The Social Roles of Bots: Situating Bots in Discussions in Online Communities

JOSEPH SEERING, Carnegie Mellon University, USA
JUAN PABLO FLORES, Universidad Nacional Autónoma de México (UNAM), Mexico
SAIPH SAVAGE, West Virginia University, USA
JESSICA HAMMER, Carnegie Mellon University, USA

Bots, or programs designed to engage in social spaces and perform automated tasks, are typically understood as automated tools or as social "chatbots." In this paper, we consider bots' place alongside users within diverse communities in the emerging social ecosystem of audience participation platforms, guided by concepts from structural role theory. We perform a large-scale analysis of bot activity levels on Twitch, finding that they communicate at a much greater rate than other types of users. We build on prior literature on bot functionalities to identify the roles bots play on Twitch, how these roles vary across different types of Twitch communities, and how users engage with them and vice versa. We conclude with a discussion of where opportunities lie to re-conceptualize and re-design bots as social actors who help communities grow and evolve.

CCS Concepts: • **Human-centered computing → Empirical studies in HCI**; *Natural language interfaces*;

Keywords: Chatbots; bots; Twitch; social roles; moderation

ACM Reference Format:
Joseph Seering, Juan Pablo Flores, Saiph Savage, and Jessica Hammer. 2018. The Social Roles of Bots: Situating Bots in Discussions in Online Communities. In *Proceedings of the ACM on Human-Computer Interaction*, Vol. 2, CSCW, Article 157 (November 2018). ACM, New York, NY. 29 pages. https://doi.org/10.1145/3274426

157

1 INTRODUCTION

Bots deployed in online communities are a combination of tools designed to serve a specific purpose and social actors intended to join conversations [25]. They are designed to do many things that humans had previously done or would otherwise do, but are also created with social elements in mind - their own names, grammatical styles, and sometimes even personas. Many are run alongside rather than within sites, a concept that Geiger calls "Bespoke Code" [12]. In certain situations, bots can create and contribute their own content to a community [8] in patterns designed by their creators. To the outside observer, limited affordances on certain sites allow cleverly designed bots even to pass as human for extended periods of time [1]. Beyond their variable humanity, bots give us new capacities, particularly by amplifying our efforts in speed or scale. They help us collect data [40], attract attention for a cause [36], and detect and respond to certain behaviors [13]. At scale, bots can even help shape or sway political conversations [1].

Authors' addresses: Joseph Seering, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA, 15213, USA, jseering@cs.cmu.edu; Juan Pablo Flores, Universidad Nacional Autónoma de México (UNAM), Mexico City, Mexico, jp@juanpabloflores.me; Saiph Savage, West Virginia University, West Virginia, USA, saiph.savage@mail.wvu.edu; Jessica Hammer, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA, USA, hammerj@cs.cmu.edu.

2573-0142/2018/11-ART157 $15.00
https://doi.org/10.1145/3274426

In this paper we explore the variety of types of social actions that can be taken by bots on Twitch[1], a growing platform that reports more than 140 million unique users[2] and hosts thousands of diverse communities. As we find in our analysis, Twitch bots are extremely active participants in Twitch chatrooms; though the overwhelming majority of accounts that participate on Twitch are human, the average bot sends well over an order of magnitude more messages than the average human, making them important actors in the space. While recent research has found similar patterns of participation for both malicious and user-invited bots in spaces like reddit and Twitter [8, 35], Twitch is both a new context for the study of bots and a space with a different conversational structure than Twitter or reddit, with bots participating in a public chatroom rather than a threaded conversation or Twitter network. We show here Twitch bots' importance to the social discourse on Twitch both through analysis of their rates of participation across different types of communities and through description of the types of engagement they have with their communities.

In order to guide our exploration of the roles of bots in Twitch communities, we aim to answer the following research questions for a number of different types of communities on Twitch:

(1) How frequently do bots send messages on Twitch?
(2) What kinds of language do bots on Twitch use, and for what purposes?
(3) What do bots and users say to each other?

The first of these questions addresses whether bots on Twitch are active participants in social spaces, placing their social dynamics in conversation with prior work [8, 35]. The second question supplements previous work that identified patterns of bot language in different contexts, e.g., a particular subreddit [8]. While some work has looked at users' desired features for bots in community-style social platforms [28], most language-specific analysis of bots has been done on networked spaces like Twitter, or specific spaces on a single platform. Here we analyze the full breadth of communities on Twitch, identifying ways that bot usage varies across types of communities. The third question addresses the complexity of human-bot interactions on Twitch. Prior work has found that bots have had a variety of roles, both in completing particular automated tasks (e.g., [40]) and in holding human-like conversations with users (e.g., [42]), and in this work we aim to situate Twitch bots in this space. We answer each of these three questions across different sizes of communities and across communities built around different types of content, identifying how community characteristics contribute to bot behaviors.

We begin this paper with a brief description of the Twitch platform and then review relevant work. Next we describe prior research that has identified roles among humans online, focusing specifically on the methods that have been used. Third, we briefly review some of the literature on chatbots online and consider how the methods used to study human roles online might be applied to the study of bots. Following our review of the literature, we identify the functions for which bots have been developed on Twitch and present example interfaces for management of the two most popular and oldest third-party bots on Twitch, which have set the standard for bot designs. In the main section of this paper we quantify the influence of bots on conversations using methods influenced by structural role theory [4] and based on prior analysis of human roles to identify what bots types of messages send, how frequently they send each type, and how much they interact with regular users. We conclude with a discussion of opportunities for further development of bots' social roles on Twitch and beyond.

[1] www.Twitch.tv
[2] http://twitchadvertising.tv/audience/

2 TWITCH AND VIDEO-STREAMING "CHANNELS"

In a Twitch community, called a "channel", one or more video streamers perform via live audio-visual while spectators communicate with them and each other via a chatroom built on classic Internet-Relay-Chat (IRC) mechanics. Spectators comment on the content of the video stream and sometimes engage in conversation with the streamer. Streamers' performances can be creation of art or music in real-time, but can also be anything from exploring a city with a video camera broadcasting from their shoulder to playing computer games. Streamers have creative control of their channels, deciding what content to stream, making moderation decisions, and adding bots. Some streamers even use Twitch as a primary revenue source, earning money through advertisements, sponsorships, viewer donations, and subscription fees [22].

Though Twitch is a relatively young platform, introduced in its current form in 2011 and purchased by Amazon in 2014, it has already been the subject of much research in computer science and media studies communities. For example, in early ethnographic work on Twitch, Hamilton [19] found that Twitch channels can be meaningful social spaces where users come to hang out, learn, or make friends. More recent work found that users join Twitch channels for various reasons, from tension release to building social connections to gathering information [39], and that many users give substantial social and even financial support to streamers [44]. Twitch has also attracted attention from technical researchers, with research exploring network performance in the context of live-streamed video [24], video stream transcription [34], and development of new interfaces for audience participation [16].

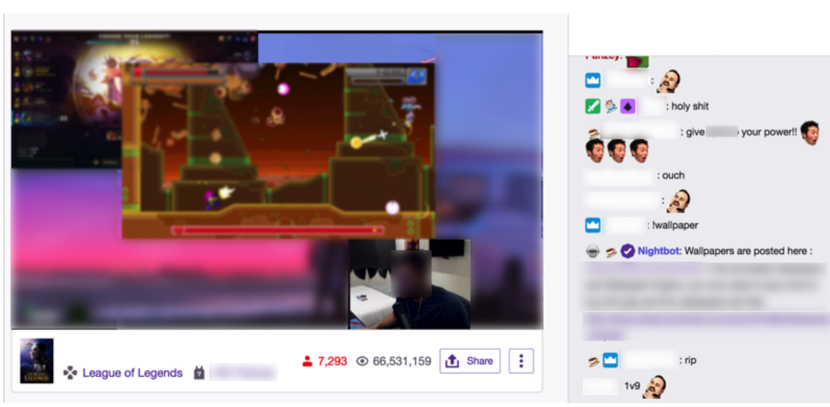

Fig. 1. Example Twitch stream interface (usernames redacted)

Figure 1 shows an example of a Twitch channel interface. The streamer in the channel is visible via a webcam shown in the lower-right portion of the stream. He is shown playing a game, and users in the chat window on the right side of the figure are actively discussing it with him and with each other. Note that one user has typed "!wallpaper", which is a type of interaction that will be discussed later in this paper called a bot command. In this case, the bot present in the channel, "Nightbot", recognizes the command and provides information about the streamer's wallpaper.

Bots in this environment are clearly present - they are seen in exactly the same way as any other user when they send a message, as seen in Figure 1. They also have the same chat capabilities as any other user; any type of message that a user sends in the chat can also be sent by a bot. Though limited by their programming, bots on Twitch do not have platform-enforced restrictions on their access to participate in the chatroom. While sites like Facebook heavily restrict the ability for automated tools to access the site and interact with users, Twitch bots are easy both to develop and to integrate within a stream community. A number of third-party developers have created bots with a variety of features on Twitch, and at the time of writing, a YouTube search for "make Twitch bot" returned more than a dozen tutorials from different sources teaching viewers how to build their own Twitch bots in a variety of programming languages. In this paper we explore the different type of bots that co-exist on Twitch.

3 PRIOR WORK

In informing our understanding of the roles bots play on Twitch, we focus on two major areas: literature on *roles*, both from traditional social science and modern computational approaches, and literature on the different *uses of bots* online.

3.1 Roles

Role theory is an established paradigm that has been used by social psychologists and sociologists for nearly a century. Beginning with George Herbert Mead, Jacob L. Moreno, Talcott Parsons, and Ralph Linton, its most recent primary voice has been B.J. Biddle [3, 4]. Per Biddle, a number of subtheories have emerged within role theory that consider roles from different perspectives, including functional [27, 33], symbolic interactionist [29], structural [26, 31], organizational [18, 23], and cognitive [3] role theories. Rather than competing to most accurately predict behaviors, these theories provide alternative perspectives from which to explore roles [4, p. 75-76].

