GAMEPLAY LIVESTREAMING: HUMAN AGENTS OF GAMESPACE AND THEIR PARASOCIAL RELATIONSHIPS

By

Alex P. Leith

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Media and Information Studies—Doctor of Philosophy

2019

ABSTRACT

GAMEPLAY LIVESTREAMING: HUMAN AGENTS OF GAMESPACE
AND THEIR PARASOCIAL RELATIONSHIPS

By

Alex P. Leith

Gameplay livestreaming is an increasingly popular form of media, with tens of thousands of people choosing to do it as either a hobby or career. Once each of these individuals creates a Twitch account and starts broadcasting themselves, they become a media figure. This dissertation examined the chats from thousands of partnered Twitch channels. The two key areas of examination are parasocial relationships and gameplay engagement. Parasocial relationshxips state that media users can begin to develop perceived relationships with media figures as they consume content containing that figure. A series of Python bots gathered chat and stream data over a month from 30 Twitch categories (e.g., *Hearthstone*, *League of Legends*, Art, and Just Chatting). The bots logged a total of 321,189,309 messages from 6,564,307 senders and 117,943 channels. After cleaning the data for partnership status, stream language, and message count, coding divided the remaining 3,224,942 messages from 1,298,148 senders and 3,127 channels into their appropriate groups (i.e., messages target and stream content). The research hypotheses subdivided the dataset several times. All hypotheses had the messages separated between streamer-specific messages and other-specific messages. Streamer-specific messages are messages which include the at symbol (@) and the channel name, thus signaling message intentionality to the streamer. Hypotheses two further divided the messages between gameplay and non-gameplay streams, and hypothesis three divided the messages from gameplay streams into entertainment and expertise streams. The hypotheses persistently found that the message target was a reliable predictor of verbal immediacy, the metric used to identify parasocial relationships. Stream content either proved to be a counter-intuitive predictor or no predictor of verbal immediacy. Grounded theory methods addressed the research questions and produced two common distinctions for gameplay involvement. Viewers can engage with gameplay by asking questions but can also elevate themselves to human agents of gamespace through providing information or suggestions.

This dissertation is dedicated to my grandfather, SGM Douglas R. Leith, who showed me the importance of lifelong learning and my mother, Janet V. Leith, who showed me the power of hard work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

This dissertation explores the world of gameplay livestreaming. Gameplay livestreaming is the practice of broadcast gameplay and commentary as it is occurring, as opposed to the more established Let's Plays, which involve content creators editing pre-recorded gameplay and commentary before uploading it onto sites like YouTube. Twitch, the dominant gameplay livestreaming platform in the Western Hemisphere, provides all of its users the capability to livestream both gameplay and non-gameplay content. Researchers often divide gameplay content into two categories: expertise and entertainment (Leith, 2015, 2018; Taylor, 2018; Uszkoreit, 2017). Expertise streams include tournaments, professionals, and non-professionals. Professionals are individuals who make money from the game they are streaming outside of streaming. Non-professionals are highly informed or skilled players that are not otherwise attached to the game. Entertainment streams include variety, single-game, and celebrity. Variety streamers will regularly stream multiple games from either the same genre or multiple genres. Single-game streamers will almost exclusively stream the same game or, possibly, games from the same franchise, such as all of *Dark Soul* or *Super Smash Bros.* games. Non-gameplay content includes any stream in which gameplay is not the primary focus, such as music and art streams.

The first stage of analysis examines the affective messaging of Twitch chat users through the lenses of parasocial relationships (PSRs). PSRs occur when media users develop specific feelings toward media figures as they consume content with the figure (Klimmt, Hartman, & Schramm, 2006). Under PSRs, users convert these positive dispositions and content consumption into a level of perceived closeness that mimics traditional face-to-face relationships, like colleagues and neighbors (Perse & Rubin, 1989). As media users can similarly view these media figures as to how they would view a friend, they are also likely to reach out to them. While communication would fall outside the scope of parasocial interactions (Horton & Wohl, 1956), it is not uncommon for individuals to contact those with whom they feel close. These attempted communications will be the focus of the research hypotheses. Hypothesis one examined the variance in verbal immediacy (see Bazarova, Taft, Choi, & Cosley, 2013) between streamer-targeted and viewer-targeted messages. A streamer-targeted message is a message in which the viewer includes the exact channel name in their messages, thus signaling to the streamer that the viewer is attempting to communicate directly with the streamer. A viewer-specific message is one in which the viewer includes the specific name of another Twitch user than the streamer. Hypothesis two continued separation streamer-targeted messages and viewer-targeted messages but also separated chat according to whether or not the channel was streaming gameplay. Hypothesis three examines expertise and entertainment streams separately.

Gameplay livestreams provide a unique look at gameplay and gamespace. Researchers can approach gamespace in multiple ways: intrinsic and extrinsic. Intrinsic gamespace limits gamespace to the developed game elements presented on the screen. This including non-player characters (NPC), tutorials, and item storage (Wainess, Kerr, & Koenig, 2011). Extrinsic gamespace expands gamespace to include anything that directly affects gameplay, including controllers and people (Newman, 2002a). The inclusion of people into gamespace is the critical area of interest for this dissertation.

Humans can have multiple roles within gamespace. Traditional gamespace, such as gaming at an arcade or in a living room, provides two roles for human agents: players or onlookers. A player is an individual actively controlling the game while the onlooker is the individual watching the gameplay. The onlooker can become an element of gamespace by providing information or directions to the player (Newman, 2002a, 2002b). Streaming gamespace consists of similar roles with slight variations: streamers and viewers. Streamers are the same as players but have taken on additional duties, such as controlling their broadcast and interacting with their chat. Viewers differ from onlookers as they watch the player from a different location and communicate with the streamer through chat. The added complexity to these roles provide barriers to their entry to gamespace. Despite potential barriers, viewers, as they begin to develop a PSR with the streamer, may be more likely to communicate with the streamer as they would with a friend. The two research questions are posed to examine this new relationship. Research question one asks if viewers are attempting to engage with the gameplay through chat and research question two asks if any of these attempts at engagement are sufficient for including viewers into gamespace.

Twitch chat provides all the incites necessary for addressing the research hypotheses and questions. Python bots collected data from Twitch by connecting to identifying popular Twitch channels, connecting to the chats for these channels, and logging the chat messages. The bots first identify the most popular channels from the chosen categories and then logs, among other things, the name of the channel and game. Once the channels are selected, additional bots connect to these channels to log the name of the channel, the name of the message sender, the message, and the time.

A Raspberry Pi cluster was used to both run the bots and store the PostgreSQL database. To assist with data management, each of the game and nongame categories, which were identified using third-party sites, had distinct tables on the PostgreSQL database. After data collection was complete, manual coding identified channel characteristics. Coding included identifying partnered streamers with English-speaking chats and whether the streamers are expertise or entertainment streamers. LIWC (Linguistic Inquiry and Word Count) addressed the research hypotheses, and n-grams addressed the research questions.

LIWC counts specific words from predetermined dictionaries. The datasets were separated and analyzed as necessary for each research hypothesis and question to identify communication variances. Mixed-design ANOVA testing then verified these variances. N-grams and manual coding analyzed gameplay engagement. N-grams condensed all streamer-targeted gameplay messages into the most common 5-grams (quintgrams). Manual coding identified examples of gameplay and gamespace engagement from these quintgrams and their source.

The research hypotheses and questions provided significant insights. Hypothesis one, which posited that streamer-targeted messages would have a greater verbal immediacy than other-targeted messages, was supported. Hypothesis two, which divided all the chats into gameplay and non-gameplay streams while maintaining the distinction between streamer-targeted and other-targeted messages, was partially supported. The verbal immediacy was still greater for streamer-targeted messages than for other-targeted messages; however, the content was inverse to verbal immediacy than posited by the hypothesis. Hypothesis three, which removes all non-gameplay stream messages and divided gameplay streams between entertainment and expertise streams, found a continued distinction between streamer-targeted and other-targeted messages while it found no statistically significant variance between messages of different content types. The analyses to address the research questions found that viewers are actively attempting to engage with gameplay through Twitch chat. Viewers can alter the role of streamers within gamespace by posing questions to the streamers through chat. If the viewers are more interested in personally entering gamespace, they can also do this through chat by providing new information or directions instead of asking questions. Each of these findings produces a list of potential limitations and directions for future research.

