform some research for tools listed in sources for each account. However, this is easy as he/she mostly only needs to go to the URL of a source given along with the source name in the source list. Using all this information a human worker will annotate a Twitter user as either human or bot along with reasons why did he/she annotate it as such, as done in the format and example below (Table A.1).

Table A.1: HAT example.

Twitter screen_name	**Reason**	**Annotation (bot, human)**
khloekardashian	uses iPhone and iPad to post tweets	human
nytimes	292K tweets since May 2010 = 130 tweets a day and uses an automating tool socialflow.com	bot

Rules for payment:

- Successful annotation = payment.
- If the worker fails to provide an annotation, payment for that annotation will be discounted or withheld.
- If the worker provides an annotation but the annotation fails to provide a well-defined reason in a phrase or a sentence, then the payment for that annotation will be discounted or withheld.

A.1.2 Ethics Approval #379

Ethics approval form as filled below, and subsequently approved.

TITLE: Characterising usage and impact of bots on Twitter

APPLICANTS: Syed Zafar ul Hussan Gilani, Jon Crowcroft

EMAIL: szuhg2@cam.ac.uk, jac22@cam.ac.uk

DATES: 01/07/2016 to 30/09/2016

STUDY TYPE: Other

FUNDING BODY: EU MARIE CURIE METRICS ITN

DESCRIPTION

The WWW has seen massive growth in variety and opportunistic usage of OSNs. Most of these pursuits are exploited via automated programs, aka bots. We know for a fact that more than 45% of clicks we get on tweets are from bots. ***Stweeler*** is a framework under development to study usage and impact of bots on Twitter from social media and systems perspectives. Our aim is to define and measure metrics to analyse how automated programs impact (1) user engagement, (2) content dissemination, (3) geographical spread of tweets, and (4) traffic contributed on the Web due to tweets (or due to Twitter CDN). Our goal is to model the impact of automation on information propagation in OSNs.

For this purpose we require a labelled dataset. Essentially, a list of Twitter accounts categorised / annotated / labelled into either humans or bots. Since the Machine Learning techniques fall short of accurately judging an account as either human or a bot, we would like humans workers to carry out the task. This will be done using Amazon Mechanical Turk. We are not studying any human workers or their responses/behaviour. This is purely an activity to create lists of human accounts and bot accounts divided into four buckets: (*i*) approx. 1M followers, (*ii*) approx. 100k followers, (*iii*) approx. 1k followers, and (*iv*) approx. 500 followers. The labelled dataset will be used to characterise the differences between human Twitter accounts and bot Twitter accounts, measure the impact of bot accounts on Twitter, and evaluate Machine Learning approaches to bot detection against this dataset.

We will provide four lists and their corresponding sources lists, one for each bracket. The human workers will have to look at the Twitter profile of those users, compare their attributes such as when was account created, number of tweets, do they reply to their tweets, what kind of stuff they tweet about, number of

favourite tweets, number of following (friends), number of followers, account profile description, account profile picture, *etc.* They will then look at the sources list to find the number of sources and what sort of sources a Twitter user employs to post content on Twitter: smartphone, tablet, Web interface, third party app, API, scheduling tools, *etc.* Using all this information a human worker will annotate a Twitter user as either human or bot.

PRECAUTIONS

All collected data from Twitter is public. Collection is done via the Twitter Streaming API. All annotations will be done using a controlled method and will reflect the outputs of a method along with what a human worker rates as a more important attributes *e.g.* number of tweets vs number of followers. No personal information will be collected regarding human workers.

A.2 Honeypot Experiment

The Honeypot Experiment adheres to the ethical considerations of the institutional ethics review board at the University of Cambridge Computer Laboratory[2]. This task is non-intrusive and non-engaging.

A.2.1 Task Description

A honeypot bot is deployed on a web server that operates a Twitter account. The bot uses the Twitter Streaming API to tweet job opportunities including shortened URLs. These URLs are shortened by the shortener service running on the web server. The shortener is needed to enable redirecting click traffic to the web server in order to collect click logs. The bot is non-intrusive and non-engaging. This experiment helps to find bots that exist on the Web, *i.e.* crawlers, indexers, spiders and curators.