Though other lenses may certainly be useful, in this paper we use structural role theory as a lens that informs our analysis and methods for understanding the ways bots are used on Twitch both because it has been used in previous work on human roles online [45, 46] and because it matches the traditionally-functional conception of bots. Per Biddle [4],

> [S]tructural role theory is focused on "social structures," conceived as stable organizations of sets of persons ... who share the same, patterned behaviors ("roles") that are directed towards other sets of persons in the structure. Such concepts lead to formal discussions of various concerns including social networks, kinships, role sets, exchange relationships, comparison of forms of social systems, and the analysis of economic behaviors (p. 73)

In brief, roles are patterns of interactions that different types of actors in a system display toward each other, generally with the goal of furthering the ends of the community. Biddle notes that this framing concerns itself more with mathematical analysis of behavioral patterns than with explorations of norms.

Structural role theory fits the study of bots neatly because they are designed more or less in this exact same way - while norms certainly impact their development, bots are designed to behave and interact with users in specified, structured ways to achieve certain goals within a space. Most bots' behaviors change over time only in the sense that they may be upgraded to better perform their assigned tasks or additional capabilities may be added, fitting structural role theory's focus on stable patterns of interactions. Long et al. [28] note that when reddit users ask other users to make bots for them, their requests are based on desired functionality. Thus, it makes sense to analyze bots' roles within communities based on their functionalities.

3.1.1 Analyzing roles online. Primary means of detecting roles have been computational, though some more recent work has integrated initial qualitative steps. We note four general approaches in the literature: *initial qualitative investigation*, *textual analyses*, *network analyses*, and *action analyses*. Of these, the latter two are most common, though the increasing presence of linguistic analysis and machine learning online have facilitated some textual analysis in recent years.

In early work on roles online, Fisher, Smith, and Wesler [7] identified roles in Usenet groups by looking at network structures of replies, noting how different newsgroups had different structures of replies and thus could be classified as having different social structures based on their topic. These authors further expanded on mathematical models for specific role identification in [43]. Gleave et al. [15] built upon this approach by combining initial qualitative inquiry with a network approach. Gleave et al. is also the first work in this area to cite formal Role Theory [3, 4]. Using similar methodological approaches, Wesler et al. [43] identified four social roles in Wikipedia beginning with qualitative analysis and moving to network analysis. This network modeling approach was subsequently used in [21] and [47] on an email corpus and a social network respectively.

More recently, Bamman, O'Connor, and Smith [2] took an alternative approach to identifying identifying roles in film summary text, using textual analysis paired with use of metadata, identifying what characters do, what is done to them, and how they are described. Ferschke, Yang, and Rosé [6] advanced text-based conversational role modeling further by identifying prevalence of text-based labels in Wikipedia users' Talk-page conversations, working with a less-formal conversational structure than in standard discussion forums. Yang et al [45] performed a similar task on Wikipedia using users' action histories, also building on Role Theory via Biddle [3], adding four additional roles to those established by Welser et al [43].

In this work we aim to use lessons from prior work in human role analysis to approach the novel challenge of analyzing bots' roles in communities. Formal analysis of roles has thus far been reserved primarily for humans, an implicit (and perhaps unexamined) assumption upon which role theory rests. However, we suggest that bots can play as much of a role in online systems as humans, in particular when they are visible to human users, when they can speak, and when they can act on other users, often through moderation features.

3.2 Bots

Early bots, whose legacy is visible in many bots today, were a combination of function automation and "chatbots", computer programs designed to hold a conversation with humans. The famous first iteration of a "chatbot" in conversation is ELIZA [42], which allowed very basic "conversation". Early work on online bot automation [32] noted the early use of "(ro)bots" in Internet-Relay-Chat channels to automate certain moderation functions and to respond to commands or send messages at routine intervals. Many of these same functions are now seen in Twitch bots, likely because Twitch is built on the same IRC technology and can be accessed with standard IRC code packages. More recently, [40] discussed a more elaborate, outward-facing form of automation through bots as companion tools, automatically crawling portions of the web as directed for archival purposes.

The functionalities of bots have been expanded in recent years, particularly with the proliferation of social media, evolving to actively engage users, provide information, automate moderation and governance tasks, and participate in gameplay. With "Botivist", [36] researchers demonstrated the potential for use of bots to call volunteers to action in a political context. Geiger and Ribes [13], in their analysis of vandal-fighting on Wikipedia, made the argument that bots can do more than automate or force-multiply; they can transform a process. Bots have also been developed to play online games, though perspectives are split on whether this is an opportunity for creativity [30] or harmful and disruptive [14, 17]. Recent work in Audience Participation Games has considered deeper roles bots might have in gameplay on Twitch, but this work is still in its infancy [38].

Following Geiger's call for deeper understanding of the social foundations of bot development [11], Long et al., present a comprehensive exploration of what tasks bots are intended for, focusing on requests to bot developers on three reddit communities (subreddits) dedicated to the discussion, requesting, and creation of bots with specific features [28]. They identify five primary "Issues" that inspired reddit bot requests: *Administration*, *Archiving*, *Community*, *Functionality/Quality*, and

Play/Humor. Of these, Archiving follows Summers and Punzalan's description of bot archiving [40]; Administration and Community describe bots that make contributions to automated moderation and governance tasks; Functionality bots provided useful information upon request; and Play/Humor bots facilitated jokes or gameplay. A notable missing type of bot here is bots for *self-promotion or advertising*, likely because of the strong ethos (and rule) on reddit against such activities. These task categories are not limited to bots designed for reddit. For example, political bots, as per e.g., [1], can be designed to promote causes or political groups, spread information or disinformation, or engage users for a particular cause, in similar fashion to those discussed above. We therefore use these categories to inform our analysis of bots on Twitch.

4 BOTS ON TWITCH

We present here a brief overview of established bots on Twitch. First, we identify features of a number of commonly-used bots on Twitch, all of which appeared in the data we collected. We use examples of interfaces from "Nightbot", "Moobot", and "Streamelements", three popular third-party bots, to illustrate common feature categories and to show how streamers interact with their bots. Any streamer can add any of these bots to their channel for free and customize them to a limited degree.

We began our inquiry into Twitch bot features with a spreadsheet maintained by Twitch power users[3], which lists common features of widely-used Twitch bots. We searched for and examined documentation provided by the bots' developers in order to verify the accuracy of this spreadsheet and identify additional features that might not have been captured. We also added a feature list for one bot that we had previously observed, "Streamelements", which was not included in the spreadsheet. The full list of features for all major bots observed in our sample is shown in Appendix C, but we summarize major categories of features below. Note that we do not present this as novel work, as it builds on significant prior work from the aforementioned reddit users, but rather summarize it as an introduction to bots on Twitch.

As shown previously in Figure 1, bots' primary mode for engagement with users is in channel chatrooms. Figure 2 shows what a standard bot looks like in a chat.

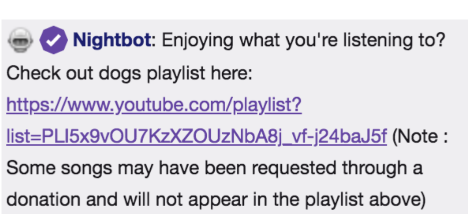

Fig. 2. A message posted by Nightbot to a chat

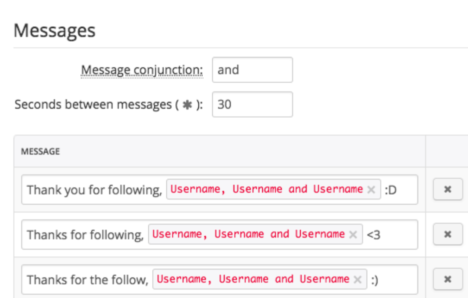

Fig. 3. Moobot interface for editing welcome messages to new followers

The first set of features is based on the concept of engaging new members and recognizing contributions. Bots can identify new followers and new subscribers to a stream and can alert the streamer when a user has donated money to them. These bots can also publicly welcome or thank these users in the chat. Figure 3 shows Moobot's interface for editing welcome messages. Some

[3] https://www.reddit.com/r/Twitch/comments/4qcsfq/an_updated_twitch_bot_list/, created by reddit user u/LordNazo

bots also recognize long-standing members of a community through a loyalty system, tracking users' time spent in the channel and giving them stream "points" accordingly.

Next, bots can be set to filter certain types of content from the chat. Available content types for filtering include disruptive but easy to detect behaviors, from "excessive use of caps" to banned words to unauthorized posting of external links links to spamming excessive symbols or repeated messages. They can also make special exceptions for regular channel members, who may be trusted enough to permit them to post non-malicious links. Figure 4 shows the Moobot interface for selecting which content to filter.

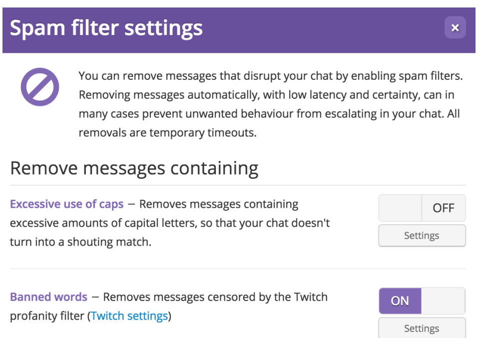

Fig. 4. Moobot interface for setting chat filters

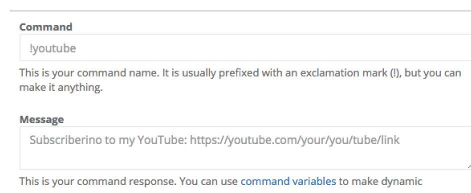

Fig. 5. Nightbot interface for adding informational commands

Third, bots can also be set to share information with users via commands. Figure 5 shows an interface for setting up these types of commands, which are phrases users can type to get a specified response from the bot.

Finally, bots can run various side-entertainment features, from allowing users to request songs to be played to running mini-games in the chat to managing raffles for prizes. Figure 6 shows an interface for adding games to a channel.