GAMEPLAY LIVESTREAMING

Capitalizing upon the growing interest in watching others play video games, sites like YouTube and Twitch have developed large communities of gamers turned content creators. Most spectators will turn to YouTube for pre-recorded gameplay, called Let's Plays. Twitch is a platform that predominantly broadcasts gameplay livestreams, though other content, such as talk shows and music, are rising in popularity. Twitch had over 2 million unique streamers and 360 million chat messages scanned by Automod each month in 2017 (Freitas, 2018). The same report accounts for more than 27,000 partnered and 150,000 affiliate streamers. A partnered streamer is one in which a channel has provided a sufficient following and viewership to justify specific perks. Most streamers earn partnership by averaging at least 75 viewers for an extended period (Twitch, 2017), though it is possible to circumvent if the streamer represents a sufficiently large off-site following (e.g., musicians, athletes, or YouTubers). Affiliate status serves as a path for partnership, providing some of the perks of partnership, such as the ability to subscribe to the streamer or receive some channel-specific emotes, with a smaller viewership requirement.

Twitch provides for a unique example of media convergence, which is the "intersection of distinct media and information technology systems that have previously been thought of as separate and self-contained," such as Internet Protocol TV (Dwyer, 2010, p. 2). In the case of Twitch, the convergence occurs between videogames and text-based chat. This specialized social media frame converges by allowing users to follow, be followed, and interact with other specific users. Each Twitch user is given their own channel upon account creation from which they can stream (e.g., twitch.tv/APLeithTV). When visiting a channel, a user sees a channel-specific video player, chat, and information panels. If the channel is live, the broadcast will immediately begin. Channel visitors will also have access to past broadcasts (i.e., Videos on Demand or VODs) through a tab on the channel page. The chat is a Twitch channel-specific Internet Relay Chat (IRC) channel. While on the channel's homepage, the chat is synchronous. If watching a VOD, the chat will be synched to the content, so that viewers can see what the live viewers were discussing (Twitch, 2017). VOD viewers can also add additional messages to the live chat for future viewers. The informational panels are used on channels by streamers to present information the channel owner identifies as important, such as links to other social media platforms or answers to frequently asked questions (FAQs).

Among the various motivations for watching streams on Twitch, Hamilton, Garretson, and Kerne (2014) found that unique content and community participation generally drew viewers into watching streams. They argued that the primary activity of viewers who engage in chat (chatters) is sociability through "humorous banter" and "light-hearted conversation" (p. 1315). Unlike other media, Twitch streams are regularly 8-10 hours long and 5-7 days a week. Viewers can spend dozens of hours a week watching their favorite media figures and interacting with them and other members of those figures' community. Viewers can drastically reduce the perceived distance between themselves and the streamers during these viewing sessions and increase the sense of intimacy (Horton & Wohl, 1956; Korzenny, 1978).

Gameplay Livestreaming Typology

Streamers are regularly seeking methods to distinguish themselves from other streamers (Hamilton, Garretson, & Kerne, 2014). Despite this ambition, past scholars have identified recurring themes among streams and streamers. A commonly identified characteristic of streamer identity is whether they are an expert or if they are more of an entertainer (Leith, 2015; Taylor, 2018; Uszkoreit, 2017). Leith (2018) later built a fuller typology around these more standard streams (see Table 1). Streams of unique novelty, such as roleplay and telethon streams, were classified as nonstandard streams. A standard method for distinguishing between an expertise and an entertainment streamer is through their variation in video game choice. Experts will predominantly or exclusively play a single game or genre of games; entertainers will generally choose from a variety of games. Though game variations are a vital distinction between expertise and entertainment streams, it is less universally relevant to the nonstandard streams. Considering the intention of this dissertation is to identify broader trends across the entirety of the Twitch platform, the more novel nonstandard stream types will not receive special attention.

Table 1

Gameplay Livestream Typology

Type	**Description**
Standard	
Expertise	
Tournament	Streamer is producing an event around the gameplay.
POV	*Streamer is producing a single perspective of the tournament.*
Professional	Streamer is playing a game they earn money from outside of streaming.
Non-professional	Streamer is playing a game they do not earn money from outside of streaming.
Entertainment	
Variety	Streamer is playing one of many games from which they choose.
Single-game	Streamer is playing one game but is not classified as an expert.
Celebrity	Streamer is streaming in addition to their primary source of fame (e.g., musician, athlete, actor).
Nonstandard	
Educational	Streamer is either coaching or teaching while engaged in gameplay.
Mobile	Streamer is playing a game while in motion.
Playerless	Streamer is not the individual playing the game.
Podcast	Streamer is primarily focusing on discourse and not gameplay.
Roleplay	Streamer is portraying a character either in-game or in-stream.
Telethons	Streamer is primarily attempting to raise money, often over an extended period.
Tabletop	Streamer is playing a non-digital game.
Virtual	Streamer is playing a virtual reality game.
Other	
Non-gameplay	Streamer is doing something other than gaming, such as art or music.

Expertise

A majority of the highest-profile Twitch channels are considered expertise streams. For example, Riot Games hosts major *League of Legends* (*LoL*) tournaments on their Twitch channel, and Ninja, a former *Halo* pro who is now the largest independent streamer on the platform, is a *Fortnite* streamer. Expertise-based streamers are not limited to the most popular streamers, especially since not all expertise streamers are from esports. Leith (2018) found three general categories for expertise type streams: tournament, professional, and non-professional.

Tournament. Major tournaments draw a large audience to the Twitch platform. Whether the parent company of the game hosted the tournament (e.g., Riot Games and Blizzard) or third-party organizations (e.g., DreamHack and ESL), major tournaments regularly average hundreds of thousands of concurrent views. Tournament streams, regardless of their size, follow a similar format to traditional sports broadcasts. Namely, a team of casters provides commentary throughout the event, including in-game and post-game analysis. The broadcast generally rotates between points of view (POVs) of players as they become involved in the action. Some games, such as *Hearthstone* or *The Binding of Isaac: Afterbirth*, can easily combine multiple feeds into a single broadcast, making POV switching unnecessary. Unlike other standard streams, organizers segregate players from broadcast components, including the chat.

Broadcast segregation creates a stream environment for viewers that closely resembles traditional media. Therefore, the chats for tournament streams are distinct from other types of streams. For example, viewers cannot communicate with players through the chat, which means that tournament chats should exclude player-directed messages. Major tournaments also regularly stream to hundreds of thousands of viewers, creating a massive chat experience where thousands of new messages are continually being posted, replacing previous messages. A superficial examination of chat during these tournaments closely mirrors the stands of a sporting event with basic, broad communication (e.g., cheers and boos).

POV. Tournament broadcasts may include additional streams that prioritize different POVs. POV streams broadcast a singular perspective. For example, a Counter-Strike: Global Offensive (CS: GO) stream may choose to either stream a single player's POV or the POVs of a single team. The stream may broadcast in-game sounds or include the team's voice communications (comms). Tournament POV streams may also occur when individuals stream their gameplay for an online tournament. For example, many Hearthstone streamers regularly compete in open cups, tournaments, during their regular streams, thus providing their POV of the tournament to their chats.

Professional. Streaming is not a full-time job for all streamers. Whether they make money through esports or game development, video game professionals can profit from the expertise in their selected games by streaming on platforms like Twitch and YouTube. Esports' personalities include competitors and casters. While the top casters and competitors can have a similar game knowledge, they also bring different expertise to their streams. Competitors are amongst the best in the world mechanically, and casters are often better at expressing their gameplay. Considering the ladder systems associated with most competitive games, competitors will generally be playing against more skilled opponents. However, since many competitors do not have time to ladder, and many casters are former competitors, a skill discrepancy is not a guarantee to exist.

Individuals associated with the development and distribution of a game (developers) are another unique type of professional. Given their intimate game knowledge, developers can present a unique perspective and expertise. Depending on their role in development, developers exhibit a range of both knowledge and skill. Before the release of a game, developers can invite viewers into the development process and share early gameplay. Following the release, developers can engage in marketing on stream through gameplay and discussion. Developers can also continue to stream their game following its release.