The bot algorithm follows the steps as outlined: (*i*) The bot searches for a popular job-related tweet from the Twitter Streaming API. It then disassembles the text and URL in the tweet. (*ii*) The URL is then fetched into the web server. (*iii*) The bot reassembles the tweet using the text and shortened URL. (*iv*) The tweet is then posted to my bot's Twitter account. (*v*) Finally, whenever a user

[2] `https://www.cl.cam.ac.uk/local/policy/ethics/`

(Twitter user or from the Web) clicks on a tweet(s) or URL(s), the web server records the click.

A.2.2 Ethics Approval #556

Ethics approval form as filled below, and subsequently approved.

TITLE: The impact of Web bots on Twitter content

APPLICANTS: Syed Zafar ul Hussan Gilani, Jon Crowcroft

EMAIL: szuhg2@cam.ac.uk, jac22@cam.ac.uk

DATES: 21/11/2015 to 08/01/2017

STUDY TYPE: Other

FUNDING BODY: EU MARIE CURIE METRICS ITN

DESCRIPTION

This application is to check if the experiment (detailed below) fulfilled ethical considerations since the type of study does not study people, recruit outside participants, collect information on people or even release software.

The experiment deploys a honeypot bot that operates a Twitter account. The bot only tweets job opportunities including shortened URLs. These shortened URLs are shortened by a shortener service running on a deployed web server. The bot is non-intrusive and non-engaging, *i.e.* the bot does not engage in communication with other Twitter users. The purpose of this experiment was to find bots that exist on the Web, *i.e.* crawlers, indexers and curators, among others.

The web server collects all clicks performed on tweets posted by the bot. Click data can only be collected for those tweets which contain a shortened URL. Once a click is performed on the URL, the URL is redirected to the web server where the click is logged, before the click is redirected to the original source. The following pieces of information were collected: {timestamp, web browser or app name, IP address of web browser or app}.

As data is collected outside the Twitter platform, no user data (such as Twitter username, profile info, *etc.*) was collected. In fact, it was impossible to collect user data, since the web server can only collect clicks data and no information of who clicked it. Timestamp is date and time of click, web browser or app name is the software used to click (*e.g.* Chrome, Twitter Web App, Googlebot, Applebot, *etc.*), and IP address is collected to plot a time series of repeating sources as a heuristic to identify Web bots (crawlers, indexers, curators).

We did not share this data with any external entity and we did not try to identify the source of clicks.

PRECAUTIONS

Data is not shared with any external entity.

Minimum information is collected, *i.e.* timestamp, browser or app name, IP address of browser or app

Appendix B

Publications

This is a comprehensive list of papers published in conferences in reverse chronological order during my PhD (September 2014 to August 2017). **Bold face** shows publications that are directly relevant to this dissertation.

[36] **Zafar Gilani, Jon Crowcroft, Reza Farahbakhsh, and Gareth Tyson. "The Implications of Twitterbot Generated Data Traffic on Networked Systems." In Proceedings of the SIGCOMM Posters and Demos, pp. 51-53. ACM, 2017.**

[40] **Zafar Gilani, Ekaterina Kochmar, and Jon Crowcroft. "Classification of Twitter Accounts into Automated Agents and Human Users." In Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017, pp. 489-496. ACM, 2017.**

[39, 38] **Zafar Gilani, Reza Farahbakhsh, Gareth Tyson, Liang Wang, and Jon Crowcroft. "Of Bots and Humans (on Twitter)." In Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017, pp. 349-354. ACM, 2017.**

[37] **Zafar Gilani, Reza Farahbakhsh, and Jon Crowcroft. "Do Bots impact Twitter activity?." In Proceedings of the 26th International Conference on World Wide Web Companion, pp. 781-782. International World Wide Web Conferences Steering Committee, 2017.**

Andrés Arcia-Moret, Zafar Gilani, Arjuna Sathiaseelan, and Jon Crowcroft. "Peer

provided cell-like networks built out of thin air." In Consumer Communications & Networking Conference (CCNC), 2017 14th IEEE Annual, pp. 369-372. IEEE, 2017.

Sarim Zafar, Usman Sarwar, Zafar Gilani, and Junaid Qadir. "Sentiment analysis of controversial topics on Pakistan's Twitter user-base." In ACM DEV, pp. 35-1. 2016.