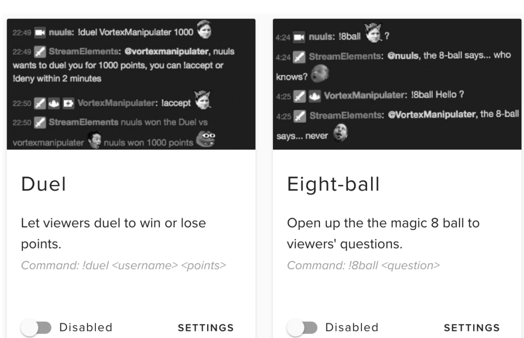

Fig. 6. Streamelements interface for adding mini-games to chat

The features mentioned above comprise the full variety of available features for bot engagement with viewers in major bots on Twitch, and in our analysis we found that custom bots mostly followed these models as well. All of these functionalities can be created by users manually via scripting; none require special access to Twitch. Note that these bots have a number of additional behind-the-scenes logistical features to help the streamer manage logistics, but we do not discuss these in depth here because they do not affect the behavior of the bot in the chatroom.

As discussed in Section 6.2 of this paper, these categories closely match the that emerge from our qualitative analyses of bot behavior, which mirror many of the functionalities discussed in prior literature. The core capabilities of bots on Twitch are similar to capabilities identified in prior research, e.g. on Wikipedia [13], Twitter [1], Usenet [7], and IRC [32].

5 METHODS

We work here from a primary dataset of 7,143,563 messages collected from 125 publicly-viewable Twitch channels over the course of 104 hours in April 2018 using an IRC-based script [37]. Of these, 115 were active during message collection. All messages sent in these channels were collected. Because the channels were all of different sizes and were active for different lengths of time, a different number of messages was collected from each. See Appendix B for a full list of messages sent by both human and bot users in each channel. The median number of messages collected in a channel was 22,415, with mean of 57,149. The mean is significantly higher than the median because of the skewed nature of Twitch channel viewer numbers and our decision to sample across multiple size categories; Figure 7 shows an example snapshot of Twitch viewership numbers through a log-log histogram. There are a small number of very large and very active channels, and many much smaller channels.

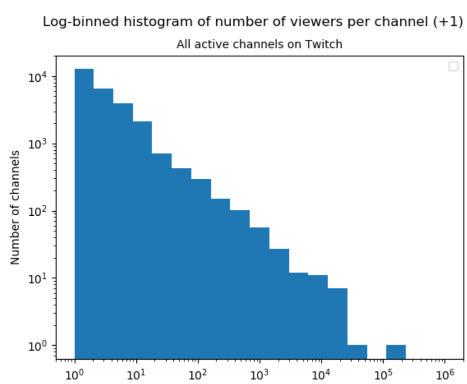

Fig. 7. Log-log plot of sample Twitch viewership distribution

The scraping captured username, channel where the message was posted, message text, timestamp, and type of user (moderator, subscriber, or regular user). This scraping is permitted by Twitch Terms of Service and also aligned with community norms; the bots we studied here use the same protocol and similar scripts to scrape and interact with user messages, and some save the type of chat logs that we used for our analysis (see Table 14). This research was also approved by the Institutional Review Board at Carnegie Mellon University.

In order to choose which channels to include in our sample, we ran a pre-scrape of all channels on Twitch. We used Twitch's API to scrape data from all English-language streams, collecting stream-level metadata including the stream's name, the game currently being played or activity being done, the title of the stream, and the size of the audience.

The 125 channels included in the sample were selected to represent five different categories of community size and four different types of content (see Table 1). While prior work has assigned cutoffs for channel size based on observation (e.g., [19] defined small audiences as involving less than 1,000 audiences members, while massive audiences had over 1,000 individuals), we aimed to find emergent groups of channel sizes through clustering analysis of audience size. The size categories used were developed by use of a mean shift algorithm[10] to group channels with similar sized audiences. For each channel, we calculated audience size as the average number of live viewers the channel presented daily throughout our pre-scraping period. We used a mean shift algorithm to group streams with similar sized audiences. We opted to use a mean shift algorithm because it is based on a non-parametric density estimation, and we would not need to know the number of clusters beforehand (unlike K-means). Our clustering algorithm identified five different audience scales, i.e., clusters. These are shown in Table 1.

While originally a site dominated by gaming content, Twitch has broadened its focus in recent years to include a much wider variety of content. As a supplement to the size categories described above, we identified four major categories of streams, each of which was determined by metadata in the pre-scrape: Gaming streams, which focused primarily on playing different computer, console, or tabletop games; Creative streams, which centered around the production of art or music; IRL streams, in which a person documented events in their lives; and Talk shows, where individuals or panels discussed specific topics. The latter three categories are taken directly from the scraped metadata of streams, while the "Gaming" category is an amalgamation of streams tagged as playing different games. Together, they account for nearly all types of streams on Twitch; two minor exceptions include "Music", which is much smaller than "Creative", and rare special events streamed on Twitch like TV show marathons. These content categories are also shown and described in Table 1.

For our final sample, we drew 25 streams from each of the five size groups, with the exception of the largest group where only nine streams were present during the time period we collected data; though the largest streams on Twitch account for a significant volume of views (see Figure 7) and comments, there are relatively few of them. We also drew 25 streams from each content category. Note that stream size category and content category could not be perfectly balanced in our sample - for example, there are no Creative streams on Twitch that reach size category five with any regularity, so the average Creative stream in our sample was smaller than the average Gaming stream. These differences are reflected in the message volumes shown in Table 1.

To supplement our data, we performed a qualitative analysis of a subset of 100 streams selected with this same methodology, which we detail in Section 6.2. We spent 15 minutes viewing each of these streams and took notes on the presence of bots, what bots did and said, and how users interacted with bots.

6 ANALYSIS OF BOT ROLES

As noted above, we identified certain users as bots starting from a list of major bots maintained by Twitch power-users.[4] We used the above list as a guide in order to identify other, less widely-used bots, also using qualitative observations of 100 Twitch channels that we discuss in more depth in Section 6.2 to find bots that might vary from established features used by major third-party bots. Thirty-one bots were identified overall, of which twelve appeared in our dataset. Of these twelve, six were standard, freely-available bots, and six were custom-made for their channel. Note that we make no attempt here to identify bots that were designed to pretend to be human to deceive users.

Prior to any comparative analysis of bots and humans, we first compared the distributions of human and bot message sending across all humans and bots in each of the five size categories and four content categories. We used a Kolmogorov-Smirnov test (KS test), which is a nonparametric test comparing the equality of two distributions. For each category, we compared the distribution of the numbers of messages sent by human vs bot users. Tables 2 and 3 show the results of these tests

[4] https://www.reddit.com/r/Twitch/comments/4qcsfq/an_updated_twitch_bot_list/

Table 1. Types of streams scraped

	Classifier	Characteristics	# of Messages
Size categories	Category 1	0-6 average concurrent viewers	1597
	Category 2	6-1879 average concurrent viewers	561308
	Category 3	1879-7703 average concurrent viewers	1213524
	Category 4	7703- 21,678 average concurrent viewers	2555629
	Category 5	21,678+ average concurrent viewers	1341413
Content categories	Creative	Focused around the creation of art, music, or performance	89271
	IRL (In-Real-Life)	Following an individual or group during various life experiences from assembling furniture to traveling around the world	892083
	Talk show	Individual or panel focused on discussing a particular type of content	77200
	Gaming	Streams built around streaming gameplay, whether of computer, console, or tabletop games	1641402

for the size category and content category groups respectively. We found significant differences in distributions of messages sent in all four content categories and the four largest size categories. We did not find a statistically significant difference in the smallest size category, likely because relatively few messages were collected (1597, of which 183 were sent by bots). Because of this, we exclude Size Category 1 from our remaining analyses. In order to visualize the distributions of messages sent per hour by each user type, we created log-binned plots for each of the nine channel types. Figures 8 and 9 show examples of these. All nine plots can be found in Appendix A.

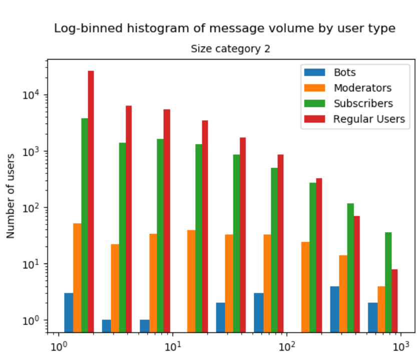

Fig. 8. Distribution of number of messages sent per hour by user type in sampled size category 2 channels

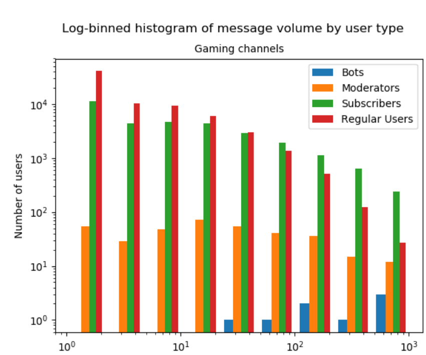

Fig. 9. Distribution of number of messages sent per hour by user type in sampled gaming category channels

Table 2. KS test statistics comparing human and bot messaging distributions across size categories

	N(Bots)	N(Humans)	KS test statistic	p-value
Category 1	3	162	0.33	0.83
Category 2	22	53963	0.71	<0.001
Category 3	14	136414	0.93	<0.001
Category 4	13	232108	0.93	<0.001
Category 5	8	179645	0.92	<0.001

Table 3. KS test statistics comparing human and bot messaging distributions across content categories

	N(Bots)	N(Humans)	KS test statistic	p-value
Creative	10	10284	0.61	<0.001
IRL	29	86959	0.83	<0.001
Talk show	14	5730	0.66	<0.001
Gaming	15	103099	0.89	<0.001

6.1 How much do bots "talk"?

In order to answer our first research question, "How frequently do bots send messages on Twitch?", we analyzed volumes of messages sent by user type. Across our whole dataset, human users sent 6,999,749 messages and bots sent 143,814, a ratio of 48.5:1. Tables 4 and 5 illustrate this discrepancy across the size categories and content types. For example, in across all of the Category 3 size streams scraped, 3.18 messages were sent per second by humans and 0.06 were sent per second by bots. For every 53 messages sent by humans, there was one bot-written message.