Non-professional. Expertise streams are not limited to professionals. Twitch provides skilled players with many opportunities. As previously discussed, many competitive games have ladder systems. Though esports competitors are commonplace at the top of ladders, they are not alone. Streaming provides novice players an opportunity to build a following, court sponsors, and attract professional teams. Former and prospective professionals have also found that a streaming career can prove more lucrative than a middling professional career. Streamers like Shroud, Imaqtpie, and IWillDominate are former pros that have found success streaming. In the case of Shroud, he has moved on to streaming new games in similar genres to his professional career (i.e., shooters) and applies his expertise to these new games.

For non-competitive games, streams can still present expertise gameplay. One approach to expertise among non-professionals is through repeatedly playing the game, including unique playthroughs like challenge runs (i.e., playthroughs of a game which include rules not established by the game). For example, LobosJR is a popular streamer of the *Dark Souls* franchise. Throughout his many runs of the first *Dark Souls*, LobosJR has done a variety of challenge runs, including Fist Only, No Heal/No Bonfire, and 1 Million Souls runs. There are also streamers whose expertise is more genre-specific, like Arumba07 and Quill18, and will continue to play the same games with additional modifications (mods) or objectives.

Another unique approach to non-professional expertise is through speedruns. Speedrunners are a unique subsection of Twitch streamers that carry a lot of the narrowness of professional streamers but considerably smaller viewership. Like professional gamers, speedrunners are considered among the most knowledgeable for their respective games and are unlikely to stream much with other games. Speedrunners generally have more specific gameplay than professional gamers, with some in-game maneuvers being frame-specific. The amount of interaction is also considerably diverse since many speedrunners do not stream for the chat interaction but to ensure video proof of their gameplay in case they set a new personal best (PB) or world record (WR). However, also like professional gamers, they will occasionally stream games outside of their expertise.

Entertainment

Contrary to expertise streams, most streams are more entertainment-based. Entertainment streamers are not necessarily without skill; however, they prioritize entertainment. Variety-streamers, or streamers who play a range of games, are the predominant force in entertainment-based streams. Nevertheless, there are still entertainment streamers who play a single game. With the rise of Twitch's popularity, some celebrities have also joined the platform and streamed. Celebrities can be either variety or single-game streamers.

Variety. Streamers who choose to play a variety of games populate much of Twitch. Though they play a variety of games, many still have preferences. For example, Summit1g, Lirik, and GoldGlove are major streamers who primarily play first-person shooters (FPS). Players with the most variety in their gameplay will similarly be amongst those with the least expertise in the game, often playing the game for the first time on stream. CohhCarnage, one of the largest variety streamers, gets most viewers during blind playthroughs of new games. He is also known for doing franchise playthrough in preparation for the release of a new game, such as playing all the *Fallout* games before the release of *Fallout 4*. It allows both him and his viewers to catch up with a franchise before the latest installment. Variety streamers also add another level of entertainment that viewers would not have if they played the games themselves.

Single-game. There is a single type of entertainment stream in which the streamer does not play a variety of games, single-game entertainment streams. Non-professional single-game streamers fall into two broad categories: casual and competitive (Leith, 2018). Casual (i.e., non-competitive) games tend not to get much replay in streaming unless it has found a niche among the speedrunners. An exception that has dominated gameplay broadcasting for years now is *Minecraft*. Bacon Donut is an example of a longtime streamer known for playing *Minecraft*. Competitive games are unique among entertainment single-game streamers as these streamers often compete against professionals. Entertainment, for them, becomes a counterweight, with more entertainment necessary as expertise decreases. Some streamers, such as BoxBox or Cowsep, increase their skill with one character, or champion in the case of *League of Legends*, despite developers building their games for players to choose their character based on the needs of their team. This specialization allows them to bring in viewers and compete with professional players.

Celebrity. Individuals who have found fame outside of Twitch (e.g., athletes and musicians), will likely develop unique stream environments from other entertainment streams. For example, musicians like Brendon Urie or T-Pain may field an atypical number of messages about music, as Quentin Jackson and Demetrious Johnson would with mixed-martial arts messages. Therefore, their chats would consist of proportionately fewer interpersonal communication and game-related messages, which can further divert celebrity streams from other entertainment streams. Celebrity stream communities may also be less defined since celebrities are less likely than other streamers to stream according to a regularly recurring schedule.

Non-Gameplay

Along with gameplay streams, there is a growing number of non-gameplay streams. Non-gameplay streams are not entirely surprising as Twitch was born out of the popularity of gameplay streams on Justin.TV, a service intended for streaming real life (IRL). The most common types of non-gameplay streams are Just Chatting, Music and Performing Arts, Talk Shows and Podcasts, Art, and ASMR. Just Chatting is a broad category that is distinguished by its lack of gameplay. For example, when the Offline TV streaming house visited one of their housemates in Taiwan, they carried a specialized backpack (The Gunrun IRL Backpack). This backpack allowed them to provide a first-person POV of their travels to their viewers. LilyPichu, a member of Offline TV, has since used the backpack for more regular outings, such as eating out, going shopping, or visiting the dog park. Ki is an excellent example of non-gameplay streaming because, despite gaining a large portion of her following from playing *LoL*, she does regular Art and Music and Performing Arts streams. Many of her past Art streams have been instructional, such as an afternoon where she taught her viewers how to draw a cartoon version of T-Pain. Following the lesson, T-Pain joined LilyPichu to review and grade the drawings submitted by the viewers. She also does regular Music streams with her significant other Sleightlymusical, wherein she plays the piano while he plays the violin. Talk Shows and Podcasts, the fourth most common non-gameplay stream, is structurally similar to the podcast type of gameplay stream, without the gameplay. Offline TV member Pokimane started doing a podcast in January 2018; however, it is still in its early stages. Better examples of Talk Shows and Podcasts streams would be ItmeJP's *Dropped Frames* or the late TotalBiscuit's *The Co-*

Optional Podcast. Both streams have a recurring cast of streamers and a guest. They both discuss gamer and streamer related topics, with *The Co-Optional Podcast* prioritizing games and *Dropped Frames* prioritizing streaming.

PARASOCIAL RELATIONSHIPS

Horton and Wohl (1956) argued for particular roles taken by media users and figures when consuming traditional media (i.e., television). Media figures were in control of how they engaged with media users, and media users could choose to follow or unfollow media figures. Through continued parasocial interactions, users can develop parasocial relationships with these figures (Klimmt, Hartman, & Schramm, 2006). When the users no longer find these relationships satisfying, they can withdraw. Research has continued to find television and radio as consistent tools for developing parasocial relationships (Palmgreen, Wenner, & Rayburn, 1980; Rubin & Perse, 1985; Sood & Rogers, 2000; Park & Lennon, 2004; Spitzberg & Cupach, 2008).

Parasocial Interactions

Parasocial interactions are designated by the immediate response to media consumption (Schmid & Klimmt, 2011). For instance, a parasocial interaction is the instance of exposure that occurs as a media user consumes a television show. The television viewer is participating in an atypical interaction in which they are personally engaging with a piece of media in which the media figure (e.g., actor or presenter) intends for a vast audience. As media users continue to read books, watch television, and listen to the radio, they will continue to engage in these parasocial interactions. Media users will then perceive these parasocial interactions as more intimate, especially if they have had additional exposure to the media figures (Perse & Rubin, 1989).

Using external methods of contacting the figure (e.g., mail, phone) is possible but exists outside of parasocial interactions (Horton & Wohl, 1956). Since 1956, contacting media figures has become considerably simpler. The diminishing divide between parasocial and social interaction requires a more precise delineation between a parasocial relationship and a parasocial interaction. Parasocial interactions are distinct instances of media exposure in which media figures one-sidedly engage with media users, and parasocial relationships are illusory cross-situational relationships perceived by a media user toward a media figure (Klimmt, Hartmann, & Schramm, 2006). Users can interact with the figure in which they have a parasocial relationship; however, “these lie outside the para-social interaction itself” (Horton & Wohl, 1956, p. 215). These distinctions are especially significant when considering social media platforms (e.g., Twitter, YouTube, Twitch).