[41] **Zafar Gilani, Liang Wang, Jon Crowcroft, Mario Almeida, and Reza Farahbakhsh. "Stweeler: A framework for twitter bot analysis." In Proceedings of the 25th International Conference Companion on World Wide Web, pp. 37-38. International World Wide Web Conferences Steering Committee, 2016.**

Zafar Gilani, Arjuna Sathiaseelan, Jon Crowcroft, and Veljko Pejović. "Inferring network infrastructural behaviour during disasters." In Consumer Communications & Networking Conference (CCNC), 2016 13th IEEE Annual, pp. 642-645. IEEE, 2016.

Appendix C

Press, News and Print Media

This is a list of my research work covered by press, news and print media. The list only shows coverage by 1st hop entities (*i.e.* entities who covered me directly), and does not include others who picked stuff from elsewhere.

Celebrities Tweet Like Bots. In *Scientific American* 60-second Science podcast on Saturday, 5 August 2017.

Cambridge Study finds that Celebrity Twitter Accounts act like Bots. In *Digital Trends* on Sunday, 6 August 2017.

Celebrity Twitter accounts display 'bot-like' behaviour. In *University of Cambridge* Office of External Affairs and Communications on Wednesday, 2 August 2017.

Twitter 'Celebrity' Accounts Behave Like Bots, Not Humans, Study Finds. In *International Business Times* on Wednesday, 2 August 2017.

'Celebrity' Twitter accounts act like bots. In *The Hindu* on Wednesday, 2 August 2017.

Appendix D

Environment - Platforms, Systems, Resources, Dashboard

Given that there was a lot of work that studied a variety of related questions, it required to make available a number of different environments, platforms and systems. These are summarised below.

Platforms: Ruby, Ruby Gems (nokogiri, rest-client 1.1, thor, tree, mechanize, twitter 5.15, tweetstream, json, twitter_ebooks, shortener), Ruby on Rails, Embedded Ruby (ERB), Python, Python modules (numpy, scipy, sklearn, langdetect, textblob).

Systems: I used a desktop/workstation for data collection from the Twitter Streaming API as well as all of the processing involved. Figure D.1 shows the CPU utilisation during data processing workloads.

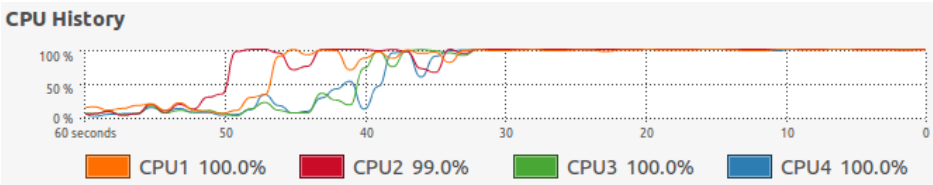


Figure D.1: A typical CPU workload graph during data processing.

I also used a VM in Cambridge University Information Services DMZ as a live Web server to deploy the Twitter bot[1] (for a honeypot experiment), a Web server to capture the alternate clicks dataset and a URL shortener. The Web server presents a dashboard[2] to display analytics around the clicks dataset (Figure D.2). Table D.1 shows the specifications of the two systems.

[1]The bot was non-invasive and did not engage in direct communication with Twitter users.
[2]***Stweeler*** dashboard – `http://svr-szuhg2-web.cl.cam.ac.uk/graph/graphs`