Table 4. Messages per second by size

Chat messages per second by user type and cluster size				
	Humans	Bots	Ratio	# of Messages
Category 2	1.46	0.04	38:1	561308
Category 3	3.18	0.06	50:1	1213524
Category 4	6.71	0.12	58:1	2555629
Category 5	3.50	0.09	40:1	1341413

Table 5. Messages per second by category

Chat messages per second by user type and content category				
	Humans	Bots	Ratio	# of Messages
Creative	0.23	0.01	33:1	89271
IRL	2.33	0.05	43:1	892083
Talk show	0.20	0.00	50:1	77200
Gaming	4.29	0.09	46:1	1641402

If considered alone, this data might suggest that bots' impact on conversations is relatively low. Our second analysis, however, focused on the volume of contributions that each user provided. Tables 6 and 7 show the number of messages sent per user, split out by category of user, per hour of scraping. Note that here we break human user categories down into three groups - regular users, who have free accounts and no particular status or authority, subscribers, who pay $4.99 monthly to have special privileges in the channel, and moderators, who are designated by the streamer to enforce rules in the stream. In particular, we find it useful to compare bots with moderators, as both are given authority by the streamer.

Table 6. Messages per user per hour by size

Chat Messages per user per hour by cluster size				
	Bots	Moderators	Subscribers	Regular Users
Category 2	6.36	0.56	0.21	0.07
Category 3	16.50	0.82	0.23	0.05
Category 4	32.09	1.43	0.27	0.07
Category 5	38.91	0.66	0.08	0.06

Table 7. Messages per user per hour by category

Chat Messages per user per hour by content category				
	Bots	Moderators	Subscribers	Regular Users
Creative	2.51	0.62	0.18	0.07
IRL	6.79	0.97	0.34	0.06
Talkshow	1.04	0.58	0.17	0.10
Gaming	22.42	0.82	0.31	0.07

Analysis of volumes of messages sent per user resulted in the finding that bots sent far more messages than the average individual user across every type and size of community. For example, we observed 546,327 messages in Gaming streams from 71,205 unique regular users across the 104 hours of data collection, resulting in a per-user average rate of 0.07 messages per hour. In contrast, we observed 34,975 messages sent from 15 unique bots during this period, resulting in a per-bot average rate of 22.42 messages per hour, more than three hundred times higher. Bots sent more messages per hour on average than all types of human users across all channel categories, and moderators sent more messages than subscribers who sent more messages than regular users.

The increase in relative bot message prevalence as stream size increases (Table 6) likely stems from two factors. First, bots in streams are intended to perform tasks that humans do not want to do or do not realistically have time to do. As the rate of incoming messages becomes harder for humans to handle, more tasks and thus more messages are delegated to bots. Second, prior work finds that individual participation declines as groups get progressively larger, but bot design permits participation on a fixed schedule; regardless of whether there are 10 or 10,000 users in a channel, bots can be scheduled to post the same message every 15 minutes.

The relatively higher prevalence of bot messages in Gaming and IRL channels (Table 7) may relate to the structure of the streamer's activity. In talk shows and creative streams, streamers are often constantly directly engaged with the chat, answering questions as they work or discuss. Talking to the audience does not compete with their streaming activity, as both creative work and talk shows allow the streamer to set the pace. On the other hand, in IRL streams, streamers may be engaged in activities that require their full attention. Games, which have the highest bot prevalence, not only require significant streamer attention, but also often do not allow the streamer to control the pace of their game activities [5]. Even in single-player games, game time may advance independently of the player's game choices; in multiplayer games, particularly the competitive games so popular on Twitch, lapses in the streamer's attention can give opposing players an advantage. We hypothesize that bots are used as a substitute for streamer attention when possible, for example to answer common questions about what they are doing.

Two caveats to this analysis actually make it significantly conservative; first, this analysis only captures users who sent at least one message during the time period. Users who never post messages or post only once a month would not have been captured. As such, the rate of messages sent per hour that we report from regular users here (and to a lesser extent subscribers and moderators) is probably significantly higher than the reality, making the gap between bots and humans even wider. Second, we count different instances of the same bot as unique bots; for example, Nightbot was present in 54 of the 125 streams in our sample. We elect to count this as 54 different bots rather than a single bot because users can only interact with one instance of the bot at a time; they cannot ask a bot in one channel for information relevant to another, and they only see messages from the instance present in their community. Also, each instance of Nightbot was customized to fit its community in at least a small way, as links to social media profiles and answers to commonly asked questions would naturally be different. This approach is quite conservative; had we collapsed these into a single user, the bot messaging rates that we report here would have been approximately eight times higher, further increasing the discrepancy. Note that to ensure equivalent analysis, we also treated users appearing in multiple channels as unique users, but this had relatively little effect. More than 95% of users only posted in one of the sampled channel during our data collection window.

Ultimately, while bot messages are not a huge proportion of messages sent overall, individual bots send messages at a rate much higher than regular users, subscribers, or even moderators. Particularly in large streams, gaming streams, and IRL streams, users will see bots in the chat far more than the average human user. Bots have disproportionate influence on the tone of the chat if only by volume of text, and in an environment like Twitch where imitation effects are notably strong [37], this volume of messages has impact.

6.2 What do bots "say"?

The literature on social roles discussed previously used four methods to identify roles: *initial qualitative investigation*, *textual analyses*, *network analyses*, and *action analyses*. We used a combination of qualitative investigation of bot messages and large scale textual analysis to answer our second research question, "What kinds of language do bots on Twitch use, and for what purposes?". We select these two methods because text is the most visible indicator of how bots interact with their communities; Twitch does not have a substantive network shape like Wikipedia or Twitter, where previous role identification has been performed, nor is the chat threaded like traditional forums.

In order to provide a cohesive classification for types of bot messages, we first qualitatively analyzed 100 different Twitch channels for 15 minutes each. Of the 100 channels, bots were active during observation in 63. The most common bots were Nightbot and Moobot, discussed above; Nightbot was present in 25 of these channels and Moobot was present in 17. During channel observation, 11 channels showed evidence of the use of two distinct bots simultaneously, but no channel used more than two bots.

Combining analysis of bot messages collected from these channels along with notes on their context with categories identified in the work discussed above [28, 30, 32], we grouped bot messages into five categories: *Sharing Information*, where bots answered questions via commands about things like what a streamer was doing or how long they had been streaming; *Explaining Moderation*, where bots posted a message about the rules, chastised a particular user for behaving in a detected way, or explained a warning or timeout; *Engaging Users*, where bots participated in a limited manner in the community social dynamics by spouting pre-set meme phrases, linking to amusing videos, or welcoming new users; *Running Mini-Games*, including simple bot-run “point systems”, a common reward for loyalty where users are given points for each minute or hour they spend in a chat which can be spent on simple in-chat games like roulette; and *Promoting the Streamer*, where bots advertised streamers’ social media accounts communities on other sites, or provided links to donate or subscribe to the streamer. This last category is least present in the literature, which has not yet explored use of bots for pro-social advertising in social spaces. Unlike in research on spam bot detection [41], Twitch bots’ advertisements are usually accepted by users as they benefit the streamer and may allow for deeper engagement. See Table 8 for example messages from each category. These categories also relate closely to the options listed in Nightbot, Moobot, and Streamelements customization menus discussed previously.

Table 8. Example comments by category

Example comments by category	
Sharing information	Find the unfinished rebroadcast schedule for the weekend here: [link]
	The time in Singapore is: 22:08:17
Explaining moderation	No long messages allowed ([user]) (warning)
	Only subs can post links without being permitted! [warning]
Engaging users	BibleThump NO FRIEND BibleThump PLEASE COME BACK BibleThump WE WERE HAVING FUN BibleThump
	[User] has just subscribed to the stream! You also unlocked !subperk
Running Mini-games	[User] pays 10 coins to play the Slots, and... won nothing!
	Security is normal. You may attempt to hack [Streamer’s] bank by typing !gamble and the amount to invest.
Promoting the streamer	[Streamer] has a merch store with comfy sweaters and t-shirts! Check it out at [link]
	Make Sure to Follow me on Twitter! [Twitter link] Stay Updated!

Next, we developed a classifier to classify bot messages into these five categories at scale. We used the XGBoost module in Python to develop this classifier based on simple bag of words features, training the model on a set of 1150 hand-classified messages and using a five-fold cross-validation to evaluate performance and tune parameters. XGBoost is based on the boosted trees model [9], a variant on traditional decision tree modeling.

When tested on an additional set of 160 hand-classified messages, this model achieved an 84% accuracy with $\kappa = 0.78$ and overall weighted F1 score of 0.80. Individual F1 scores were highest for Promotion and Moderation, at 0.90 and 0.91 respectively, moderate for Engagement and Games at 0.78 and 0.79 respectively, and lower for Information at 0.71. Promotion and moderation are likely easiest to identify because promotion messages frequently contained keywords related to social media channels, e.g., "Follow me on Twitter!", while Moderation messages contained punishment related words, e.g., "warning" or "timed out". Engagement took a variety of forms from welcoming users to spouting meme phrases, and the former were quite easy to identify based on keywords like "welcome" while the latter were highly variable. Games had a number of core patterns, like references to "points", but also took custom forms in certain channels that were difficult to identify. Informational messages were the hardest to detect accurately because information was specific to each channel and, per the interfaces discussed above, many pieces of information had to be entered manually rather than via a template.

A number of clear sub-classes emerged from examination of this data, though we did not explicitly test for them. Moderation actions could be separated into pre-punishment warnings and post-punishment explanations. Engagement actions could be parsed for welcoming phrases, possibly to build on prior work looking at retention of newcomers [20]. Informational messages could be separated into responsive messages where a bot reacted to a question from a user and repeated, timed messages where bots posted regular information about the stream or streamer. Both self-promotion and promotion of friends were present in the dataset, and these could also be worthwhile to differentiate.

The ability of such a simple model to achieve very good accuracy overall can be explained by the simplicity and homogeneity of bot messages on Twitch; though users can customize messages freely, most follow standard templates or use common language. Therefore, decision trees are likely to model Twitch bot messages effectively and without overfitting because of the utility of simple word-based rules. Figures 10 and 11 show the prevalence of each type of message on Twitch, separated out by size groups and categories.

