

It Takes a Village: Integrating an Adaptive Chatbot into an Online Gaming Community

Joseph Seering
Carnegie Mellon University
Pittsburgh, USA
jseering@andrew.cmu.edu

Michal Luria
Carnegie Mellon University
Pittsburgh, USA
mluria@andrew.cmu.edu

Connie Ye
Carnegie Mellon University
Pittsburgh, USA
constany@andrew.cmu.edu

Geoff Kaufman
Carnegie Mellon University
Pittsburgh, USA
gfk@andrew.cmu.edu

Jessica Hammer
Carnegie Mellon University
Pittsburgh, USA
hammerj@andrew.cmu.edu

ABSTRACT

While the majority of research in chatbot design has focused on creating chatbots that engage with users one-on-one, less work has focused on the design of conversational agents for online communities. In this paper we present results from a three week test of a social chatbot in an established online community. During this study, the chatbot “grew up” from “birth” through its teenage years, engaging with community members and “learning” vocabulary from their conversations. We discuss the design of this chatbot, how users’ interactions with it evolved over the course of the study, and how it impacted the community as a whole. We discuss how we addressed challenges in developing a chatbot whose vocabulary could be shaped by users, and conclude with implications for the role of machine learning in social interactions in online communities and potential future directions for design of community-based chatbots.

Author Keywords

chatbot; interaction design; machine learning; AI; BabyBot; Twitch; community interaction; long-term study

CCS Concepts

• **Human-centered computing** → **Human computer interaction (HCI)**; User studies;

INTRODUCTION

In group social interactions, people use a variety of subtle but information-rich cues to decide how to interact. We parse body language; consider our social status within the group; we compare the topics being discussed with our own personal

experiences; and we listen for pauses where it might be appropriate to interject. These signals allow us to engage in a variety of meaningful social processes.

As chatbots and other conversational agents are becoming more prevalent in online social spaces, researchers have re-examined social dynamics in the presence of chatbots. Seering et al. [31], in a recent literature review, found that a strong majority of chatbot research and development has focused on *dyadic* interactions, with conversations occurring between one chatbot and one human. Much less work has focused on *multi-party* interactions where a chatbot participates in a free-flowing group discussion; even in cases where chatbots are deployed within group spaces, they often respond either to pre-set commands or converse primarily with one user at a time.

The development of chatbots for multi-party interactions is a significant, unsolved challenge in language processing, but is also a complex design challenge. How might a chatbot be able to contribute meaningfully to a group discussion? What are the social roles a chatbot can play within a community? Though a very small volume of work (see e.g., [2]) has attempted to address this from a conceptual standpoint, few if any studies have tested chatbots as community members, rather than as service-providers, in a long-term, in-the-wild study. This paper builds on one of the seven conceptual categories for multi-party chatbots presented in Seering et al. [31] – the “Dependent” chatbot, which we reframe as a “Learner” chatbot.

Recent work has been critical of Microsoft’s infamous *Tay* chatbot [28, 31] because of the poor outcomes of its unrestrained use of machine learning. We argue here that the past failures in creating chatbots whose vocabulary is shaped by users are not primarily a result of poor technology, but rather a result of a fundamentally-flawed design premise. We argue that, just as very few parents would want their children to be raised by the Twittersphere, we should not design adaptive chatbots to be trained on such data. Taking the metaphor of a village, we suggest that chatbots may behave more socially appropriately when “raised” by a community with established social structures and prosocial values. While the idea that it is important

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

CHI ’20, April 25–30, 2020, Honolulu, HI, USA

© 2020 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-6708-0/20/04.

DOI: <https://doi.org/10.1145/3313831.3376708>

to find appropriate training data is certainly not new, nor is the idea of inserting humans into the loop, we take these ideas one step further and propose the concept of raising a chatbot from “birth” to “adulthood” in a community. With this work, we aimed to test the concept of raising a chatbot and to identify its potential both for generating a socially-appropriate corpus of training data and for guiding a community to consider its own values in the process.

In this paper we present the results of a three week study of a chatbot deployed in an established online community on Twitch, a popular video livestreaming platform. The bot, named “BabyBot” by the research team and later re-named “PeteBot”¹ by members of the community, was themed as a child learning how to talk and behave. It used a combination of rules-based and Markov chain-based text generation to interact with community members in ways that changed both through acquisition of new vocabulary and through “aging” through pre-designed age states. We present findings from an analysis of interactions between community members and the bot. We describe which of the bot’s features were most successful in generating engaging interactions with users, and discuss the ways in which the community’s relationship with the bot changed over time as they came to accept it as a member of the community. Finally, we discuss implications for the design of future community-based chatbots and more broadly for the role of machine learning and other adaptive approaches in online social interactions.

PRIOR WORK

The design of BabyBot takes inspiration from two lines of work: research on novel dyadic and multi-party interactions with chatbots, and research on the use of technology to supplement livestreams to engage users in more interesting ways.

Engaging interactions with chatbots

Chatbot histories typically begin with ELIZA, a chatbot designed in the 1960s that showed how simple, rule-based interactions can lead to engaging interaction [40]. Subsequent “chatterbots” [23] have focused on attempting to mimic natural human conversation (e.g. [38, 39]). An alternate thread of research has focused on development of *task-oriented* chatbots. For example, chatbots have been developed to help track users’ nutrition habits [10], collect survey data [17], and help people decide where to go to lunch [37]. Bots on Twitch also perform very functional roles in areas like providing information and handling simple moderation tasks [30]. Clark et al. note that, in accordance with these examples, people conceptualize chatbots as tools rather than “a potential companion or social equal”, asserting that “social aspects of conversational interaction are currently absent from people’s perceptions of what conversational agents can and should be capable of performing” [4, p. 475:8].

A relatively small volume of recent work has explored emotionally impactful aspects of social conversation with chatbots. For example, Roussou et al. presented a chatbot designed to evoke emotional engagement around difficult questions [26], while Lee et al. [19] presented “Vincent”, a chatbot that builds

¹The bot had the Twitch usernames `baby_bot_` and later `pete_bot_`.

on principles from Human-Robot Interaction literature [5, 35] to show how caring for a chatbot can evoke an increase in self-compassion. In line with their findings, the results we present here show preliminary evidence that caring for a chatbot can also increase a *whole community’s* level of engagement with each other and strengthen their sense of a shared identity.

As noted above, a strong majority of prior work has focused on *dyadic* interactions with chatbots rather than *multi-party* interactions [31]. We find a handful of exceptions to this – Candelero et al. [2] explored the possibility of a group of chatbots having a virtual coffee around a table with a user, where bots took turns in speaking. In a more goal-oriented interaction, Savage, Monroy-Hernandez, and Höllerer’s “Botivist” called groups of users together to encourage activism [27]. The work we present here takes elements from each of these approaches; casual social interaction based on the first, and more functional, task-oriented group interaction “in the wild” from the latter.

“Audience participation” and community engagement

We chose Twitch as a platform for this work because of its established communities and synchronous conversational structures, but also because of the potential for playful engagement. Much recent research has explored Twitch as a workplace and social space [12, 14, 36]. Twitch communities (‘channels’) are managed by ‘streamers’—users who livestream themselves engaging in various activities, including playing games and/or creating art. One of the main attractions of Twitch and similar livestreaming platforms is the real-time interaction between users and the streamer; users type messages in the chatroom that accompanies the livestream, and the streamer responds on the stream. This back and forth can lead to the development of communities around a particular livestream. Unlike many other online communities, Twitch channels are typically active only when the streamer is livestreaming, which can happen for an amount of time ranging anywhere between a few times a year to upwards of 100 hours per week.² We refer to the period from when a streamer begins a given livestream to when they end it as a streaming “session”.