Social Media

The advent of social media platforms has provided a unique twist to user-figure relationships. Sites like Twitch now provide media users with a limited separation from media figures. Twitter and Instagram now allow celebrities the freedom to broadly share their thoughts and images without requiring outside intervention (e.g., magazines, newspapers). These celebrities are also able to integrate more personal information into their social media strategies (Colliander & Dahlen, 2011; Lueck, 2012). Conversely, media users can use these same methods to communicate directly with the objects of their parasocial relationships. When media users and figures interact through social media, figures can change the perceptions held by users (Thorson & Rodgers, 2006; Fredrick, Lim, Clavio, & Walsh, 2012), and figures can further facilitate parasocial interactions (Kim & Song, 2016). YouTube could be considered a perfect evolution of how Horton and Wohl viewed media figures on television. YouTubers, individuals who regularly create content on YouTube, are provided much greater control of how they will present their persona to their viewers. The YouTube platform also provides for numerous methods for connecting with the YouTuber, and the communication methods are asynchronous enough that YouTubers are in full control over with whom they wish to interact if they choose to interact.

This approach has proven to be effective in building relationships as 40% of millennials, individuals born between 1981 and 1996, who use YouTube reported being better understood by their favorite YouTubers than their real-world friends (O'Neil-Hart & Blumenstein, 2016). Rasmussen (2018) validated this assertion when they found parasocial interactions on YouTube.

The livestreaming platform Twitch takes an additional step beyond YouTube by providing a nearly direct means of communication with the media figure at the same time as the media figure is producing their traditionally one-side parasocial interaction. The resulting experience is a hybrid of sorts in which the Twitch streamer is consistently alternating between parasocial and social interactions. Unlike other media figures, Twitch streamers spend the majority of their time – 8-10 hours a day, 5-7 days a week – public facing, as opposed to traditional actors who can spend hours working on the same few minutes of product. Esports athletes, as another example, are even further connected to their fans than traditional athletes, as they regularly stream gameplay and interact with their chats (Taylor, 2012, 2018).

The combination of synchronous video and chat provides users an unprecedented degree of closeness, which was otherwise reliant on extensive engagement. Gameplay streamers spend most of their time and concentration playing and discussing their games. While playing, streamers will occasionally read donations and chat messages from their viewers. Parasocial interactions, in part, cease when the streamers are directly interacting with a viewer. Viewers can take advantage of the dozens of hours their favorite streamers are streaming, which far exceeds the number of hours most media figures engage with the public. These social interactions with individual viewers may still be considered parasocial interactions for other viewers as it is a more concrete example of "the gestures, conversational style, and milieu of an informal face-to-face gathering” (Horton & Wohl, 1956, p. 217) which assist in developing parasocial relationships. As viewers continue to experience instances in which streamers will mingle chat interaction with their streams, viewers may believe they are closer to the streamer, thus further strengthening their parasocial relationship.

Research Hypotheses

Though self-reported surveys have historically been the primary method for researching PSI (see Schiappa, Allen, & Gregg, 2007), natural language processing (NLP) allows for the investigation of parasocial relationships through the review of media user communication tendencies. This dissertation is particularly interested in how users differentiate their communication on Twitch between media figures (i.e., streamers) and other media users (i.e., viewers) in the synchronous chat provided on each channel. Similar to other social media sites, viewers can target their messages at particular users by using the at symbol, @, with the intended target’s username, such as @APLeithTV. A targeted message will appear highlighted in the chat of the intended recipient. NLP can be used to investigate these messages for signals of personal and experiential language (Borelli, Sbarra, Mehl, & David, 2011), which identify psychological closeness and immediacy (Pennebaker & King, 1999).

Bazarova, Taft, Choi, and Cosely (2013) found that individuals who reported a greater partner familiarity with their communication partner would vocalize a greater verbal immediacy through their messages. Because parasocial relationships are characterized by the perceived relationships held by a media user toward a media figure, verbal immediacy should prove to be a strong predictor for parasocial relationships. Namely, this dissertation argues that parasocial relationships exist if streamer-targeted messages present a greater verbal immediacy than other-targeted messages, which largely serve as a control group.

RH 1: *Streamer-targeted messages* from the chats from *all streams* will have a greater verbal immediacy score than *viewer-targeted messages* from the chats of *all streams*.

A specific component for developing parasocial relationships is through parasocial interactions, which more closely reflect face-to-face communication (Horton & Wohl, 1957). In the case of Twitch, nongame streams should more closely reflect a face-to-face communication than game streams as game streams introduce additional barriers to the communication process that do not traditionally exist. Despite the introduction of the stream type variable, streamer-targeted messages should still produce a greater verbal immediacy than viewer-targeted messages. These two variables should interact in such a way that nongame streams and streamer-targeted messages should produce greater verbal immediacy scores.

RH 2a: *Streamer-targeted messages* from the chats from *nongame streams* and *game streams* will have a greater verbal immediacy score than *viewer-targeted messages*.

RH 2b: *Streamer-targeted messages* and *viewer-targeted messages* from the chats from *nongame streams* will have a greater verbal immediacy score than from *game streams*.

The final set of research hypotheses continue the same arguments except extend them to the two primary types of game streams: entertainment and expertise. Like the difference between nongame and game streams, entertainment and expertise streams differ in the expected level of communication difference to nongame and game streams, respectively. Expertise streams require an increased level of focus than entertainment streams. Entertainment streamers often have more casual conversations with their streams than expertise streamers. Therefore, entertainment streams should produce greater verbal immediacy scores than expertise streams. The other research hypotheses further reflect the points from research hypothesis two with entertainment and expertise replacing nongame and game, respectively.

RH 3a: *Streamer-targeted messages* from the chats from *entertainment streams* and *expertise streams* will have a greater verbal immediacy score than *viewer-targeted messages*.

RH 3b: *Streamer-targeted messages* and *viewer-targeted messages* from the chats from *entertainment streams* will have a greater verbal immediacy score than from *expertise streams*.

It is necessary to examine what the convergence of streamers and viewers means for games. Twitch provides viewers nearly endless, direct access to streamers while the streamers are playing games. Twitch bots and human moderators are the only intrinsic limiters to what viewers are allowed to discuss (Taylor, 2018). Among the various topics, viewers who attempt to communicate directly with streamers by tagging them in their message (e.g., @APLeithTV), will likely attempt to discuss the game being played, which may include attempts to alter gameplay or game space.

Video games have carved a distinct niche in the landscape of both media and technology. Differing from traditional games through the underlying technology (see Loftus & Loftus, 1983), video games continue to more closely resemble other forms of media (e.g., television and movies) than traditional games. Nevertheless, many of the first video games graphically recreated classic games, such as tic-tac-toe, checkers, and tennis (Cohen, 2016; Donovan, 2010; & Link, 2012). Hackers from MIT advanced the medium in 1961 by creating *Spacewar!*, a game where players could pilot ships in an E. E. Smith inspired universe (Brand, 1972).

Along with technology distinguishing video games from traditional games, video games differ from traditional media through interactivity (see Andersen, 1992) and represent the next stage in the evolution of "make-belief" (Ritterfeld & Weber, 2006). From books to television to

Table 2

The Evolution of Make-Belief from Ritterfeld and Weber (2006)

Characteristics of the Medium	Books	Television	Video Games
Narrativity	Yes	Yes	Yes
Simulation	No	Yes	Yes
Interactivity	No	No	Yes
Intelligence	No	No	Yes

video games, each form of media provides additional characteristics. Namely, books provide narrativity, television provides simulation, and video games provide interactivity and intelligence (see Table 2).

Defining Gamespace

Defining gamespace, according to Wainess and Koenig (2010), requires the definition of both game and gameplay. Games are “the rules, goals, affordances, and effects within a game space;” gamespace is “the bounds in which game play occurs;” gameplay is “the actions that occur as part of a game” (Wainess & Koenig, 2010, p. 5). It is crucial, therefore, to identify what should be classified as components of a game to identify what occurs within gamespace. There are two general approaches to defining gamespace under these principles: intrinsic and extrinsic. Intrinsic is the more traditional approach in which in-game elements comprise the entirety of gamespace (Wainess, Kerr, & Koenig, 2011; Wainess & Koenig, 2010) while extrinsic refers to instances in which out-of-game elements are considered components of gamespace (Newman, 2002a, 2002b). Depending on game developers and streamers, Twitch may be able to be included within gamespace under either structure.