Though bot message content patterns are very similar across size categories, we find major differences in messages across content groups; nearly three quarters of messages sent in Creative streams were of the Running Mini-Games message type, while less than one quarter of messages in IRL streams fulfilled this function. We suggest that mini-games may help fill slow periods in Creative streams, as tasks are not always as purposefully continuously engaging as in Gaming, IRL, or Talk show streams. Gaming streams also often have down-time between matches, perhaps explaining the similarly high prevalence of Mini-Game messages.

Creative streams had the lowest percentage of messages where the bot was tasked with promoting the streamer. This may be due to a lower amount of external infrastructure established by creative streamers, or it may be due to the fact that the means for supporting the streamer are more clear. They are, through their creative work, directly creating a product that can be sold, while Gaming, IRL, and Talk show streamers must rely on ads, subscriptions, and donations.

6.3 What do users and bots say to each other?

In our final exploration of bots' roles, addressing our third research question - "What do bots and users say to each other?" - we attempt to find places where bots and users converse. We first analyzed how many times in our dataset users' messages contained "@" messages directed at bots. Typing "@" followed by a user's name is a standard convention for getting their attention in Twitch chat because the message is then highlighted only on their screen. This tactic is often used in

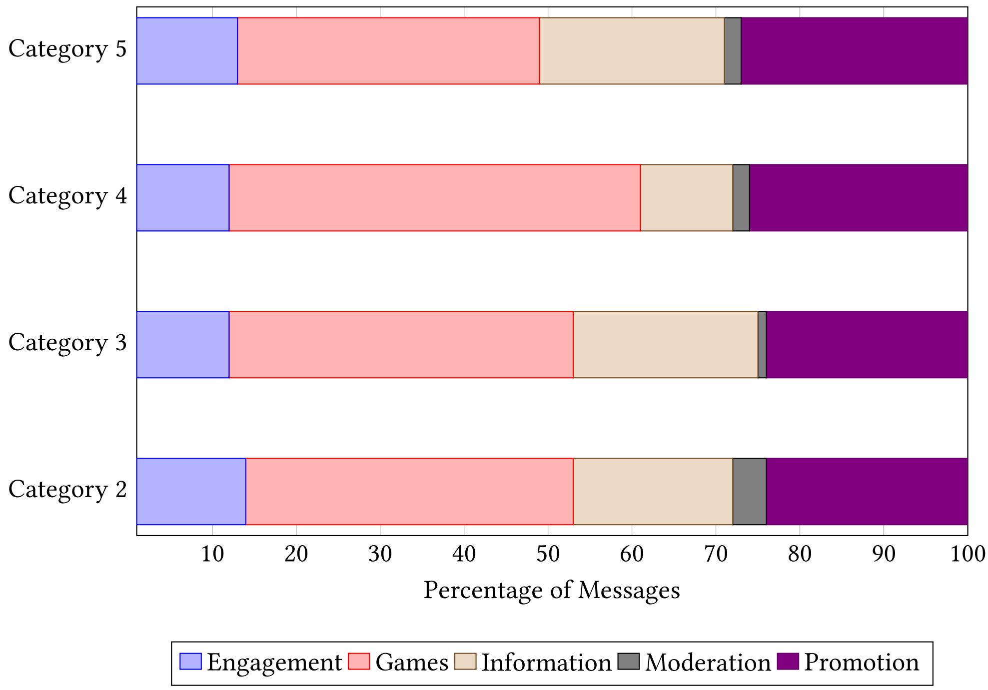

Fig. 10. Prevalence of message type by size group

conversations so that users can track each other's messages through the many other users sending messages at the same time. We do not present this data in table format here because across all categories and sizes of channels, fewer than one in ten thousand users sent an "@" message to a bot each day. In many of our categories, such a message did not happen at all. Intuitively, this makes sense. The concept of "getting a bot's attention" is not a natural way to conceive of a relationship with a bot; users know that, by design, bots are reading every single message.

We next look at how often users "command" bots to perform a particular task. Bot commands are the primary way users solicit information from or interact with bots. Many bot messages are sent in response to a user "command", which is typically a message starting with an "!". Streamers set up these commands to answer frequently asked questions, e.g., "!keyboard" to have a bot share the brand and model of their keyboard. Technically, bot commands work through parsing the text of all messages sent to the channel. If the beginning of a message matches a known bot command, the bot posts the designated response to the command. The control panels for the most-used bots allow users to add commands through a simple interface (see Fig 5).

Bot commands are also core to mini-game play, as users type them to start or join games. Responses to commands can fall within any of the five categories listed above, from listing rules to linking a funny video to sharing the streamer's YouTube channel. Note that the use of "!" is not a technical requirement of Twitch or IRC; bot developers have come to use this syntax because of its convenience and widespread use.

Tables 9 and 10 show total bot commands per user over the two weeks of scraping.

The general decrease in use of bot commands as channel size increases is intuitive; once the information has been posted, it is visible to every user in the channel, so bots in channels with small viewerships can reach ten viewers with a response to a command while bots in channels with large

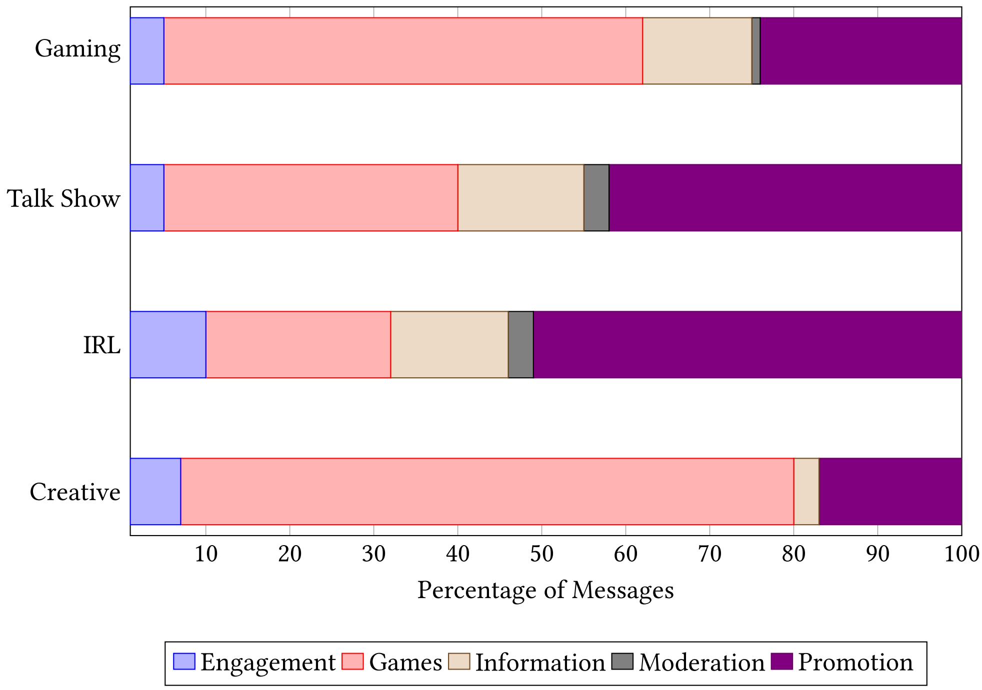

Fig. 11. Prevalence of message type by content category

Table 9. Bot commands per user per hour by size category

	Number of commands	Per user per hour
Category 2	14991	0.003
Category 3	29196	0.002
Category 4	67019	0.003
Category 5	37570	0.002

Table 10. Bot commands per user by content type

	Number of commands	Per user per hour
Creative	3264	0.003
IRL	27782	0.003
Talk Show	1391	0.002
Gaming	47490	0.004

viewerships can reach thousands. It is more difficult to explain the differences in bot command usage across different stream categories; the high proportion of bot commands in creative streams likely follows from their frequent use of bot-led mini-games, but channel sizes may also factor into these results.

6.4 Bots and moderation

Bots on Twitch have two primary methods of engagement - they can post messages in the chat or they can take moderation actions (i.e., timeouts or bans) against offending users. The previous two sections of this paper addressed the former of these, and here we briefly address the latter. Moderation actions on Twitch are taken on approximately 2% of messages on Twitch. They cannot be directly attributed to a single bot unless accompanied by a public message; when users are timed out or banned their messages are removed and may be replaced by "<message deleted>", but from users' perspectives this action could have been performed by any moderator. Note that moderators have access to information about which moderator (or bot) timed out or banned a user, but we have not been made moderators for the streams in our sample.

While we cannot know for certain whether any moderation action was performed by a human or a bot, we can attempt to infer patterns at scale. This section proceeds under the assumption that, in most cases, bots that automatically parse chat messages in real-time and respond with moderation actions when appropriate are faster on average at taking these actions than human moderators who must read each message and click a button on screen to take action. If this is true, we would expect to see a bimodal distribution in duration between message sending and moderation action taken (if any); bots would respond quickly with relatively little variance in time taken, and human moderators would be slower with significant variance depending on how long it takes them to decide what to do, how long it takes them to navigate the interface, and whether they were paying attention when the message was sent.

Figure 12 shows a distribution of the number of seconds taken between message posting and resulting moderation action in messages from across the dataset, using log scale for number of messages. Note that only messages that resulted in a moderation action are included in this figure. This distribution follows the expectations outlined above, providing some support for our assumption. The small spike at just after two seconds may be explained by Twitch rate limits on message-sending, which include ban messages, for non-verified bots. Widely-used bots can receive formal approval from Twitch to exceed these limits[5]; in exploring our data, we found that a substantial number of the bans in this spike came from channels with custom-made or less-widely-used bots, so we exclude these channels from our further analysis because of significantly lower confidence in differentiating automated bans and human bans in this range.

Working from the distribution shown in Figure 12, we define the line for predicting human vs bot bans at 1.6 seconds, the local minimum between the maxima of the two distributions. This is a preliminary estimation, but it is sufficient for brief exploration of ban dynamics. Figures 11 and 12 show human and bot bans by channel categories.