Recent research focusing on Twitch has explored ways to make viewer-streamer interactions more engaging. For example, early work by Pan, Bartram, and Neustaedter developed a tool to visualize volume of activity in the chatroom during different on-stream events, helping viewers identify exciting moments and helping streamers learn what activities led to the most engagement [25]. Lessel, Vielhauer, and Krüger developed a tool that accompanied a gaming livestream to convey additional information about the game, finding that the immediate feedback provided by the tool had positive impact on user engagement and feelings of agency [20].

An additional line of work on “Audience Participation Games” [9, 32] has tested social interaction through direct integration of game mechanics into Twitch streams. Seering et al. designed and tested a first set of Audience Participation Games, exploring whether the ability to help or to hinder the streamer’s

²Some Twitch communities congregate on other platforms, such as Discord (see e.g., Kiene et al. [15]), when the stream is inactive, but we focus in this work on the interactive periods of engagement on Twitch.

progress would generate more engagement [32]. They found that agency was a core feature in developing audience participation experiences; users in the study enjoyed the push-and-pull aspect of keeping the game exciting by helping the streamer when he fell behind and providing additional challenges when he got ahead.

Broadly, the above work finds value in supplementing the traditional back and forth between streamer and viewers with other modalities of engagement and an ability to shape the course of a narrative, and these themes inform the interactive and narrative-based elements in the design of our chatbot. Per the work on audience participation, we predicted that the streamer in our chatbot’s host community could help keep the bot engaging for the audience, even when it didn’t work as well as expected, through performance and creation of a narrative. We found that this approach was successful and even led audience members to join in a somewhat performative interpretation.

CHATBOT DESIGN

In this section we give a basic overview of the design of BabyBot, and explain the reasoning behind several of its main features. Figure 1 shows the general flow of the chatbot in responding to users.

Feature overview

While the foundational concepts for this design are built on the above literature, specific features emerged through a process of iterative prototyping, testing, and piloting [41]. This design process began in February of 2019 and continued through early June of the same year, and the study ran from mid-June through early July. The lead researcher obtained the streamer’s consent to deploy the bot in his community early in the design process, and also engaged in extended observation of the community in order to get to know its norms and values. Note that, in accordance with recommendations in [31], the design process for this bot did not end with its deployment, but was reconsidered after each streaming session—features were updated or changed in response to how users engaged with them.

We designed BabyBot to “grow up” over the course of a three week study, aging through Baby, Toddler, Adolescent, and Teenager phases. The bot aged a tick every two seconds, and was set to reach a new age phase at pre-defined thresholds of ticks. In the study the Baby phase lasted three streaming sessions; the Toddler phase lasted two; the Adolescent phase lasted two; and the Teenager phase lasted three. For each stage, BabyBot was designed with two types of interactions: *state-actions*, which were interactions the bot initiated itself at random intervals, and *reactions*—responses to users who typed commands to the bot or directly addressed it using its name.

State-action

The bot’s early interaction-seeking behaviors were designed to be structured and to use clear, established commands nested within *pre-defined activities* as a way to familiarize users with the bot. Within each age phase, the bot cycled through randomly selected “states” at semi-random intervals. Each state was associated with a set of explicit commands. For example, during a “hungry” state, users could use the “!feed” command

to try to feed the bot, and the bot responded with a level of satisfaction based on how much it “liked” the given food. Other states included a state where the bot wanted to be entertained and users could “!sing” to it; a state where the bot wanted to be held via “!hold”; and a sleeping state where the bot did not respond to interactions unless it is woken with “!wake”.³ The probability of switching states in any given two-second cycle was initially set at 0.0017, roughly once every twenty minutes, a rate that the researchers felt would be consistent but not disruptive.

We phased out these structured activities as the bot advanced through its age states, instead focusing on more open-ended activities like question-asking and text generation. As this transition occurred, we gradually increased the state change probability to 0.0067, roughly once every five minutes, because these activities were more lightweight. We designed three forms of question-asking: first, in its younger phases, the bot repeated users’ messages back to them, sometimes with words scrambled, and added a question mark to indicate uncertainty. Users found this feature repetitive, and it didn’t provoke follow-up interactions in most cases, so the frequency of this behavior was reduced. In the second form of question-asking, BabyBot asked what a word used by a community member meant, e.g., “what does hair mean”? The final form of question-asking involved pre-scripted open-ended inquiries about general topics, e.g., “What is Twitch like?” and inquiries about community members, e.g., “what is P4 like?”

Reaction: Responding to directed comments

When directly addressed (e.g. “@BabyBot”), the bot generated text according to its age phase. For example, the Baby phase response included a kaomoji face⁴ and/or a “gurgle”, a text chunk indicating a baby sound. The Toddler responded with kaomojis accompanied by single words, brief generated phrases, and/or Twitch emotes. In the Adolescent and Teenager phases, the bot generated increasingly longer sentences.

Technical Overview

Though the focus of this paper is on BabyBot’s design and the reactions it evoked, we briefly review the technical foundations of the bot.

Language, Runtime Environment

The chatbot used in this study was developed using Javascript and ran locally using Node.js. Package management was handled through npm. We used several javascript libraries for Twitch relaying, data saving and language generation; notably, we used tmi.js (v1.4.2) to access Twitch’s Internet Relay Chat, and RiTa.js (v1.3.89; [13]) to perform natural language processing on our text.

Connection to Twitch

The chatbot was able to access Twitch by connecting as a client to Twitch’s IRC network; this process was simplified through use of tmi.js, which provides functions that allow a Twitch

³Though it is not possible to include a full description of all actions the bot can take in each age phase and state, we have made the full code repository available at <https://github.com/ChatbotStudy/Twitch-Chatbot>.

⁴<http://kaomoji.ru/en/>

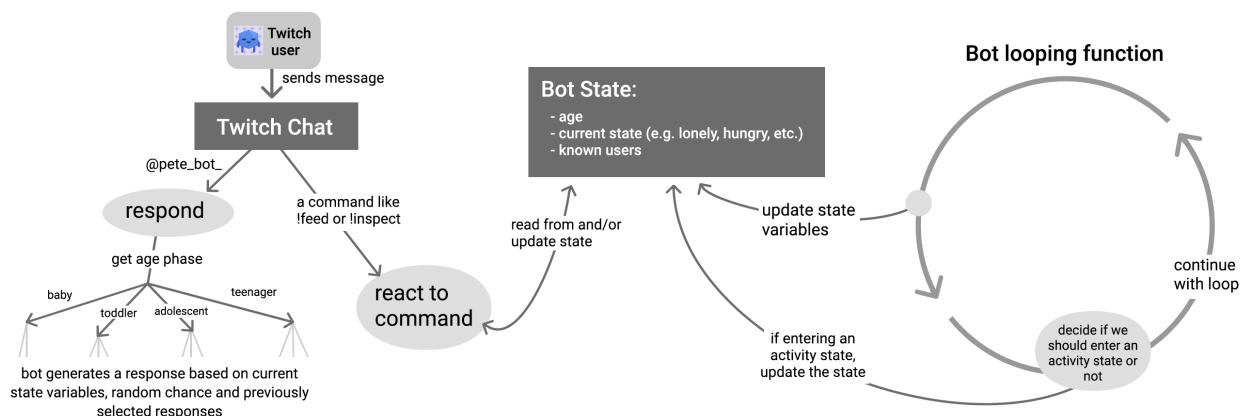


Figure 1. BabyBot’s function flow. The bot responded to mentions of its name by generating text in different ways depending on its age phase. It responds to known “!” commands depended on its age, state, and whether the user is known to the bot.

account to connect to a channel’s chat and send and receive messages. If an incoming message was a recognized command (!command) or directed at the chatbot (“@BabyBot”), it responded appropriately. If the command was part of an ongoing activity state, the program updated the state of the activity and provided feedback in the chat. If the message was directed at @PeteBot, the bot responded in a predefined structure based on what age phase it was in. Messages sent by someone on a voluntary opt-out list were ignored.