Intrinsic Gamespace

The base definition provided for gamespace outlines an intrinsic gamespace. From their original definition, Wainess, Kerr, and Koenig (2011) developed the Game Play Model and Player Interaction Framework as an attempt to visualize the agents of gamespace. The Game Play Model proposes the interaction between game components within gamespace and identifies four components: rules, goals, affordances, and effects (see Figure 1). Rules are limitations placed on what the player can and cannot do. Goals are milestones in which players meet in-

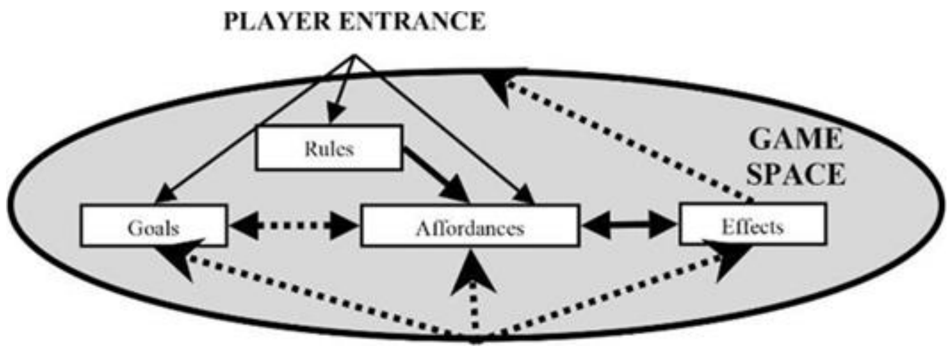

Figure 1. Game Play Model from Wainess and Koenig (2010)

game, end-game, or both. Affordances are the player's abilities. Wainess, Kerr, and Koenig (2011) argued that each of these components exists within the gamespace. The Player Interaction Framework elaborates on the Game Play Model such that individuals better understand how the player interacts within the developed gamespace (see Figure 2).

The Player Interaction Framework identifies three types of objects: presentation, background, and storage and workshop. Presentation objects are directly presented to the player, while background objects require additional effort. Storage and workshop objects are a "group of functions" that exist outside of primary gamespace (Wainess, Kerr, & Koenig, 2011, p. 6). The

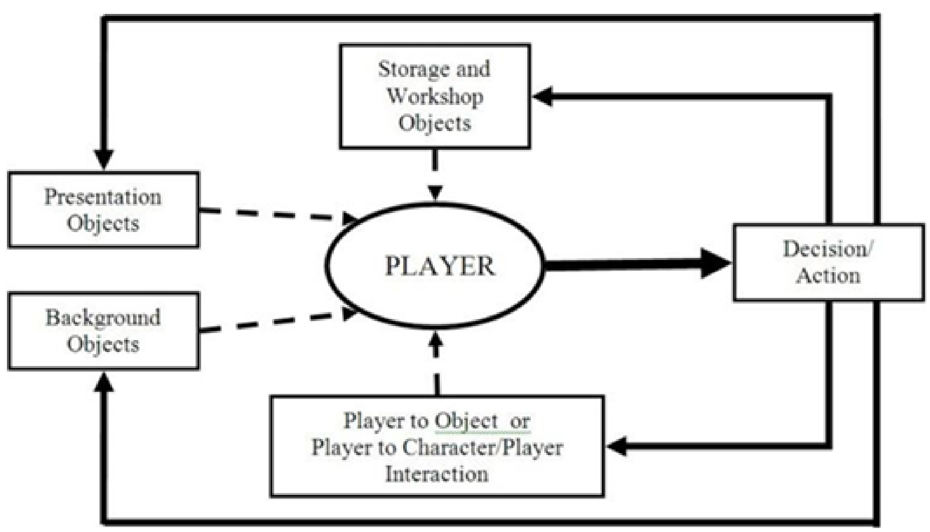

Figure 2. Player Interaction Framework from Wainess and Koenig (2010)

framework also identifies player-to-object or player-to-character/player interaction. In both cases, gamespace presents players with an entity with which they can interact.

Extrinsic Gamespace

Along with the intrinsically included elements within classic gamespace frameworks, elements beyond the digitally designed space may also be active agents within gamespace. The simplest example comes from old Odyssey games, which relied on physical components to fully utilize the digital game, such as translucent overlays, game chips, and cards (Newman, 2017). Actor-network theory is one approach previously taken to identify the roles of external elements, such as game systems and chairs, on traditional gamespace (Egliston, 2015; Taylor, 2009). Another potential element of extrinsic gamespace is human agents.

Unlike the Game Play Model, which depicts game design elements, the Player Interaction Model "depicts how a player interacts with information (instruction and assessment) in a game space" (Wainess, Kerr, & Koenig, 2011, p. 6). Newman (2002a) argued that video games are a structured and segmented experience. He purposely avoids defining it as an interactive media. By so doing, he can address the effects video games have on what he calls the "non-playing observer" (Newman, 2002a, p. 411) or "onlookers" (Newman, 2002b, p. 1). Individuals can consume videogames by watching others play and compare their skills (Taylor, 2012), turning places like arcades into public performances (Lin & Sun, 2011; Newman, 2004). Streaming gameplay extends this public performance and provides viewers the opportunity to contribute to the content (Anderson, 2017).

Technological advancements are regularly altering how individuals consume videogames. Each new generation of mobile phones can better support playing more complex games and viewing higher quality gameplay livestreams. Burroughs and Rama (2015) argued that both mobile gaming and gameplay livestreaming alter the magic circle. The magic circle is a closed system with absolute rules, and it begins once a game is started (Salen, Tekinbaş, & Zimmerman, 2004). Despite the magic circle protecting players from the world (Sniderman, 1999), the outside world is always invading gameplay (Fine, 1983). Consalvo (2009) argued that it would be better to consider gameplay occurring between frames than inside or outside of a magic circle.

Gamespace and the magic circle are distinct concepts. The magic circle and gamespace both concern the connectedness between the game elements and the human agents. The magic circle fundamentally differs from gamespace as the magic circle concerns a more cognitive state of gameplay engagement than gamespace, which is a technical approach to interpreting the interconnectedness of gameplay and external elements.

Human Agents within Gamespace

By prioritizing the roles of human agents within gamespace, this dissertation builds from both the intrinsic and extrinsic approaches to gamespace. Human agents can take two ingress routes into gamespace: direct and indirect. Direct human agents are those who are playing the game. Indirect human agents are those who connect to the game through the direct human agent. In traditional gameplay scenarios, whether in a living room or an arcade, the direct human agents are the player, and the indirect human agent is the onlooker. In the gameplay livestreaming environment, the direct human agent is the streamer, and the indirect human agent is the viewer.

Traditional Gamespace

There are two general roles taken by human agents in traditional gamespace: players and onlookers. Players, the primary human agents, are individuals who directly control the game. Onlookers, the secondary human agents, are individuals who, at best, indirectly control the game.

Gamespace inherently includes neither players nor onlookers; however, depending on their gameplay engagement, it is possible to make a case for their inclusion.

Players. The primary human agents are the players. Players have a degree of control over the environment in which they play games. For example, Newman (2002a) wrote about the physicality of playing video games, with players moving their bodies in response to the gameplay and tightly gripping their controllers. He argued that physicality prompted the changes in physical hardware, such that controllers would include vibrations to immerse players better. Specialty controllers have also been created to mirror steering wheels, tennis rackets, and fishing poles (see Figure 3). Gamespace may also include displays as the experience of playing a game on television is different than on a handheld system (e.g., Game Gear or Switch).

Many games are not limited to a single player. For such games, there are multiple human agents directly engaging with gamespace and other players. Social interaction has historically been a significant predictor for gameplay motivation (Sherry, Lucas, Greenber, & Lachlan,

Figure 3. Specialty Controllers

2006). It is not always possible to get people together to play a video game. Because video games exist in the digital space, a non-collocated multiplayer is possible.