Table 11. Timeouts and bans by size

Timeouts and bans performed by humans and bots by size category			
	Humans	Bots	Ratio
Category 2	581	1159	0.50:1
Category 3	1491	1475	1.01:1
Category 4	2298	5720	0.40:1
Category 5	1489	8915	0.17:1

[5] https://discuss.dev.twitch.tv/t/trying-to-understand-chats-rate-limit/11693

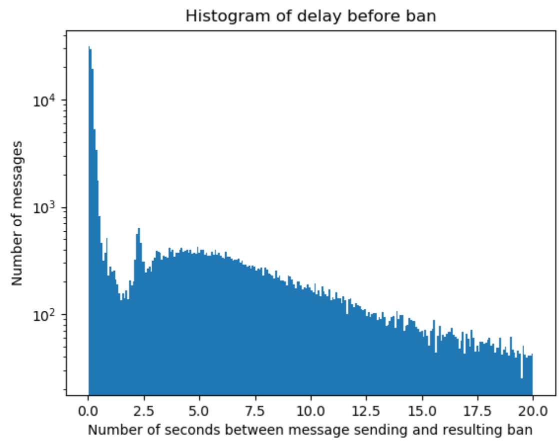

Fig. 12. Distribution of number of seconds between message posting and resulting moderation action

Table 12. Timeouts and bans by category

Timeouts and bans performed by humans and bots by content category			
	Humans	Bots	Ratio
Creative	131	788	0.17:1
IRL	3010	2650	1.14:1
Talk show	56	89	0.63:1
Gaming	788	1326	0.59:1

The ratio of human-made moderation actions to bot-made moderation actions gradually decreases with channel size from categories 3-5, which matches what we would expect - larger channels are harder for humans to moderate quickly because more messages are flowing in, so bots do more of the work. The relatively low ratio observed in category 2 counters this trend and as such is worth further investigation. The ratios observed in content categories also do not have immediately obvious explanations, but might result from different norms in different communities.

7 CONCLUSION

In this paper we have explored the concept of bots' roles in social spaces on Twitch, primarily inspired by a structural concept of roles and previous work exploring humans' roles online. We find that they send messages at a rate substantially higher than any other type of user, including human channel moderators. We identify five types of content that these bots provide - information, moderation messages and warnings, user-engagement, mini-games, and promotion. Though these align in many ways with previous work on bots, e.g., [28, 36, 37, 40], the concept of bot as socially-accepted promoter has not yet been explored in the chatbot literature. Future work could look at what constitutes "acceptable" automated advertising within communities, especially in communities that are patronage-based like many on Twitch [22]. While automated self-promotion is taboo on reddit [28], it clearly is not on Twitch in certain circumstances.

We also find that though users act upon Twitch bots at modest rates via bot commands, they rarely engage in any sort of "conversation" with them in any way comparable to work on conversational bots [42]. The potential roles for a conversational chatbot in a community like Twitch is an open question, with much room for design and development.

Finally, we provide evidence of differences in volumes of human-made and bot-made moderation actions across different types of communities. While these results are preliminary and rely on an assumption about the relative capabilities of bots and humans, they inspire some questions that may provoke further research. What does it mean for a community to delegate a large percentage of its moderation "decisions" to bots? Though these decisions are determined by settings selected or programmed by streamers or moderators, the moderation actions likely take place before human moderators even have a chance to read the affected content, leading to questions about oversight of algorithms that govern behaviors. Though most bots have similar features overall, there are a variety of different specific moderation settings that can be chosen on different third-party bots, and programmers can create their own. Future research could explore what settings streamers in different types of communities choose and why, and also what leads streamers to choose a free third-party bot vs. creating their own.

There remains much work to be done in analysis of bot roles in social spaces and many valuable lenses to direct and interpret this work. In this paper we have used a structural framing, identifying bots' roles through analysis of their action and text patterns, and this functional approach appears to be the way many bots are designed [28]. However, other approaches could be used in design. Symbolic interactionist role theory provides one alternative lens that could inspire new concepts. Per Biddle [4], roles in symbolic interactionist role theory emerge from norms, attitudes, context and contextual demands, negotiation, and the actors' evolving understandings of the situation. These sorts of roles are fluid, socially constructed, context-dependent, and continuously evolving. Though one might argue that the roles bots currently occupy are the result of social construction, bots have never been engaged in the construction of their own roles. Considering a symbolic interactionist approach to roles opens up the potential to design for all sorts of "fuzzy" roles, per Biddle [4], impacted by concepts such as self-presentation, impression and identity management, involvement, and deviance.

Though one might suggest that the solution here is to create increasingly intelligent bots that "learn" norms over time, we suggest that "evolving" is as valid a direction as "learning"; even if we cannot develop a bot capable of analyzing and reconsidering its social role, we can certainly develop bots that push users to think about what social roles bots can have. For example, what would it mean to have a bot that isn't paying attention all the time? That didn't always feel like obeying commands or even telling the truth? That didn't like everybody equally? While Twitch bots can currently be used to amplify memetic content via repetition of in-jokes on command, could a bot actually create new memetic content in ways that surprise and excite its creators?

The work that we present here shows one perspective on what roles bots on Twitch currently have, but questions remain both about what implications these roles have for bots' communities and how Twitch bots' roles might look in the future.

ACKNOWLEDGMENTS

We would like to thank Amazon for supporting this line of work, as well as the FY2018 WV Research Challenge Grant award and the J. Wayne and Kathy Richards Faculty Fellowship in Engineering. We would also like to thank Seth Glickman for feedback on use of technical tools on Twitch.

REFERENCES

[1] Norah Abokhodair, Daisy Yoo, and David W McDonald. 2015. Dissecting a social botnet: Growth, Content and Influence in Twitter. In *Proceedings of the 18th ACM Conference on Computer-Supported Cooperative Work*. ACM, ACM, 839–851.

[2] David Bamman, Brendan O'Connor, and Noah A Smith. 2014. Learning Latent Personas of Film Characters. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. ACL, 352.

[3] Bruce J Biddle. 1979. *Role theory: Expectations, identities, and behaviors.* Academic Press.

[4] Bruce J Biddle. 1986. Recent developments in role theory. *Annual review of sociology* 12, 1 (1986), 67–92.

[5] Christian Elverdam and Espen Aarseth. 2007. Game classification and game design: construction through critical analysis. *Games and Culture* 2, 1 (2007), 3–22.

[6] Oliver Ferschke, Diyi Yang, and Carolyn P Rosé. 2015. A Lightly Supervised Approach to Role Identification in Wikipedia Talk Page Discussions. In *Proceedings of the International AAAI Conference on Weblogs and Social Media. Workshop on Wikipedia, a Social Pedia: Research Challenges and Opportunities*. 43–47.

[7] Danyel Fisher, Marc Smith, and Howard T Welser. 2006. You Are Who You Talk to: Detecting Roles in Usenet Newsgroups. In *Proceedings of the 39th Annual Hawaii International Conference on System Sciences*, Vol. 3. IEEE, 59b–59b.

[8] Claudia Flores-Saviaga, Brian C Keegan, and Saiph Savage. 2018. Mobilizing the Trump Train: Understanding Collective Action in a Political Trolling Community. *Proceedings of the 12th International Conference on Web and Social Media* (2018).

[9] Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics* (2001), 1189–1232.

[10] Keinosuke Fukunaga and Larry Hostetler. 1975. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory* 21, 1 (1975), 32–40.

[11] R Stuart Geiger. 2013. Are Computers Merely Supporting Cooperative Work: Towards an Ethnography of Bot Development. In *Proceedings of the 2013 Conference on Computer-Supported Cooperative Work Companion*. ACM, 51–56.

[12] R Stuart Geiger. 2014. Bots, Bespoke Code and the Materiality of Software Platforms. *Information, Communication & Society* 17, 3 (2014), 342–356.

[13] R Stuart Geiger and David Ribes. 2010. The Work of Sustaining Order in Wikipedia: the Banning of a Vandal. In *Proceedings of the 2010 ACM conference on Computer-Supported Cooperative Work*. ACM, 117–126.

[14] Steven Gianvecchio, Zhenyu Wu, Mengjun Xie, and Haining Wang. 2009. Battle of botcraft: fighting bots in online games with human observational proofs. In *Proceedings of the 16th ACM Conference on Computer and Communications Security*. ACM, 256–268.

[15] Eric Gleave, Howard T Welser, Thomas M Lento, and Marc A Smith. 2009. A conceptual and operational definition of'social role'in online community. In *42nd Hawaii International Conference on System Sciences*. IEEE, 1–11.

[16] Seth Glickman, Nathan McKenzie, Joseph Seering, Rachel Moeller, and Jessica Hammer. 2018. Design Challenges for Livestreamed Audience Participation Games. In *CHI PLAY '18*.

[17] Philippe Golle and Nicolas Ducheneaut. 2005. Preventing Bots from Playing Online Games. *Computers in Entertainment (CIE)* 3, 3 (2005), 3–3.

[18] Neal Gross, Ward S Mason, and Alexander W McEachern. 1958. Explorations in role analysis: Studies of the school superintendency role. (1958).

[19] William A Hamilton, Oliver Garretson, and Andruid Kerne. 2014. Streaming on twitch: fostering participatory communities of play within live mixed media. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1315–1324.

[20] Gary Hsieh, Youyang Hou, Ian Chen, and Khai N Truong. 2013. Welcome!: social and psychological predictors of volunteer socializers in online communities. In *Proceedings of the 2013 Conference on Computer-Supported Cooperative Work*. ACM, 827–838.

[21] Xia Hu and Huan Liu. 2012. Social Status and Role Analysis of Palin's Email Network. In *Proceedings of the 21st International Conference on World Wide Web*. ACM, 531–532.

[22] Mark R Johnson and Jamie Woodcock. 2017. 'It's like the gold rush': the lives and careers of professional video game streamers on Twitch. tv. *Information, Communication & Society* (2017), 1–16.

[23] Robert L Kahn, Donald M Wolfe, Robert P Quinn, J Diedrick Snoek, and Robert A Rosenthal. 1964. Organizational stress: Studies in role conflict and ambiguity. (1964).

[24] Michel Laterman, Martin Arlitt, and Carey Williamson. 2017. A campus-level view of Netflix and Twitch: Characterization and performance implications. In *The 2017 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS)*. IEEE, 1–8.