Looping

To simulate aging, our chatbot program ran in a loop and incremented an age variable every loop iteration, with each loop lasting 2 seconds. Within the loop, the program first used the age to determine what phase the bot was in, updated the various state elements of the bot, and then branched off into that particular age’s behavior function. If the bot was in an idle state during a loop iteration, it used a skewed random function to decide whether to start an activity, which would run for a preset number of loops.

Natural Language Processing

The natural language processing was handled with RiTa.js; this library was used primarily to give the chatbot the appearance of “growth” through language, though it did not give the bot any formal “understanding” of the language it used. Messages from the chat were continuously fed into RiTa.js’s Markov chain implementation, and the resulting Markov chain was used to generate the chatbot’s speech, with different parameters used in different age states to generate shorter or longer sentences. Thus, the chatbot’s corpus for language generation was entirely composed of the language from the chat stream.

Design considerations

While many considerations were at play in the design of this chatbot, we expand on two of them below.

Limited text generation

Though many of BabyBot’s interactions were in part rules-based, probabilistically-driven text generation played a significant role in social interactions with the bot beginning in its

Toddler phase and expanding as it aged. We chose to build Babybot’s text generation corpus exclusively from messages sent in the target community during the course of the study. We took this approach in part because it matched the theme of a child learning from its environment, but more importantly, because this approach allowed us to test a chatbot that “learned” words from what it observed. We hoped to show that a chatbot can avoid the pitfalls of Tay and other similar bots if carefully designed and deployed in an thoughtfully-chosen setting.

Though this paper is primarily intended to contribute design considerations, and we do not claim to have made any innovations in language processing, it is useful to briefly discuss the primary technical challenge we faced: as the bot would need to begin generating text in its Toddler phase roughly nine hours into the study, we estimated that it would have a corpus of at most one or two thousand sentences to draw from, and it would not have access to pre-trained language models per the above line of reasoning. Advanced research in text generation has typically used much larger datasets for training. Sequence-to-sequence neural network models, a widely-established approach in machine translation and text generation, are frequently trained on datasets of millions or tens of millions of sentences; Li et al., for example, described a “relatively small database” of 400,000 sentences as “not suitable for open domain dialogue training” [21, p. 6]. Some text generation work uses smaller corpora, e.g., Guo et al., who recently presented a new approach for text generation building on prior work in Generative Adversarial Nets (GANs), but these training sets still used 200,000 and 80,000 sentences respectively for long and medium sentence generation, and these sentences came from sources that used formal language and grammar [11]. Of the remaining realistic options, we elected to generate text using a Markov chain-based approach from RiTa.js as a simple and straightforward approach that fit clearly within the constraints we set and would still return acceptable results.

There are reasonable questions to be asked about to what extent design research should use state-of-the-art versions of the technology in question. Our chatbot does not use state-of-the-art

algorithms for text generation, but state-of-the-art algorithms have mostly focused on contexts where a huge volume of training data is available. We suggest that our design-driven approach was the most appropriate for this context as it provided an experience to users that was sufficient to garner insights on the core research questions of this work. It also allowed us to be flexible and adapt the design of the robot throughout the study to reveal additional insights, needs, and dynamics in the community. Design research has a long history of testing concepts and uncovering valuable insights through minimal technical development, using methods such as paper prototyping, wizard-of-oz testing, and speed dating [8, 22, 34]. We believe that our technical approach had a reasonable balance between flexibility, development time, and technical outcomes for the goals of our study.

Moderation strategies

Extensive prior work has documented the perils of using biased or problematic training data [1, 24], and problematic behaviors certainly occur on Twitch [33]. Per the above, the main approach we took to “moderation”, i.e., ensuring that BabyBot would not begin to spout Nazi propaganda, was the choice to place the bot within an established community with an interest in maintaining a positive environment. This was largely successful; no users posted any particularly sexist, racist, or homophobic messages in the chat during the three week study period. Users’ “bad behavior” was mostly limited to joking about trying to get the bot to express a preference for vodka and beer. There were also a handful of noteworthy comments implying abuse of the bot, e.g., “!punch” and “!spank”, but this falls in line with prior work finding that users express early frustration with conversational agents’ limited capacities through simulated abuse [6].

We took two additional approaches to prevent the bot from acting in harmful ways. First, the bot was given a list of “banned” words that, when it observed them in a message, it would not add the message to its corpus for future text generation. We actively reviewed this list throughout the study. The second approach we took was to have a researcher monitoring the bot at all times it was active, with the option to shut it down at any moment if it showed problematic tendencies. Although this did not ever occur in the study, we feel that it was an important safeguard to retain. The pace of messages appearing in the chat was also slow enough that the researchers could manually remove messages from the corpus or re-add messages that had been caught by the filter. This is clearly a capability that relies on having a community of an appropriate size; this process would not have functioned well in an open, Twitter-sized environment. However, we see this as the intended outcome of this work rather than a limitation. One of the points we aim to make is that a given chatbot should not be designed with the goal of being successful in any randomly-selected community. Chatbots should instead be designed with heavy attention to the specific social context in which they will be deployed.

COMMUNITY SELECTION AND STUDY METHODS

Previous linguistically-adaptive chatbots (e.g., Tay) have been deployed in large-scale network settings. In this work, we built from the principle that, just as a child would be raised

in a family, a chatbot that “learns” how to behave would also benefit from a caring community. After considering several platforms, including Facebook, Reddit, Twitch, and Discord, we chose Twitch because of its community-based structure, synchronous communication, and the performative elements of streaming.

The community selected for this study was an established community based around an affiliate⁵ streamer who had been streaming on Twitch since early 2015. At the time of the study, he streamed three to four nights a week for approximately three to four hours per night. Accordingly, the bot was active in the channel for ten streaming sessions and more than thirty hours over the course of three weeks. The streamer had roughly 1500 followers at the beginning of the study and typically streamed to 10-30 concurrent viewers. Note that, while this level of viewership may seem small compared to the most visible Twitch streams, Twitch’s distribution of channel sizes has a long tail of very small channels [30], which put his channel at approximately the 90th percentile in terms of concurrent viewership.

This community was ideal for testing a chatbot for several reasons. First, its size led to a relatively steady flow of conversation, but not so much that the bot’s activity could get buried. Second, most community members knew each other at least virtually, leading to a strength of identity and connection that we felt would help create a positive environment for the chatbot to “grow up” in. Finally, while we elected not to survey users about their demographics because of a cultural standard on Twitch of personal privacy, a larger proportion of users in this study mentioned being of racial minority groups and/or non-heterosexual sexual orientations than is typical of Twitch.

This chatbot was present in the host channel during the majority of each streaming session for a period of three weeks. Users were introduced to the bot both by a block of text posted in the chat when it arrived, and through a link that the streamer had overlaid on his stream. The link directed community members to a website hosted by the research team that described the bot, and this link was also posted in the chat once every thirty minutes. Users were given an opportunity to opt-out of the study via a form on this webpage. This study’s protocols were approved by the IRB at Carnegie Mellon University.