Video games have grown alongside the Internet, having existed online since the 1980s. Internet advancements have also allowed for multiplayer games to develop communication tools. Whether text or voice, in-game communication allowed players to organize better and complete the tasks set by the game. Communication tools also provided for closer communities of players. For example, *Doom* and *Quake* were two of the first games to latch onto the developing culture of the Internet. The communities for both games were passionately involved with the game and developed virtual communities of networked gamers that competed online (Lowood, 2006). These networks are, therefore, able to limit the effects of geography on interconnectivity (Mitchell, 1995). When competing against other players, in-game communication also provides for another means of conflict when competing players directly communicate with each other.

Onlookers. The gameplay experience is not limited to the individuals actively playing. Along with players, video games will often include onlookers. Newman (2002a, 2002b) proposed several ways in which gamespace may include individuals not controlling the game. Arcades provide insight into the role of onlookers, especially when examining it within the scope of traditional proxemics. Borrowing from Hall (1966), there are four key communicative distances: intimate, personal, social, and public. Intimate space is 0-18 inches and reserved for close relationships. Personal space is 18 inches - 4 feet and for friends. Social space is 4-12 feet and for acquaintances and strangers. Public space is upward of 12 feet and for speeches and presentations. Arcade goers can be seen to adopt this spacing.

Imagine you are presently playing the 1989 *Teenage Mutant Ninja Turtles* at a busy arcade. Despite the business, you would expect that anyone within your intimate space would be there of your choosing. Those people within the intimate space would be the most likely contributors to the gameplay experience, having the fewest hurdles to clear communication. The further out you move, the harder it becomes to influence your play. Similar to intimate space, the majority of people within your personal space are those with which you have a relationship. They would, therefore, be able to communicate with you, and you would be more likely to follow their advice. Onlookers in your social space would have a harder time positively contributing to your gameplay; however, it is not impossible. Communication nearly breaks down entirely once the onlooker is in your public space, allowing nothing more than basic utterances being understandable through the noise. It is challenging to find onlookers at the public space level in an arcade, due to its crowdedness, but the growth of esports has made public space onlookers an increasingly common occurrence for gamers.

Streaming Gamespace

When examining the role of human agents within gamespace for streaming, it is necessary to revisit the broader human agent categories from traditional gamespace: player and onlooker. The human agents involved in streaming gamespace adopt new roles. Players generally become streamers, a role that includes additional roles. There are instances, such as tournaments, in which the players are not necessarily the ones running the stream; however, the player would still maintain most of the other characteristics of a streamer. Onlookers are separated from the players in streaming, requiring a text-based chat to communicate with the player, thus relegating them to the role of a viewer.

Streamers. Players take on new roles when they choose to stream their gameplay; however, streaming includes more than just broadcasting gameplay. Streamers are uniquely involved in developing distinct gamespaces. Namely, streamers create specialized environments, or "bounds in which game play occurs" (Wainess & Koenig, 2010, p. 5). They have individually developed personal approaches to what gamespace means for them. With streamers ranging from experts to novices in the games that they are playing, how each streamer approaches a game can differ. Similarly, streamers can differ in their approach to livestreaming.

Regardless of their approach, streamers are fulfilling multiple roles during a livestream. At the least, the average independent streamer is serving as the engineer, producer, and on-screen talent. Streamers set up their equipment for streaming and troubleshoot their broadcast problems. Though some streamers have very minimal stream setups, it is not uncommon for a streamer to build relatively complex broadcast studios to deliver a more professional livestream. The streaming setup will often reflect the streamer's produced content. The streamers must decide whether they want to broadcast their voice over gameplay. Many streamers choose to also broadcast video of their face in the corner of the livestream video (Anderson, 2017). Other than the video, streamers must choose whether to include other content in their overlay, which may require them to become graphics editors.

Depending on their approach to the additional elements of streaming, streamers may be inviting additional components into their gamespace. For example, ItmeJP, one of the most senior and prolific streamers on Twitch, has developed a relatively elaborate stream setup (see Figure 4), which includes multiple PCs and professional audio, video, and lighting equipment (itmeJP, 2017a), to meet the needs of the diverse content he streams, including podcasts and a variety of tabletop and video games. ItmeJP's viewers are a vital component of his streams and are, therefore, invited into his gamespace while he is playing a game. A more literal convergence of stream setup and gamespace occurred with another streamer: CohhCarnage. While on one of ItmeJP's podcasts (2017), CohhCarnage recounted his time streaming *Elite: Dangerous* in virtual

Figure 4. J. P. “itmeJP” McDaniel Stream Setup

reality (VR). As seen with ItmeJP, streamers will often run multiple screens in their setup, allowing Twitch chat to be viewable on one monitor while gameplay is occurring on another. CohhCarnage personally has four screens in his setup. Designers did not intend for VR headsets to have such a setup. To compensate for the missing monitors, CohhCarnage used third-party software to put the chat on his VR screen (itmeJP, 2017b). The newly created gamespace was CohhCarnage’s answer to imposed limitations, allowing him to maintain the same gamespace in which he usually plays.

Viewers. Regardless of the circumstance, Twitch actively promotes a perceived closeness between streamers and their viewers. Viewers, in turn, become active participants in chat to communicate with other viewers and in hopes of communicating with the streamer. The desire to feel close to the broadcaster, or other viewers, is one of several potential motivations for watching streams. Taylor (2018) explicitly listed six motivations: aspirational, educational, inspirational, entertainment, community, and ambiance. Streams on Twitch are all accompanied by a synchronous chat service that allows viewers to interact with each other and the streamer, thus meeting the viewer's desire for community. Social information processing theory (SIP) posits that individuals will adapt to the media they use to make the most of it, though it takes more time through mediated than face-to-face communication. The synchronous chat will produce a minimal amount of information per input (i.e., utterance). Therefore, the synchronous chat must occur over an extended period in order to accumulate a similar quantity of information (Hian, Chuan, Trevor, & Detenber, 2004; Wilson, Straus, & McEvily, 2006). Prior research has noted asynchronous communication media as a better information source (Peter, Valkenburg, & Schouten, 2005; Ramirez, Zhang, McGrew, & Lin, 2007). Twitch provides minimal asynchronous communication and a fledgling social network, so the bulk of communication would occur over chat. Because the medium does not differentiate between gameplay and non-gameplay streams, just the content, it is first necessary to identify the communication practices of all Twitch users.

Streams exist outside these communicative spaces, with streamers being geographically separated from their audience (see Figure 5). The inclusion of chat into the streamer's environment allows for these spaces to be artificially mirrored. As it was with onlookers in the arcade example, viewers can take on three general roles: instructors, learners, and admirers. Each of the three roles can be vocal elements of the gameplay livestream experience.

Instructors can give streamers advice through chat. For competitive games, viewers can inform streamers of better strategies or how mechanics work. Viewers can also provide insights into more casual, narrative-driven games. Streamers can then choose to what extent they review the chat, with some streamers completely avoiding game-related information from their chat while others welcome it.

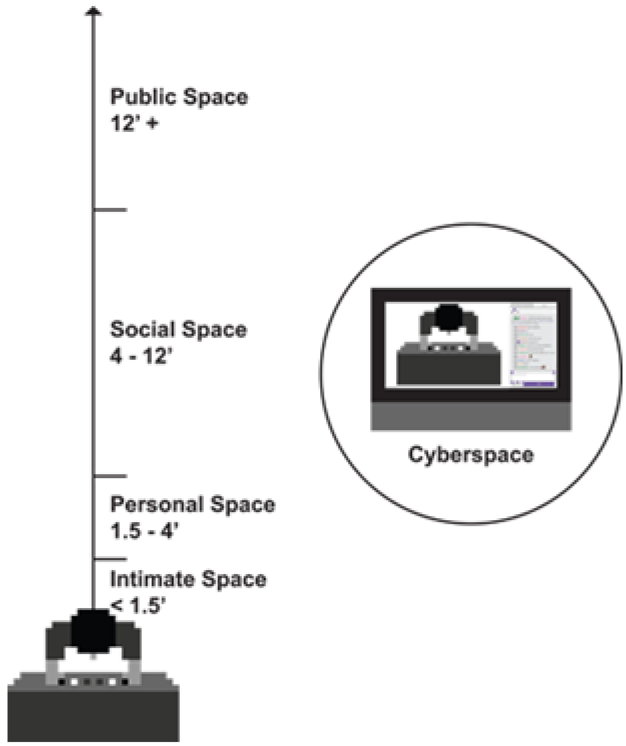

Figure 5. Gamespace Expansion

Learners can take on a more active role than their arcade counterparts through the chat system without imposing on the streamer. Namely, it is less invasive to pose questions over a text-based chat than shout the questions in the arcade setting. Some streamers, such as Day[9], began streaming to supply information to these learners. For Day[9] (2010), a former *StarCraft: Brood War* pro, the Internet provided him an opportunity to share strategy through guidebooks and gameplay videos on forums, such as Team Liquid. Sharing videos led him to create the Day[9] Daily, which focuses on strategy sharing. Sharing has also been done through live-streaming, allowing the chat to add information or ask questions as he went over prior gameplay.