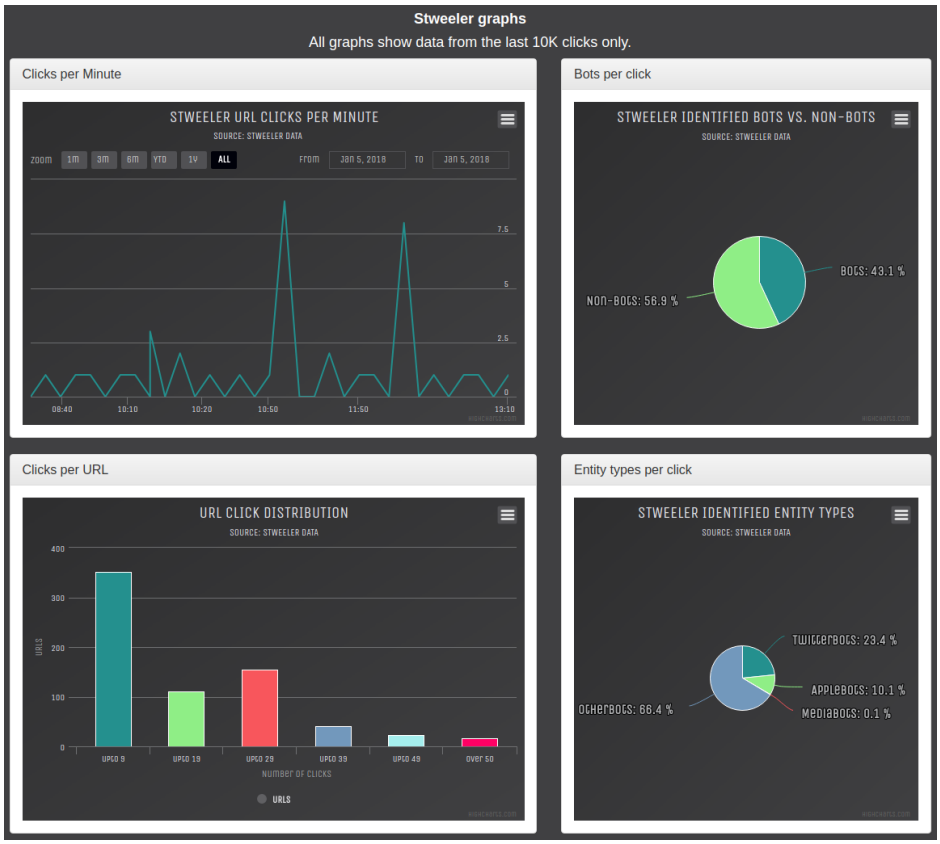


Figure D.2: ***Stweeler*** dashboard.

Resources: I used the University of Cambridge network to obtain data from the Twitter Streaming API. Figure D.3 shows a screen capture of the network utilisation during the typical data collection routine. The code for data collection is available here[3] as part of ***Stweeler***.

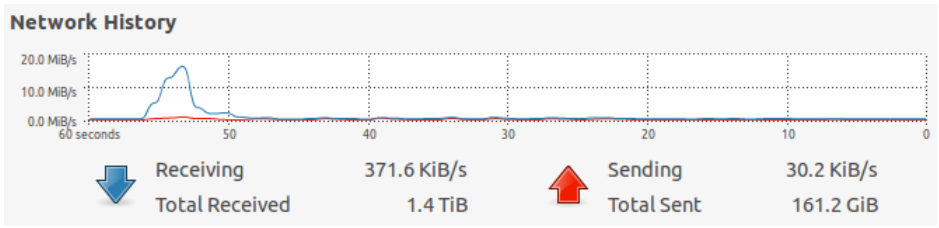


Figure D.3: A typical time graph during data collection.

Challenges: As briefly mentioned multiple times during the course of this dissertation, I used the Twitter Streaming API for collecting data on a daily basis. This

[3] ***Stweeler*** collector – `https://github.com/zafargilani/stcs/blob/master/lib/collector.rb`

Table D.1: System specification.

System	Specification
Desktop/Workstation	Ubuntu 14.04 LTS 64-bit 15.5 GiB Intel® Core™ i5-4690 CPU 3.50GHz 4 Intel® Haswell Desktop
Web Server	Ubuntu 16.04.3 LTS 64-bit 4.0 GiB Intel®Xeon®E5-2650L v3 @ 1.80GHz x 2 Intel® Haswell Desktop

constituted of 2.5 to 3 million tweets per day. I did not use any keywords, which let me collect everything that was available from the API. During the data collection process I encountered the following challenges: expiring OAuth tokens and keys, API errors, and local system failures.

I also deployed a Twitter bot as a part of the honeypot experiment, which was operationalised using the web server. During the operational life of the bot I encountered the following challenges: tweet rate limits, limits on following people, API errors, and occasionally passing two-factor verification by Twitter.