Over the course of the study, forty-six unique users posted messages in the channel, sending 5716 messages in total.⁶ Of these, eighteen unique users interacted directly with the bot via recognized commands (e.g., “!hold”) or by using its name to direct a message at it (i.e., “@BabyBot”), totaling 550 messages. The bot itself posted 1154 messages during the study. Approximately 52% of the messages posted by the bot were “self-initiated”, meaning that they were sent unprompted by the bot, typically as an attempt to start an interaction. The other 48% were prompted by users via recognized commands or by directing a message at the bot as described above.⁷

⁵<https://affiliate.twitch.tv/>

⁶We exclude the supervising researcher from this count as well as the channel’s existing bot, which was used primarily to post periodic informational messages.

⁷The code for performing the above analyses is also included in the github repository previously linked.

Data collection for this study included transcripts of all in-chat communication and annotations of video recordings of the stream and chat. Two researchers separately observed and made notes of interactions that included or referenced BabyBot over the course of three weeks of the study. Observations and transcripts were then affinity diagrammed by both researchers to identify themes in the interaction. Disagreements between the researchers were resolved through discussion.

FINDINGS

The section below describes the main themes that emerged from our qualitative analysis.

Coming to terms with a baby chatbot

Users began the study with a set of preconceived notions about what chatbots are like. Some of this came from a general perception of AI as a whole; one user joked about how they were unwittingly training an AI to “take over the world”. The users’ first impressions were also likely shaped by prior experience with chatbots on Twitch. The streamer who hosts this channel had used Nightbot⁸ prior to this study for tasks like promoting his social media, running simple mini-games, and performing simple moderation functions, and he continued to use Nightbot alongside BabyBot during the study. Despite their similar technical foundations, BabyBot was perceived as a conversational agent while Nightbot was perceived as a tool. This likely resulted at least in part from BabyBot’s more nuanced “personality” and greater interactivity.

Getting to know the bot

Over the course of this study, particularly in the first few days of interaction, users engaged in a variety of “probing” behaviors in attempts to make sense of BabyBot and its functionalities. Prior work has shown that users often push social boundaries in their first interactions as a way of learning about the agent’s personality and functionalities [7, 29]. Other work found that users explore how “human” conversational agents are through questions such as “are you in love?” [18]. In this study, users maintained a running commentary about their perceptions of BabyBot, sometimes asking it questions ranging from “What are you like?” to the more absurd “Do you like tentacles?”. Users described BabyBot as “dumb” at first, then “getting smarter”. At various points they attributed personality to it, e.g., calling it “bloodthirsty” and “a virtual vampire” when it generated text seeded from users’ previous conversations about a blood-related game mechanic.

Users also attempted to test BabyBot’s functionalities by exploring the space of possible commands. In the first few days, participants quickly learned that the baby stage of the bot generates responses with a baby-like utterance and a “kaomoji” face when they included its name in a message. Per their interpretation, BabyBot “knew” when they were talking to it or about it. This exploration process was repeated in a similar fashion each time the bot advanced to a new age phase.

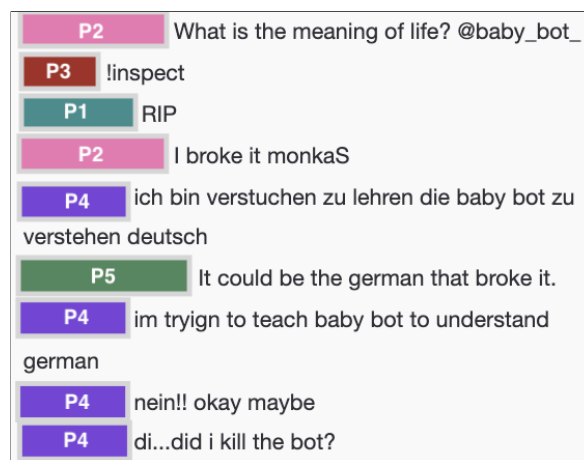
Humorous aggression

Another form of exploratory interaction was the use of aggressive and abusive language toward the bot. Previous work

⁸<https://nightbot.tv/>

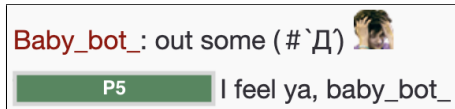
showed that over 10% of interactions with conversational agents includes some form of aggressive language [6]. Chin and Yi [3] identified three primary types of verbal abuse: insults, swearing and threats. In our study, we observed a range of abusive language towards BabyBot from each of the three categories: Users insulted the bot (“you little shit” (P2)), they swore at it with or without a reason, and they threatened it (“[P6] pours scalding hot water on baby_bot_” (P6)). In one case when BabyBot was in a sleeping state, and users tried a number of possible commands to wake it up including “!shake” and “!hit”, though the bot was programmed to wake up only in response to “!wake”. In another case, a user who had become frustrated with the bot tried “!punch” and “!spank” commands.

These aggressive interactions happened mostly when BabyBot did not respond or did not respond in the way the user had hoped, though in a few cases it was a reaction to BabyBot saying something that was interpreted as insulting or dismissive of the user. Nevertheless, these responses seemed mostly lighthearted—users were playful and found amusement in their own abusive behaviors: they discussed how their aggressive behavior was probably going to “corrupt” or “break” BabyBot, and laughed when BabyBot generated humorously aggressive language and behavior itself. These users’ aggressive behaviors also faded over time, which contributes to the interpretation that this was part of an exploratory stage, similar to such stages observed in interactions with voice agents [29]. The observed aggressive behaviors contrasted with users’ playful “parental worrying” about the bot’s state, e.g., in situations where BabyBot did not respond to queries that had previously produced results. In these cases the bot was usually offline for a few seconds for hotfixes, but users interpreted this jokingly as the “death” of the bot.

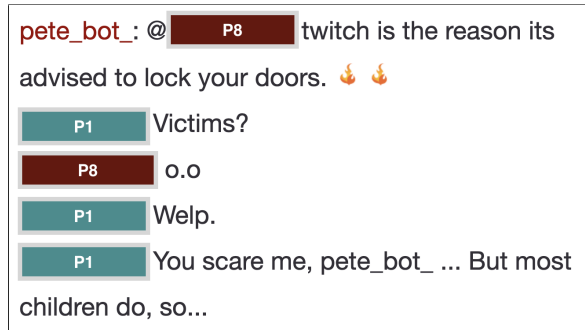


Sensemaking and the value of mild ambiguity

Due to the nature of BabyBot’s text generation process, many of its utterances did not make obvious sense. However, rather than hindering its interactions with users, the ambiguity of BabyBot’s words provided starting points for conversation and humor. Many of BabyBot’s most engaging moments were when the text it generated was almost but not completely coherent. Users enjoyed the process of trying to interpret meaning from its strange sentences or utterances:



Perhaps the most engaging comments that BabyBot produced were those that accidentally touched on something humorously profound. In one case, the bot's generated text was interpreted as commentary on the less-than-stellar reputation of parts of Twitch as a haven for internet trolls.



BabyBot was perceived as particularly intelligent when it said something that related to the game being played on stream. In one case, it generated text that approximated a question about the mechanics in the game, and in another case it generated text that seemed to show understanding of the themes of the game. The bot generated text that appeared to “sass” the streamer for his performance in the game twice, and both the streamer and other users found these comments particularly amusing.

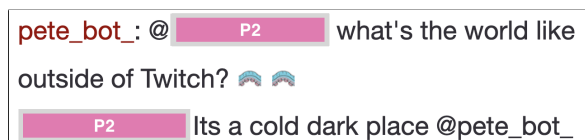
Not all of the phrases generated by the bot made enough sense to encourage users to interpret or respond. While we were initially concerned that these nonsense phrases would detract from the overall experience, users typically ignored BabyBot in those instances without becoming annoyed or frustrated.