Admirers are a growing subset of video game players due, in part, to the rise of esports. Major tournaments are akin to traditional sports tournaments, allowing fans to cheer for their favorite players or teams from the stands (Taylor, 2012). Fans can also find these eSports personalities livestreaming. Through livestreaming, these fans can make use of the synchronous chat as a substitute for in-person cheering and statements of support. eSports professionals are not the only gamers that acquire admirers through a livestream. A large portion of streamers on Twitch, for example, would not be considered experts at the games they play; however, they still invite others to watch them play. Non-expert streamers can lean into the public performance elements of the arcade culture (see Lin & Sun, 2011; Newman, 2004) and create enjoyable environments for their viewership.

Research Questions

Gameplay livestream viewers are rapidly changing what it means to enjoy gameplay. Unlike other human agents, viewers are entirely physically removed from the gameplay. As platforms, such as Twitch and YouTube, attempt to lessen these separations through text-based, synchronous messaging systems, viewers should be able to take on similar roles as their in-person counterparts. Just because viewers can, it does not mean that they are taking on similar roles as their traditional onlookers. PSR argues that individuals are more likely to have increased verbal immediacy toward the media figure and, in turn, have an increased likelihood to engage with these streamers. Of the various subjects viewers may choose to address, viewers can attempt to alter or influence the gameplay. As with traditional onlookers, viewers may be included within the gamespace as they take on the roles previously defined as components of gamespace. It is for this purpose, identifying the potential inclusion of viewers into gamespace, that this dissertation proposes the following research questions.

RQ 1: Are *streamer-specific messages* for gameplay streams attempting to engage with the gameplay?

RQ 2: Do these messages elevate the senders to inclusion within gamespace?

METHODOLOGY

This dissertation will take advantage of the growing field of machine learning to identify the presence and significance of parasocial relationships and gamespace engagement. A set of bots will be used to collect and log the chats of partnered streamers from a variety of game and nongame categories. LIWC (Linguistic Inquiry and Word Count) will then identify variances across groups and t-tests. ANOVAs will be used to validate the variances. N-grams will then be used to address gameplay-related messages.

Data Collection

Third-party sites provided an updated list of the most popular Twitch categories for each condition (e.g., gameplay and non-gameplay). A series of bots augmented for this dissertation were then deployed to collect stream and chat data from the most popular streams and chats from each of these categories. These Python bots logged all collected data into a PostgreSQL database. Due to the bots using a deprecated version of the Twitch API, the bots collected data beyond the scope of this study and required manual coding to restrict the data to the most appropriate selection.

Category Selection

Third-party websites provide for a fair and balanced approach to identifying which categories to log for this dissertation. Sites like Twinge, TwitchMetrics, and SullyGnome closely track which categories are the most popular in both viewer count and the number of channels streaming in that category. Nongame categories were first selected from the historical data to identify which five nongame categories were consistently the most viewed and streamed. The nongame categories were limited to these five as most streamers will stream under the Just

Chatting category as long as their content is loosely related and nongame categories beyond the top five are only sporadically popular enough to produce any meaningful data.

Most game categories are widely more fluid in their number of streamers and viewers than nongame categories, so a broader swatch of categories was necessary. TwitchMetrics (2019) and SullyGnome (2019) identified the twenty most popular categories using two separate metrics: most streamers and most viewers. These sites were periodically rechecked throughout the month to identify changes in popularity. Bots were then regularly readdressed to the most popular categories as the month progressed. The most common purpose for these changes is that variety streamers, which are streamers that regularly change the game they are playing, will often switch to the newest games to maintain viewership.

SullyGnome was used to assist in tracking these changes as it would list the most popular categories from the last three days. Using the three days data would ultimately result in not tracking a category as soon as it gains popularity; however, this method also allowed for the bots to not immediately transfer to a category just because a couple of popular streamers decided to be streaming under an otherwise unpopular category during the time in which I am checking for the most popular categories. Single channels would also be able to distort the data since both GamesDoneQuick and Yogscast were running major marathons during this time, and both channels are notorious for streaming atypical content. Therefore, it was best to ensure that the bots only logged these channels when they otherwise fit within the scope of this dissertation.

Twitch Bots

Data collection first required the development of bots that are capable of identifying specific Twitch channels, logging their information, connecting to their channels, and logging their chats. Fortunately, Twitch's API (the API) allows developers to collect stream, channel, and viewer data, and sites like GitHub provide an insight into various approaches others have taken with the API. This dissertation used a modified version of a program named Twitch Chat Logger (TCL), which was uploaded to GitHub by Bernardo Pires. TCL is licensed using the MIT License.

TCL is “A simple python app for logging twitch’s chat to a PostgreSQL database” (Pires, 2015). TCL uses the now deprecated “fig up” command to create a Docker container. Within the container, TCL will create the database, twitch, with two tables, stream_log, and chat_log, and run the core TCL Python app (the app), which consists of a loop following the initial procedures.

First, the app identifies the top streams by current viewers. Second, the app logs stream statistics for top streams to the stream_log table on the twitch database. Third, the app creates bots to join the chats for the top streams. Fourth, the bots join the chats. Fifth, the bots begin logging joined chats to the chat_log table on the twitch database. Sixth, the app rechecks for top streams sixty seconds after the last top streams check. Seventh, the app logs stream statistics for new top streams to stream_log on twitch. Eighth, the bots part chats of channels no longer listed among top streams. Ninth, the bots stop logging parted chats. Tenth, the bots join chats of channels not previously joined. Eleventh, the bots begin logging newly joined chats to chat_log on twitch. From henceforth, the app continues to recheck, part/join, and log (steps 6-11) until stopped (see Figure 6). It is also noteworthy that a bot continues to log a chat until it parts, regardless of how many top stream checks occur after the bot joined. Meaning, bots continuously logged the chat until the bots remove its associated channel from the top streams list and parts from the chat.

This dissertation used a modified version of this logger to address its more targeted needs. The new logger (Logger) removed the Docker functionality to provide for more nuanced

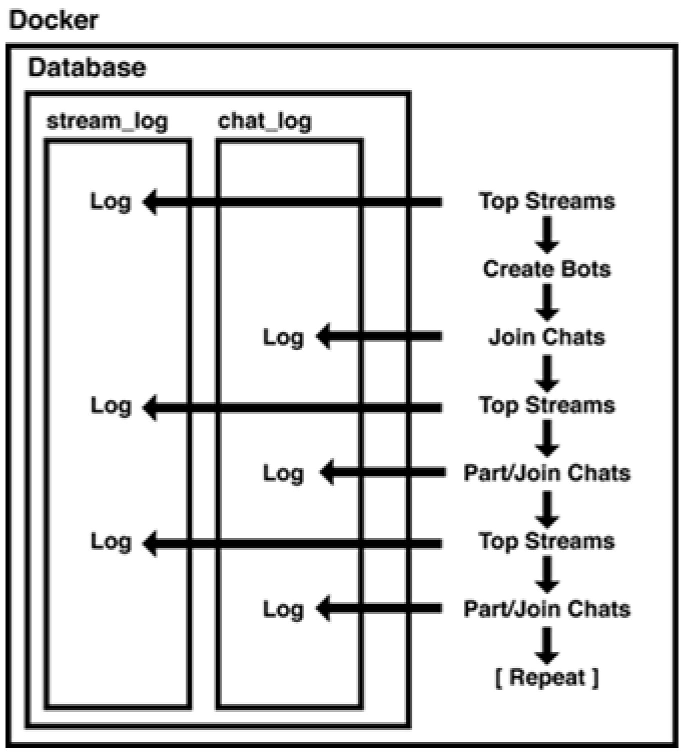

Figure 6. Twitch Chat Logger

control over the individual bots and created tables. Logger also allows for the identification of specific Twitch categories (e.g., Hearthstone or Just Chatting) and other stream content (see Appendix).