[25] Kwan Min Lee and Clifford Nass. 2003. Designing Social Presence of Social Actors in Human Computer Interaction. In *Proceedings of the 2003 SIGCHI Conference on Human Factors in Computing Systems*. ACM, 289–296.

[26] Marion J Levy. 1964. *The Structure of Society.* Princeton University Press.

[27] Ralph Linton. 1936. The Study of Man: an Introduction. (1936).

[28] Kiel Long, John Vines, Selina Sutton, Phillip Brooker, Tom Feltwell, Ben Kirman, Julie Barnett, and Shaun Lawson. 2017. Could You Define That in Bot Terms?: Requesting, Creating and Using Bots on Reddit. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 3488–3500.

[29] George Herbert Mead. 1934. *Mind, Self and Society*. University of Chicago Press.

[30] Sue Morris. 2003. WADs, Bots and Mods: Multiplayer FPS Games as Co-creative Media. In *DiGRA Conference*. Citeseer.

[31] Siegfried Frederick Nadel. 1957. *The Theory of Social Structure*. Routledge.

[32] John Paolillo. 1999. The virtual speech community: Social network and language variation on IRC. *Journal of Computer-Mediated Communication* 4, 4 (1999), JCMC446.

[33] Talcott Parsons, Edward A Shils, et al. 1951. Values, Motives, and Systems of Action. *Toward a general theory of action* 33 (1951), 247–275.

[34] Daniel Recktenwald. 2017. Toward a transcription and analysis of live streaming on Twitch. *Journal of Pragmatics* 115 (2017), 68–81.

[35] Marian-Andrei Rizoiu, Timothy Graham, Rui Zhang, Yifei Zhang, Robert Ackland, and Lexing Xie. 2018. # debatenight: The role and influence of socialbots on twitter during the 1st us presidential debate. *Proceedings of the 12th International Conference on Web and Social Media* (2018).

[36] Saiph Savage, Andres Monroy-Hernandez, and Tobias Höllerer. 2016. Botivist: Calling Volunteers to Action Using Online Bots. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*. ACM, 813–822.

[37] Joseph Seering, Robert Kraut, and Laura Dabbish. 2017. Shaping Pro and Anti-social Behavior on Twitch through Moderation and Example-setting. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*. ACM, 111–125.

[38] Joseph Seering, Saiph Savage, Michael Eagle, Joshua Churchin, Rachel Moeller, Jeffrey P Bigham, and Jessica Hammer. 2017. Audience Participation Games: Blurring the Line Between Player and Spectator. In *Proceedings of the 2017 Conference on Designing Interactive Systems*. ACM, 429–440.

[39] Max Sjöblom and Juho Hamari. 2017. Why do people watch others play video games? An empirical study on the motivations of Twitch users. *Computers in Human Behavior* 75 (2017), 985–996.

[40] Ed Summers and Ricardo Punzalan. 2017. Bots, seeds and people: Web archives as infrastructure. In *Proceedings of the 2017 ACM conference on computer supported cooperative work and social computing*. ACM, 821–834.

[41] Alex Hai Wang. 2010. Detecting spam bots in online social networking sites: a machine learning approach. In *IFIP Annual Conference on Data and Applications Security and Privacy*. Springer, 335–342.

[42] Joseph Weizenbaum. 1966. ELIZA-a computer program for the study of natural language communication between man and machine. *Commun. ACM* 9, 1 (1966), 36–45.

[43] Howard T Welser, Dan Cosley, Gueorgi Kossinets, Austin Lin, Fedor Dokshin, Geri Gay, and Marc Smith. 2011. Finding social roles in Wikipedia. In *Proceedings of the 2011 iConference*. ACM, 122–129.

[44] Donghee Yvette Wohn, Guo Freeman, and Caitlin McLaughlin. 2018. Explaining Viewers' Emotional, Instrumental, and Financial Support Provision for Live Streamers. In *Proceedings of the 2018 SIGCHI Conference on Human Factors in Computing Systems*. ACM, 474.

[45] Diyi Yang, Aaron Halfaker, Robert E Kraut, and Eduard H Hovy. 2016. Who Did What: Editor Role Identification in Wikipedia.. In *ICWSM*. 446–455.

[46] Diyi Yang, Miaomiao Wen, and Carolyn Rosé. 2015. Weakly Supervised Role Identification in Teamwork Interactions. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Vol. 1. 1671–1680.

[47] Yuchen Zhao, Guan Wang, Philip S Yu, Shaobo Liu, and Simon Zhang. 2013. Inferring Social Roles and Statuses in Social Networks. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 695–703.

A APPENDIX A: LOG-LOG PLOTS OF MESSAGE ACTIVITY BY USER TYPE

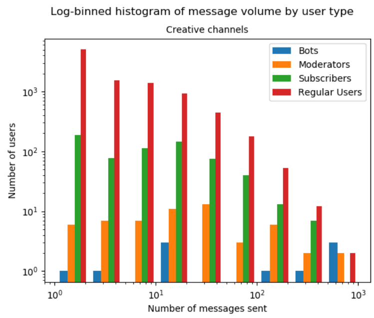

Fig. 13. "Creative" message distribution by user type

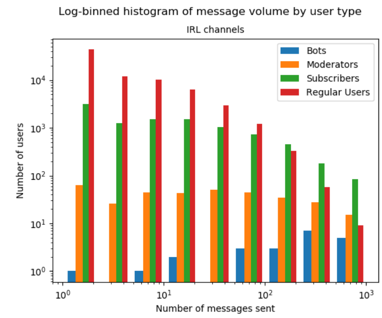

Fig. 14. "IRL" message distribution by user type

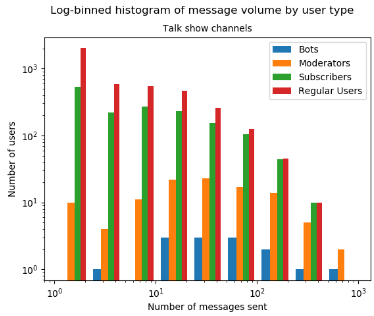

Fig. 15. "Talk show" message distribution by user type

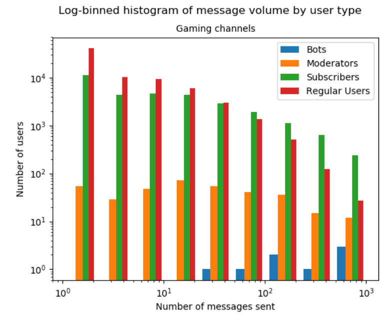

Fig. 16. "Gaming" message distribution by user type

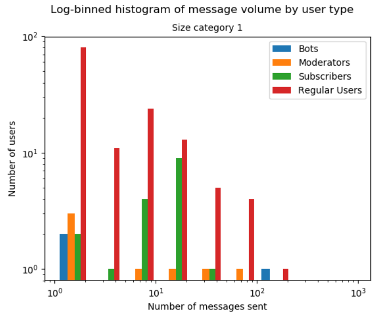

Fig. 17. Size cat 1 message distribution by user type

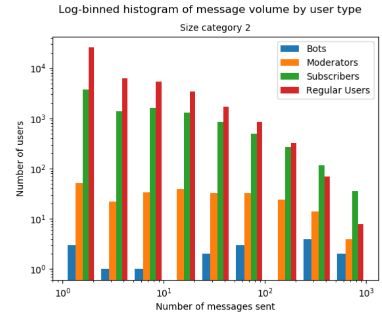

Fig. 18. Size cat 2 message distribution by user type

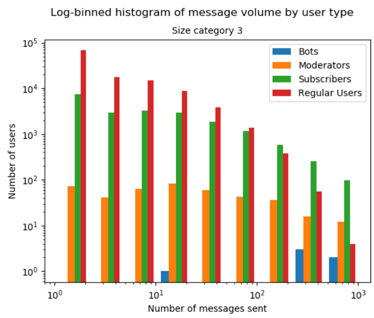

Fig. 19. Size cat 3 message distribution by user type

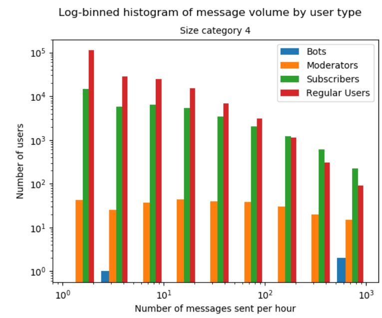

Fig. 20. Size cat 4 message distribution by user type

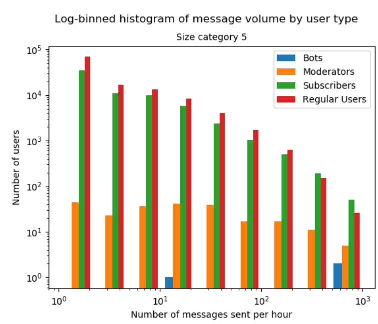

Fig. 21. Size cat 5 message distribution by user type

B APPENDIX B: CHANNELS AND MESSAGE COUNTS

The following table shows the number of messages that were collected from each channel in the sample over the course of data collection. "Total messages" is the sum of "User sent messages" and all bot sent messages, which are divided into columns by which bot sent them. Note that channels 116-125 were not active during data collection, so no messages were collected from them.