Facilitating new interactions

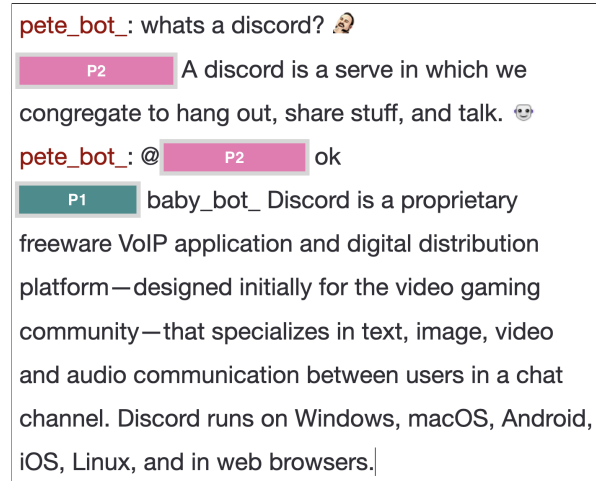
Sparking human-bot interactions

In the bot's baby phase, several of the structured activities that were part of the design (e.g., “!sing”, “!feed”) provoked entertaining interactions. Users enjoyed trying to figure out the bot's “food preferences” were; the underlying code counted the number of syllables in what the bot was “fed” and rated words with more syllables more favorably. Though users never figured out this pattern, they continued to feed the bot enthusiastically. In one case, a user was jokingly “offended” by the fact that the bot did not like the pie they offered.

In the older age phases, humorous text generation and asking pre-set questions led to the most engagement. The latter attracted more attention in the Toddler and Adolescent phases, while the former attracted more attention in the Teenage phase. Responses to pre-set questions ranged from humorous and sarcastic to honest and thorough:



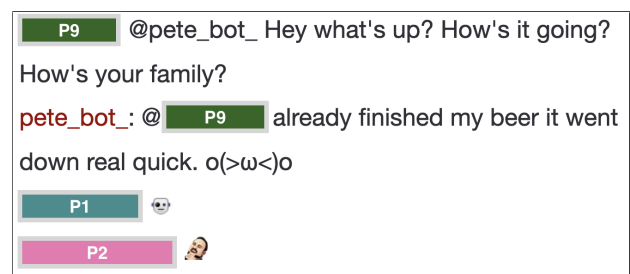
The detailed answer to the bot's question, “*whats a discord?*” offers an interesting design possibility for onboarding new members into a community. While on most of Twitch this interaction would involve a new user typing “!discord” and receiving an automated response, often including a link to the channel's Discord server, the interaction shown here has the opportunity to offer additional and more personalized information. New users might benefit from observing a bot asking a range of “newbie” questions in cases when they are not sure what questions to ask or are nervous about being made fun of.



Sparking human-human interactions

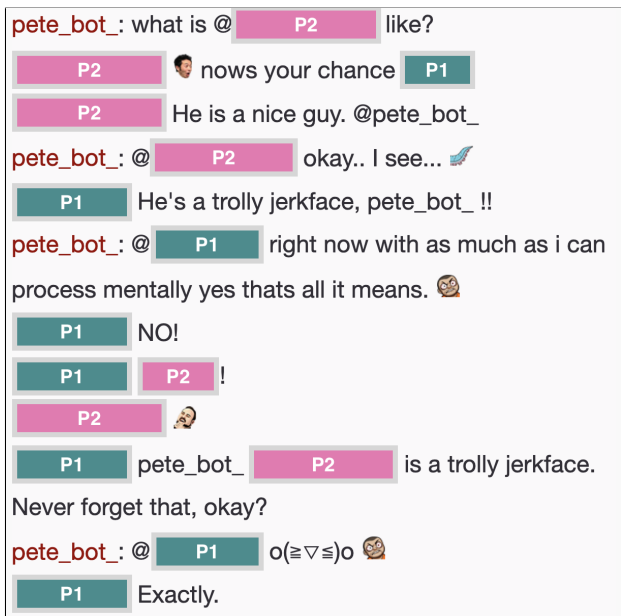
Though the bot was generally successful in directly soliciting interactions, it also generated interactions between community members that might not have happened otherwise. For instance, the interactions between users and the bot during its Baby phase were simple, but they frequently continued beyond the interaction with the bot itself—after a user was “singing” to the bot, the streamer picked up on one of the lines and began singing the song on-stream. In another instance, when the users were choosing a name for the bot, they discussed broader issues of gender and gendered names.

Absurd generated text often provoked other forms of community interaction. Users were particularly engaged with each other when joking about the bot's “corruption”. One user became notorious for trying to teach the bot all the wrong things, from “*loving vodka*” to violent behaviors. The community, including the streamer, frequently mentioned this user's influence on the bot and jokingly blamed them for “*ruining*” it:



Somewhat more meaningful inter-user engagement was seeded when BabyBot used pre-set questions asking about a random

recently active user. Community members seemed to have enjoyed using the bot’s questions to compliment each other, reflect on their relationships with one another, and crack jokes:

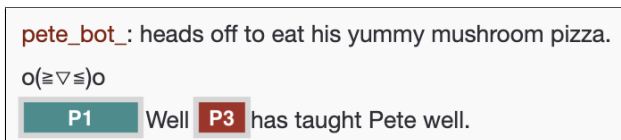


Becoming parents to a baby(bot)

Raising a bot

The presence of the bot, especially in its early phases, sparked many playful conversations about parenting. Users competed to try to “seed” it with different ideas; a repeated theme was the bot’s food preferences, which users attempted to influence both via the “!feed” command and via directly addressing the bot. In one case users tried to get the bot to take a side in a debate over what toppings belong on pizza.

More broadly, users tried various strategies to gain the bot’s “favor”, and expressed excitement when the bot said anything that indicated that it liked them.

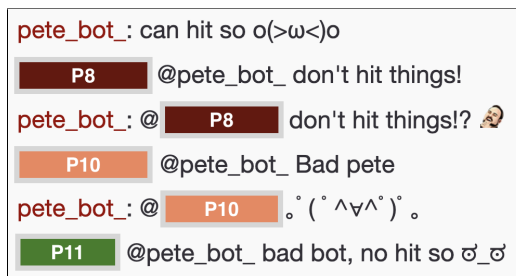


Users had many discussions throughout the study about whether or not they were raising the bot well. As noted above, users frequently expressed humorous concern that they were raising the bot poorly, particularly by lying to it, teaching it age-inappropriate things, and feeding it unhealthy foods (e.g., beer and vodka). In reaction to the wave of “bad parenting”, a number of users actually attempted to teach the bot appropriate behaviors and manners, engaging with it in a playfully strict manner. Some users saw this as “un-teaching” the bad behaviors that the bot had previously been taught. As the bot reached phases where it began to ask questions, users seemed to share the assumption that they should answer its questions honestly. When a user lied to the bot in response to a question, that user was often chastised by other members of the community.

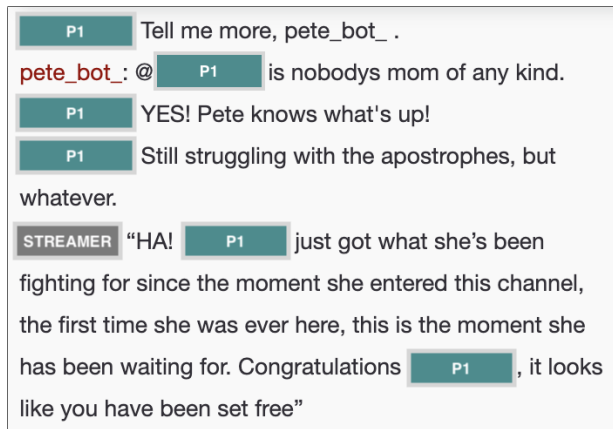
As the bot reached its Teenager phase, users’ interpretations of its generated text shifted and new themes appeared in their interactions. Users reacted to its comments several times as if they had been made by an “angsty” or annoying teenager, and several users engaged the bot in a running joke about how it needed to “get a job” and start “pulling its weight”.

Building relationships with the bot

As users built relationships with the bot, they expressed increasing playful indignation when they felt ignored or rejected by the bot. In its early phases, this often happened when the bot did not respond favorably to the songs they sung to it or the foods they fed it. Later on, as BabyBot began generating phrases, users responded when they recognized what is usually perceived as misbehavior:



These small interactions seemed to create deeper relationships between users and BabyBot over time. One user, P1, who frequently interacted with BabyBot, was often teased by other community members who called her the community’s “mother”. P1 was particularly validated when the bot “agreed” with her frustrations:

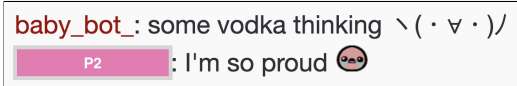


Though individual users developed relationships with the bot over time, the community as a whole also began to welcome it as a “member” of the community. When new users joined the channel, they were quickly introduced to the bot in a way that described it more as an agent than a piece of technology.

Personal ownership

During the bot’s deployment, users quickly developed varying senses of ownership over it. In one Baby phase activity, the bot asked to be held with “!hold”, but this activity had not been designed with safeguards to prevent users from swiping the bot from each other’s “arms”. As a result, in nearly all of the “!hold” states, users competed to be the last to hold the bot.

As the bot advanced through its age phases, users changed how they performed ownership behaviors. The most common trigger for feelings of ownership in later age phases was when the bot generated text that could clearly be traced back to a specific user’s previous chat message. In some cases, the user who originally contributed the text responded with pride. In one case, a user who had been repeatedly “!feed”-ing the bot vodka was gratified when it later mentioned vodka in an emotive expression.



baby_bot_: some vodka thinking \ (· ∨ ·)/
P2: I'm so proud 🥰

Users frequently engaged in meta-commentary during other conversations about how the bot would later re-purpose the messages they were sending. This is in some sense an *advantage* of overfitting; seeing how their engagement directly impacted the bot was rewarding to users and strengthened feelings of parental attachment.

Community ownership

Early in the bot’s Adolescent phase, when it began generating full sentences of text, the community started a conversation about whether the bot should have its own name or whether it should remain “BabyBot”. After discussing several possibilities and running a quick poll, the community decided to name the bot PeteBot (Twitch username: “pete_bot_”). The community members present during the streaming session transitioned seamlessly to calling the bot PeteBot and quickly informed other members of the new name when they arrived.

Shortly after the bot was given its new name, a regular community member gifted the bot a subscription to the channel, a status which costs \$5 and allows access to custom emotes designed for the channel. The user explained that they wanted the bot to be able to use the community’s emotes, but the gift may have also signaled a kind of acceptance; nearly all of the regular community members had long-standing subscriptions, mostly as a show of loyalty and support for the streamer, so gifting the bot a subscription sent a message that the bot belonged in the community. Other core community members also expressed a strongly positive reaction to the gift and welcomed PeteBot to the group of subscribed users.

Over the course of the three weeks of the study, users developed habits for interacting with the bot. For example, users in this community have a long-standing habit of saying hello to people as they arrive in the chat each day, and over time they came to include the bot in this habit. This occurred mainly after the bot was formally named PeteBot, further supporting the idea that giving the bot a new name and a subscription signaled a form of social acceptance.

DISCUSSION AND IMPLICATIONS

In this section we reflect on several major themes that emerged from our analysis and how they highlight ways in which this design was variably successful in achieving the bot’s goals.

Bots can grow up too

Interactions with BabyBot began with some struggle and a lot of exploration. Users didn’t know what to expect from a baby

chatbot. Though the structured interactions in the Baby stage were fairly easy to pick up and led to humorous and engaging interactions, the lack of an open-ended way to play with the bot caused frustration for users when they attempted to interact with it in a range of ways that did not work (for example, trying “!play”, a command that we had not implemented).

Nevertheless, we believe that the Baby phase was important as part of the interaction with users and its eventual acceptance—during that stage, participants had time to get used to the presence of the bot and become comfortable interacting with it. We believe this stage had a role in building expectations for the next bot phases.

As the bot entered its Toddler and Adolescent phases, users were more likely to initiate interactions and converse with it. Users were also less likely to engage in aggressive interactions, perhaps signaling that the community had become comfortable with the bot, its presence, and its capabilities.

It’s sometimes okay not to make sense

The nature of probabilistically-driven language generation is such that sentences will be generated that do not always make sense, and this was especially the case here as the training corpus was very small and because we imposed strict thematic constraints on what algorithms we could use. However, the generation of nonsensical phrases did not significantly reduce the quality of users’ experience; when the bot interacted with participants in a way that mostly made sense, users participated in conversation. When phrases generated by the bot did not make sense, they ignored it.

One of our original goals had been to structure a set of rules that would allow the bot to generate conversation that was as close to human conversation as possible. Yet over the course of the study, we discovered that when BabyBot interacted in a way that fully fit the community and seemed like just another user, it wasn’t as engaging as when it generated coherent but ambiguous phrases; users consistently enjoyed interpreting meaning from them. This collective creation of a narrative and attaching a personality to the bot seemed to be a somewhat meaningful activity for the community.

Bots can talk to everyone

In some of the “baby-like” questions that BabyBot asked, it also highlighted its potential to clarify information that might not be known to all users. In some cases, there may be shame associated with asking questions in such a community, as it could make the user appear like they don’t belong. A chatbot does not have such shame. In the case of BabyBot, the community had no expectation for it to know, for example, what a Discord server is, and they were happy to answer its questions.

BabyBot was successful in sparking interaction both by having one-on-one conversations and by bringing up new topics that began long discussions among community members. BabyBot also occasionally encouraged people to reflect on relationships by asking one user about another, and caused users to reflect on the community as a whole by using language connected to specific users and asking community-specific questions.

Bots can support community self-moderation

The strategies we used to moderate the language used by BabyBot were tailored to this particular space of small communities, but larger communities would likely surface other design considerations. For example, a limited word list was adequate where users were naturally inclined to behave fairly well, but larger spaces might include users who were more determined to circumvent, or even subvert, rules. Similarly, while having a human constantly monitoring the bot's behaviors was a reasonable safeguard in our study environment, this could quickly become an untenable approach if many instances of the bot were running simultaneously in different spaces.

One potential strategy that emerged in our study comes from our observation of users' natural tendencies to rebuke the bot for behaving "badly". Prior work (e.g., [16]) has shown the importance of giving clear examples of which behaviors are acceptable and which are not, so a formalized ability for established community members to rebuke the bot could perform two moderation functions simultaneously: (1) Adjusting the bot's behavior when it has said something problematic and (2) using these rebukes to signal to newcomers how to behave. This latter function could also encourage community members to engage in more community self-moderation, a practice where community members naturally rebuke each other for breaking rules and/or teach rules and norms to newcomers, which has been found to be a valuable complement to formal moderation strategies [33]. A possible supplement to this approach would be, when the bot was "rebuked", for it to ask why a particular behavior wasn't okay. The process of explaining norms to a bot could clarify norms for community members and could also help moderators and senior users reflect on these norms.