Table 13. Counts of messages appearing in channels in the sample by user and by bot type

Channel ID	Total messages	User sent messages	Deepbot	hnlbot	Moobot	Nightbot	Phantombot	Streamelements	Custom
C1	806802	785870	0	0	0	0	0	0	20932
C2	738620	733599	0	5018	0	0	0	0	3
C3	465056	457599	0	7457	0	0	0	0	0
C4	294404	288185	0	0	6219	0	0	0	0
C5	238662	225675	0	0	0	12987	0	0	0
C6	168609	166193	0	17	0	0	0	0	2399
C7	166890	166517	0	0	373	0	0	0	0
C8	159424	154680	0	22	0	0	0	4722	0
C9	148927	146376	0	0	0	0	0	2551	0

C10	148069	145577	0	0	0	2492	0	0	0
C11	145868	143690	0	0	0	2178	0	0	0
C12	143199	140943	0	0	0	2256	0	0	0
C13	137731	134027	0	0	1739	0	0	0	1965
C14	137517	134950	0	0	0	2567	0	0	0
C15	135191	133733	0	0	0	0	0	1458	0
C16	123846	123700	0	0	0	146	0	0	0
C17	122486	120232	0	0	0	2254	0	0	0
C18	119237	119237	0	0	0	0	0	0	0
C19	110599	109646	0	953	0	0	0	0	0
C20	108816	106308	0	0	1122	1386	0	0	0
C21	108310	108298	0	0	0	0	0	0	12
C22	106661	101094	0	0	0	0	5567	0	0
C23	97701	96781	0	221	0	699	0	0	0
C24	87406	85355	0	0	0	2051	0	0	0
C25	86659	84191	0	0	0	2468	0	0	0
C26	83307	82949	0	0	358	0	0	0	0
C27	82421	78697	0	0	0	3724	0	0	0
C28	79181	78320	0	0	0	861	0	0	0
C29	76853	73926	0	2927	0	0	0	0	0
C30	76639	73788	0	0	0	2851	0	0	0
C31	76595	76468	0	32	88	7	0	0	0
C32	73172	70832	0	0	0	2340	0	0	0
C33	69648	68059	0	0	448	1141	0	0	0
C34	69418	69418	0	0	0	0	0	0	0
C35	67258	64216	0	0	0	3042	0	0	0
C36	61852	60443	0	0	0	1409	0	0	0
C37	60941	60941	0	0	0	0	0	0	0
C38	57365	56876	0	0	488	1	0	0	0
C39	56425	55679	0	0	0	746	0	0	0
C40	56279	55273	0	662	0	344	0	0	0
C41	53419	53419	0	0	0	0	0	0	0
C42	45703	45232	0	0	0	471	0	0	0
C43	45160	44826	0	0	0	0	0	334	0
C44	41717	41717	0	0	0	0	0	0	0
C45	40908	38846	0	0	0	2062	0	0	0
C46	39289	38852	0	0	0	437	0	0	0
C47	38519	36377	0	0	0	2142	0	0	0
C48	37931	37931	0	0	0	0	0	0	0
C49	34823	32961	0	0	0	1862	0	0	0
C50	34014	33516	0	0	0	498	0	0	0
C51	32891	32891	0	0	0	0	0	0	0
C52	31693	29676	0	1340	0	0	0	677	0
C53	31191	29521	0	0	0	1670	0	0	0
C54	30929	30107	0	0	0	822	0	0	0
C55	27425	26482	0	0	0	943	0	0	0
C56	22621	22621	0	0	0	0	0	0	0
C57	22469	22469	0	0	0	0	0	0	0

C58	22415	21820	0	382	0	213	0	0	0
C59	22064	21408	0	0	0	0	0	0	656
C60	20471	20327	0	57	14	73	0	0	0
C61	20161	19888	0	0	10	263	0	0	0
C62	20094	18277	0	0	0	1817	0	0	0
C63	18997	18870	0	0	127	0	0	0	0
C64	18305	16599	0	0	0	1706	0	0	0
C65	18182	17636	0	0	0	546	0	0	0
C66	15082	14497	0	0	0	585	0	0	0
C67	12847	12281	0	0	566	0	0	0	0
C68	12758	12758	0	0	0	0	0	0	0
C69	12257	12219	0	0	0	38	0	0	0
C70	11581	11362	0	0	0	219	0	0	0
C71	11216	10903	0	0	313	0	0	0	0
C72	10548	9681	867	0	0	0	0	0	0
C73	9902	9798	0	0	0	104	0	0	0
C74	8870	6934	0	0	0	2	0	1934	0
C75	8820	8758	0	0	62	0	0	0	0
C76	8670	7939	0	0	0	622	0	109	0
C77	8292	7648	0	0	221	423	0	0	0
C78	8087	8087	0	0	0	0	0	0	0
C79	7581	7168	0	413	0	0	0	0	0
C80	5921	5846	0	0	75	0	0	0	0
C81	5751	5514	0	0	0	237	0	0	0
C82	5339	5339	0	0	0	0	0	0	0
C83	5020	4990	0	0	30	0	0	0	0
C84	4653	4562	0	0	91	0	0	0	0
C85	4599	4375	0	0	0	0	0	0	224
C86	4051	3958	0	0	93	0	0	0	0
C87	3915	3879	0	0	0	36	0	0	0
C88	3876	3876	0	0	0	0	0	0	0
C89	3539	3378	0	0	110	51	0	0	0
C90	3208	3148	0	0	0	60	0	0	0
C91	3100	2970	130	0	0	0	0	0	0
C92	2900	2040	860	0	0	0	0	0	0
C93	2242	2125	0	0	0	117	0	0	0
C94	2116	2049	0	0	67	0	0	0	0
C95	1655	1630	0	0	0	4	0	21	0
C96	1565	1555	0	0	0	10	0	0	0
C97	1548	1538	0	0	10	0	0	0	0
C98	1361	1361	0	0	0	0	0	0	0
C99	1313	1133	0	0	0	180	0	0	0
C100	1172	1133	0	0	0	39	0	0	0
C101	841	823	0	0	0	0	0	0	18
C102	838	818	0	0	20	0	0	0	0
C103	357	347	0	0	10	0	0	0	0
C104	282	281	0	0	1	0	0	0	0
C105	254	249	0	0	0	0	0	5	0

C106	118	118	0	0	0	0	0	0	0
C107	25	20	0	1	0	4	0	0	0
C108	23	23	0	0	0	0	0	0	0
C109	10	9	0	0	0	1	0	0	0
C110	9	5	0	0	0	0	0	0	4
C111	6	6	0	0	0	0	0	0	0
C112	5	5	0	0	0	0	0	0	0
C113	2	0	0	0	0	0	0	2	0
C114	2	2	0	0	0	0	0	0	0
C115	1	1	0	0	0	0	0	0	0
C117	0	0	0	0	0	0	0	0	0
C118	0	0	0	0	0	0	0	0	0
C119	0	0	0	0	0	0	0	0	0
C120	0	0	0	0	0	0	0	0	0
C121	0	0	0	0	0	0	0	0	0
C122	0	0	0	0	0	0	0	0	0
C123	0	0	0	0	0	0	0	0	0
C124	0	0	0	0	0	0	0	0	0
C125	0	0	0	0	0	0	0	0	0
Total	7143563	6999749	1857	19502	12655	66207	5567	11813	26213

C APPENDIX C: FEATURES OF BOTS IN SAMPLE)

The following table shows bot features adapted from a spreadsheet made by Twitch power-users[6], and added based on observation when missing. For reference, columns are labeled as follows: **M = Moobot, N = Nightbot, O = Ohbot, D = Deepbot, P = Phantombot, S = Streamelements**. Though features vary slightly across bots, all features fit within similar categories.

Table 14. Features of bots appearing in the sample

Bot feature	Category	M	N	O	D	P	H	S
Chat Notifications								
Follower	Engagement	Yes	No	No	Yes	Yes	No	Yes
Subscriber	Engagement	Yes	No	Yes	Yes	Yes	Yes	Yes
Donation	Engagement	No	No	No	Yes	Yes	Yes	Yes
Host	Engagement	No	No	No	Yes	Yes	No	Yes
Chat Filters								
Character spam filters (e.g. excessive caps)	Moderation	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Link filters	Moderation	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Blacklisting words, phrases, etc.	Moderation	Yes	Yes	Yes	Yes	Yes	Yes	Yes

[6]https://www.reddit.com/r/Twitch/comments/4qcsfq/an_updated_twitch_bot_list/

Whitelisting words, phrases, etc.	Moderation	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Ability to whitelist users from filters (e.g. regulars)	Moderation	Yes	Yes	Yes	Yes	Yes	No	Yes
Stream Interaction								
Polling	Engagement	Yes	Yes	No	Yes	Yes	Yes	Yes
Raffles/Giveaways	Games	Yes	Yes	No	Yes	Yes	Yes	Yes
Viewer queuing	Games	Yes	Yes	No	Yes	Yes	No	No
Whisper functionality	Variable	Yes	No	No	Yes	Yes	No	Yes
Fun chat games/commands (e.g. Roulette, 8ball, etc.) (not betting)	Games	Yes	No	Yes	Yes	Yes	No	Yes
Song requests	Games	Yes	Yes	No	Yes	Yes	No	Yes
Commands								
Ability to create custom commands	Variable	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Commands to set chat filters	Moderation	Yes	Yes	Yes	No	Yes	No	Yes
Timed or repeated messages/commands	Variable	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Quote system	Engagement	No	No	No	Yes	Yes	No	Yes
Commands to change channel settings (e.g. stream title, slow mode, etc.)	Moderation	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Loyalty System								
Currency system	Games	No	No	No	Yes	Yes	No	Yes
Tracks users' total time spent in channel	Information	No	No	No	Yes	Yes	No	Yes
Mini-games for wagering/betting points	Games	No	No	No	Yes	Yes	No	Yes
Ranks	Games	No	No	No	Yes	Yes	No	Yes
Bot Management Features								
External online dashboard	(Technical)	Yes	Yes	No	No	Yes	Yes	Yes
Downloadable build	(Technical)	No	Yes	No	Yes	Yes	Yes	No
Ability to give people full access to the bot (i.e. managers, editors, etc.)	(Technical)	Yes	Yes	Yes	No	Yes	Yes	No
Data and settings are saved to an online server	(Technical)	Yes	Yes	Yes	Yes	No	Yes	Yes
Can be given a custom Twitch username	(Technical)	Yes	No	No	Yes	Yes	No	No

Option to mute bot or limit bot spam	(Technical)	No	Yes	Yes	No	Yes	No	Yes
Manage donations	(Technical)	No	No	No	No	No	No	Yes
Custom stream overlays (visual elements)	(Technical)	No	No	No	No	No	No	Yes
Records chat logs	(Technical)	No	Yes	No	Yes	Yes	No	No
Tracks channel statistics (e.g. viewer count over the last 30 days)	(Technical)	No	Yes	No	No	No	No	Yes

Received April 2018; revised July 2018; accepted September 2018