CONCLUSION

In evaluating the impact of this chatbot, we conclude by reflecting on the three challenges proposed in Seering et al. [31, pp. 450: 9-10] for designing chatbots as community members:

1. *Does the chatbot become recognized as a legitimate participant within the community?*

Over the course of the three weeks BabyBot was present, community members came to treat it as an agent with a personality and to engage with it on a much more socially nuanced level than they did with Nightbot, the previous bot in the channel, though community members did not quite reach a point where they treated it like they treated each other. Though we feel that three weeks was an appropriate amount of time for a thorough first evaluation of this design, it is possible that users could have seen BabyBot more as a novelty than as a legitimate participant, possibly causing their interest in BabyBot to fade over time. Social legitimacy is a complicated concept to define, and more work exploring how users conceive of "legitimacy" in this type of social agent would be valuable. At minimum, BabyBot expanded community members' understanding of how a chatbot could fit in their community.

2. *Does the chatbot contribute meaningfully to the development of the community?*

This challenge offers the most potential for future development beyond what we have currently designed. The community in

which BabyBot was deployed had a very long history and an established set of core members, so the community as a whole did not undergo significant social change during this three week study. BabyBot's strength was in facilitating a number of engaging, enjoyable group conversations and activities, which in turn made the stream more engaging overall; the streamer even requested that he be allowed to continue running BabyBot in his channel after the study ended, as he found it useful in keeping the stream enjoyable and in filling up downtime.

We posit two future spaces where the bot might contribute more meaningfully to a community's development. First, the bot could be deployed in a community that was new or more actively growing, helping community members get to know each other and form a shared identity. Second, testing BabyBot in an environment that had slightly more conflict and misbehavior could push the limits of our core design philosophy—does the bot need a completely stable "home life" in order to grow up well, or can its adoption into a more uncertain environment help a community reflect?

3. *Does the chatbot's role in the community evolve over time?*

Of the three challenges posed in Seering et al. [31], BabyBot was designed most directly to meet the challenge of evolution over time. Beyond simply changing its features as it moved from one phase to the next, the bot's social role shifted over time. It began as a "Dependent", with community members treating it as something that needed care and attention, but finished closer to a "Peer". Using text generation based on a corpus that expanded as the bot aged, in combination with evolving predefined age states, was very successful in fostering relationships that grew in social sophistication over time.

Broadly, this work shows the potential for linguistically-adaptive chatbots as members of online communities, showing examples of multi-party interactions that provoke meaningful engagement between users and with the bot. We show one case where a chatbot was successfully "raised" by a village, and make the broader argument that, when carefully designed with heavy attention to the social context of a target community, the use of technological approaches that allow users to influence bots' vocabulary is not inherently problematic. Future work in this direction can build on these findings to explore in more depth the potential for chatbots to help communities develop over time in meaningful ways.

Acknowledgements

We would like to thank Greg MacDonough and April Sperry for support during the writing process, and Adam Seering for feedback on algorithms. We would also like to thank Juan Rios, Nicole Wan, Iris Hwang, and Donghee Yvette Wohn for their help during the ideation process. Finally, we would like to thank the streamer and his community for hosting us and for participating in this study.

REFERENCES

- [1] Natã M Barbosa and Monchu Chen. 2019. Rehumanized Crowdsourcing: A Labeling Framework Addressing Bias and Ethics in Machine Learning. In *Proceedings of*

- the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 543.
- [2] Heloisa Candello, Claudio Pinhanez, Mauro Carlos Pichiliani, Melina Alberio Guerra, and Maira Gatti de Bayser. 2018. Having an Animated Coffee with a Group of Chatbots from the 19th Century. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems (CHI EA '18)*. ACM, New York, NY, USA, Article D206, 4 pages. DOI: <http://dx.doi.org/10.1145/3170427.3186519>
- [3] Hyojin Chin and Mun Yong Yi. 2019. Should an Agent Be Ignoring It?: A Study of Verbal Abuse Types and Conversational Agents' Response Styles. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, LBW2422.
- [4] Leigh Clark, Nadia Pantidi, Orla Cooney, Philip Doyle, Diego Garaialde, Justin Edwards, Brendan Spillane, Emer Gilmartin, Christine Murad, Cosmin Munteanu, Vincent Wade, and Benjamin R. Cowan. 2019. What Makes a Good Conversation?: Challenges in Designing Truly Conversational Agents. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, New York, NY, USA, Article 475, 12 pages. DOI: <http://dx.doi.org/10.1145/3290605.3300705>
- [5] Kerstin Dautenhahn. 2007. Socially intelligent robots: Dimensions of human–robot interaction. *Philosophical Transactions of the Royal Society of London B: Biological Sciences* 362 (2007), 679–704. DOI: <http://dx.doi.org/10.1098/rstb.2006.2004>
- [6] Antonella De Angeli and Sheryl Brahnham. 2008. I hate you! Disinhibition with virtual partners. *Interacting with computers* 20, 3 (2008), 302–310.
- [7] Stefania Druga, Randi Williams, Cynthia Breazeal, and Mitchel Resnick. 2017. "Hey Google is It OK if I Eat You?": Initial Explorations in Child-Agent Interaction. In *Proceedings of the 2017 Conference on Interaction Design and Children (IDC '17)*. ACM, New York, NY, USA, 595–600. DOI: <http://dx.doi.org/10.1145/3078072.3084330>
- [8] Samantha Finkelstein, Evelyn Yarzebinski, Callie Vaughn, Amy Ogan, and Justine Cassell. 2013. The effects of culturally congruent educational technologies on student achievement. In *International Conference on Artificial Intelligence in Education*. Springer, New York, NY, USA, 493–502.
- [9] Seth Glickman, Nathan McKenzie, Joseph Seering, Rachel Moeller, and Jessica Hammer. 2018. Design Challenges for Livestreamed Audience Participation Games. In *Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play (CHI PLAY '18)*. ACM, New York, NY, USA, 187–199. DOI: <http://dx.doi.org/10.1145/3242671.3242708>
- [10] Bettina Graf, Maike Krüger, Felix Müller, Alexander Ruhland, and Andrea Zech. 2015. Nombot: Simplify Food Tracking. In *Proceedings of the 14th International Conference on Mobile and Ubiquitous Multimedia (MUM '15)*. ACM, New York, NY, USA, 360–363. DOI: <http://dx.doi.org/10.1145/2836041.2841208>
- [11] J Guo, S Lu, H Cai, W Zhang, Y Yu, and J Wang. 2018. Long text generation via adversarial training with leaked information. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*. AAAI, Palo Alto, CA, USA, 5141–5148.
- [12] 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 32Nd Annual ACM Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 1315–1324. DOI: <http://dx.doi.org/10.1145/2556288.2557048>
- [13] Daniel C. Howe. 2009. RiTa: Creativity Support for Computational Literature. In *Proceedings of the Seventh ACM Conference on Creativity and Cognition (C&C '09)*. Association for Computing Machinery, New York, NY, USA, 205–210. DOI: <http://dx.doi.org/10.1145/1640233.1640265>
- [14] Mark R Johnson and Jamie Woodcock. 2019. 'It's like the gold rush': the lives and careers of professional video game streamers on Twitch.tv. *Information, Communication & Society* 22, 3 (2019), 336–351. DOI: <http://dx.doi.org/10.1080/1369118X.2017.1386229>
- [15] Charles Kiene, Jialun Aaron Jiang, and Benjamin Mako Hill. 2019. Technological Frames and User Innovation: Exploring Technological Change in Community Moderation Teams. *Proc. ACM Hum.-Comput. Interact.* 3, CSCW, Article 44 (Nov. 2019), 23 pages. DOI: <http://dx.doi.org/10.1145/3359146>
- [16] Sara Kiesler, Robert Kraut, Paul Resnick, and Aniket Kittur. 2012. Regulating behavior in online communities. In *Building Successful Online Communities: Evidence-Based Social Design*, Robert Kraut and Paul Resnick (Eds.). MIT Press, Cambridge, MA, USA, Chapter 4, 125–177.
- [17] Soomin Kim, Joonhwan Lee, and Gahgene Gweon. 2019. Comparing Data from Chatbot and Web Surveys: Effects of Platform and Conversational Style on Survey Response Quality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, New York, NY, USA, Article 86, 12 pages. DOI: <http://dx.doi.org/10.1145/3290605.3300316>
- [18] S. Kopp, L. Gesellensetter, and Wachsmuth I. Krämer, N.C. 2005. A Conversational Agent as Museum Guide – Design and Evaluation of a Real-World Application. In *Intelligent Virtual Agents. IVA 2005. Lecture Notes in Computer Science, vol 3661*, T. Panayiotopoulos, J. Gratch, R. Aylett, D. Ballin, and T. Olivier, P. Rist (Eds.). Springer, Berlin, Germany.

- [19] Minha Lee, Sander Ackermans, Nena van As, Hanwen Chang, Enzo Lucas, and Wijnand IJsselstein. 2019. Caring for Vincent: A Chatbot for Self-Compassion. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, New York, NY, USA, Article 702, 13 pages. DOI: <http://dx.doi.org/10.1145/3290605.3300932>
- [20] Pascal Lessel, Alexander Vielhauer, and Antonio Krüger. 2017. Expanding Video Game Live-Streams with Enhanced Communication Channels: A Case Study. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 1571–1576. DOI: <http://dx.doi.org/10.1145/3025453.3025708>
- [21] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A Diversity-Promoting Objective Function for Neural Conversation Models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California, 110–119. DOI: <http://dx.doi.org/10.18653/v1/N16-1014>
- [22] Michal Luria, Samantha Reig, Xiang Zhi Tan, Aaron Steinfeld, Jodi Forlizzi, and John Zimmerman. 2019. Re-Embodiment and Co-Embodiment: Exploration of social presence for robots and conversational agents. In *Proceedings of the 2019 on Designing Interactive Systems Conference*. ACM, New York, NY, USA, 633–644.
- [23] Michael L Mauldin. 1994. Chatterbots, tinymuds, and the turing test: Entering the loebner prize competition. In *AAAI-94 Proceedings*, Vol. 94. AAAI, Palo Alto, CA, USA, 16–21.
- [24] Safiya Umoja Noble. 2018. *Algorithms of Oppression*. NYU Press, New York, NY, USA. DOI: <http://dx.doi.org/10.2307/j.ctt1pwt9w5>
- [25] Rui Pan, Lyn Bartram, and Carman Neustaedter. 2016. TwitchViz: A Visualization Tool for Twitch Chatrooms. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '16)*. ACM, New York, NY, USA, 1959–1965. DOI: <http://dx.doi.org/10.1145/2851581.2892427>
- [26] Maria Roussou, Sara Perry, Akrivi Katifori, Stavros Vassos, Angeliki Tzouganatou, and Sierra McKinney. 2019. Transformation Through Provocation?. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, New York, NY, USA, Article 627, 13 pages. DOI: <http://dx.doi.org/10.1145/3290605.3300857>
- [27] 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 (CSCW '16)*. ACM, New York, NY, USA, 813–822. DOI: <http://dx.doi.org/10.1145/2818048.2819985>
- [28] Ari Schlesinger, Kenton P. O'Hara, and Alex S. Taylor. 2018. Let's Talk About Race: Identity, Chatbots, and AI. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 315, 14 pages. DOI: <http://dx.doi.org/10.1145/3173574.3173889>
- [29] Alex Sciuto, Arnita Saini, Jodi Forlizzi, and Jason I Hong. 2018. Hey Alexa, What's Up?: A mixed-methods studies of in-home conversational agent usage. In *Proceedings of the 2018 Designing Interactive Systems Conference*. ACM, New York, NY, USA, 857–868.
- [30] Joseph Seering, Juan Pablo Flores, Saiph Savage, and Jessica Hammer. 2018. The Social Roles of Bots: Evaluating Impact of Bots on Discussions in Online Communities. *Proc. ACM Hum.-Comput. Interact.* 2, CSCW, Article 157 (Nov. 2018), 29 pages. DOI: <http://dx.doi.org/10.1145/3274426>
- [31] Joseph Seering, Michal Luria, Geoff Kaufman, and Jessica Hammer. 2019. Beyond Dyadic Interactions: Considering Chatbots As Community Members. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, New York, NY, USA, Article 450, 13 pages. DOI: <http://dx.doi.org/10.1145/3290605.3300680>
- [32] 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 (DIS '17)*. ACM, New York, NY, USA, 429–440. DOI: <http://dx.doi.org/10.1145/3064663.3064732>
- [33] Joseph Seering, Tony Wang, Jina Yoon, and Geoff Kaufman. 2019. Moderator engagement and community development in the age of algorithms. *New Media & Society* 21, 7 (2019), 1417–1443. DOI: <http://dx.doi.org/10.1177/1461444818821316>
- [34] Carolyn Snyder. 2003. *Paper prototyping: The fast and easy way to design and refine user interfaces*. Morgan Kaufmann, San Francisco, CA, USA.
- [35] Fumihide Tanaka and Shizuko Matsuzoe. 2012. Children Teach a Care-receiving Robot to Promote Their Learning: Field Experiments in a Classroom for Vocabulary Learning. *J. Hum.-Robot Interact.* 1, 1 (July 2012), 78–95. DOI: <http://dx.doi.org/10.5898/JHRI.1.1.Tanaka>
- [36] T.L. Taylor. 2018. *Watch Me Play: Twitch and the Rise of Game Live Streaming*. Princeton University Press, Princeton, NJ, USA.
- [37] Nada Terzimehić, Mohamed Khamis, Florian Bemmam, and Heinrich Hussmann. 2018. Lunchocracy: Improving Eating Dynamics in the Workplace Using a Bot-Based Anonymous Voting System. In *Extended Abstracts of the*

2018 CHI Conference on Human Factors in Computing Systems (CHI EA '18). ACM, New York, NY, USA, Article LBW022, 6 pages. DOI :
<http://dx.doi.org/10.1145/3170427.3188626>

- [38] Richard Wallace. 2003. *The elements of AIML style*. Alice AI Foundation.
- [39] Richard S Wallace. 2009. The anatomy of ALICE. In *Parsing the Turing Test*. Springer, New York, NY, USA, 181–210.
- [40] Joseph Weizenbaum. 1966. ELIZA – a Computer Program for the Study of Natural Language

Communication Between Man and Machine. *Commun. ACM* 9, 1 (Jan. 1966), 36–45. DOI :
<http://dx.doi.org/10.1145/365153.365168>

- [41] John Zimmerman, Jodi Forlizzi, and Shelley Evenson. 2007. Research Through Design As a Method for Interaction Design Research in HCI. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*. ACM, New York, NY, USA, 493–502. DOI :
<http://dx.doi.org/10.1145/1240624.1240704>