Stream selection. The new program identifies four key stream limiters: categories, amount, language, and partner. Due to the core nature of Twitch chat, they have generally coded categories as games. Amount signifies the number of channels to log per category. Language is the language designated for the stream. Partner refers to the partner status of the streamer.

Categories are the most substantial addition to the original TCL. By including a category variable, a unique set of bots are necessary for each category. The number of total categories is fluid, with a greater number of categories concerning games. There are fewer nongame categories included since there are far fewer popular categories within this subset.

The change in amount is the most essential modification to TCL, with the original designer allowing the user to select the number of streams to log in the Python command when they run main.py, the primary Python script. Due to the inclusion of category specificity, the default setting of 100 streams cast too broad a net. Instead, Logger was altered to collect 50 streams by default. The capacity to identify a custom number of streams, up to 100, remained in the code.

Language is a variable in the API that denotes stream language, not the native language of the broadcaster. Multilingual broadcasters will often stream in one of their non-native languages, especially if they feel that it will broaden or increase their viewership. English is a common choice for broadcasters seeking a more international audience. It is also not uncommon for non-English tournaments and events to simultaneously run an English stream. Therefore, only collecting data from English speaking streams should still provide a broad portrait of chat practices. Furthermore, language deficiencies would require the inclusion of additional researchers or a blind reliance on tools to analyze non-English chats.

Partner is a variable in the API that denotes whether or not a broadcaster is a Twitch Partner (partner). A broadcaster is generally required to have already established an audience (including both viewership and chat activity) that is continually growing and maintains a regular, at least three times weekly, broadcast schedule to become a Partner (Twitch, 2017). Therefore, limiting top streams only to include partnered streams will better ensure that there is sufficient activity to analyze chat logs and history to develop norms. For example, there are occasions where a new channel will have a large viewership but a disparate chat. Similarly, there is the potential for an older channel to sporadically stream to a moderate number of viewers but to have few viewers in chat. Limiting the top streams to only partners would help avoid these conditions.

Stream variables. Despite the modifications to TCL, Logger collects the same five stream variables: channel, title, game, viewers, and time. Channel returns the name of the channel. Title is the status attached to the current stream. Game is the variable that was broadly defined as category earlier, as it includes nongames. Viewers and time are both the current totals at the time of the logging.

Each Twitch channel has a unique channel name. The name is formatted exactly as it appears in the URL. Users can regularly change their channel names; however, clipped segments of past streams are still discoverable using the channel names used at the time of the original broadcast.

Title is a changeable status given to the channel. A title is a tool for gaining viewers by distinguishing one channel from other similar channels and defining the present stream content. Streamers will generally use their status to inform viewers when they are raising money for charity. Speedrunners have also historically included "#nosrl" to their status when they are not speedrunning so that third-party sites would exclude them from lists of currently streaming speedrunners.

The game variable returns the listed game presently streamed. Non-gameplay streams will similarly list their activity under the game variable. Therefore, the Logger will return a list of games that include both games (e.g., *PLAYERUNKNOWN'S BATTLEGROUND* ["*PUBG*"], *Hearthstone*, and *Dungeons & Dragons*) and nongames (e.g., Creative, IRL, and Music). It is necessary to recognize that the game is, in itself, insufficient in defining the stream. It does, however, serve multiple purposes. For example, a channel that only ever streams one game is more likely to contain expert gameplay than one that streams several games. The game, or genre of game, can also be used to differentiate between streams. For example, if anyone examined the chat as an extension of gamespace, then they may conclude that chat can similarly differ as gamespace does. Conversely, there is also the potential for no difference as the viewers' component of gamespace remains unchanged.

The viewers variable returns the current number of viewers for the stream. The current viewer count provides a rough pool size for participants in the chat. An individual can join chat without viewing the stream; however, a precursory examination has found this practice to account for a minimal number of users and did not produce a chat that outnumbered the current number of viewers, with it also being possible to view the channel without joining the chat. Understanding the size of an audience will provide insight into its effect on communicative behaviors. The dissertation will also be able to distinguish high variance streams and how chat responds. High variance streams are those that alternate between high and low viewer counts.

Time is another numerical output. The time is formatted in milliseconds. For example, the start time for data collection (December 17, 2018, at 10 a.m. UTC) was 1545040800000. Though difficult to immediately interpret, treating time natively as an integer allowed for easier sorting. Analytical tools, such as R, also have native packages for converting time to a more visually appealing format when it is necessary.

Chat logs. Logger collected four pieces of data from chats: channel, sender, message, and date. The channel, sender, and date variables are all classified as utility variables, with no applications to theory presently identified. Channel distinguishes the name of the IRC channel, which is the same as the Twitch channel to which it relates. Sender is the name of the Twitch account used to join and participate in the chat. Date is the date and time of the message to the millisecond.

The significant part of the chat logs is the message content. The Logger collects chat content as plain text, without hyperlinks or emotes. Logger is incapable of logging user icons (e.g., staff and moderator). The bots otherwise log chat as it appears. It is important to note that, despite the emotes being lost in the log, the text which produces the emotes will remain. Emotes are a core component of Twitch chat, with each emote serving a specific communicative purpose (see Table 3), and will, therefore, require interpretation to be accurately analyzed.

Pi cluster. To isolate the bots and data from other workloads and to limit the overall power consumption, this dissertation employed a Raspberry Pi cluster (Pi cluster). A Raspberry Pi (Pi) is a small single-board computer mostly used by hobbyists and for internet-of-things (IoT) applications. A single-board computer is a motherboard that includes onboard graphics, processor, and RAM. Because a Pi has a low cap on processing (1.4GHz) and RAM (1GB), a Pi cluster was used to allow into resource distribution.

A Pi cluster can infinitely scale to allow for the needs of its users. This Pi cluster consists of four Pis. One Pi serves as a controller while the other three serve as slaves. The controller has

Table 3

Top 10 Emotes of 2017 (Freitas, 2018)

Text	Icon	Meaning
Kappa		Tags statement as sarcasm or joke.
PogChamp		Expression of surprise or celebration.
4head		The streamer is cracking up or missed on a joke.
MrDestructoid		Suggests chatter is a bot or streamer is using a viewbot.
SwiftRage		Expression of anger or excitement.
OpieOP		Stands for XBOCT, derp, or an overweight person.
PJSalt		When someone is salty.
<3		Represents a heart.
Keepo		Dota 2 variation of Kappa emote.
EleGiggle		Replaces LOL or haha/jaja.

a 128GB microSD, and the slave Pis each have a 32GB microSD. Along with being physically networked together via ethernet cables and a network hub, the Pi cluster also connected via backend codes that allowed the Pis to communicate over SSH. Though the bots were not coded for cluster computing, running the Pi cluster provided resources natively designed for clusters to the elements.

The Pi cluster housed the Python bots and the PostgreSQL database. Python bots actively ran through the month on the Pi cluster and logged all stream and chat data to a single PostgreSQL database. Python bots logged stream and chat data to a single PostgreSQL database. The database consists of several tables. There was a single table for all stream logs and individual tables for each of the game and nongame categories. During the month-long data collection, the Pi cluster stored the database. Tools from pgAdmin 4 and JetBrains DataGrip regularly backed up the database to external drives over SSH. After data collection, the full database was dumped onto multiple storage devices to ensure duplicates. SQL syntax was then used from the primary work computer to output .txt and .tsv files that consist of the information necessary for the channel coding and analyses.

Data Cleaning

Due to issues that can arise following the deprecation of an API, some limiting variables built into the bot no longer worked. The broken limiting variables relevant to this dissertation was channel partnership status and stream language. Data cleaning requires checking for these limiting variables for each stream and cleaning for more anemic channels. Anemic includes viewer count, viewer consistency, and message counts. The final stage of data cleaning is coding the remaining channels for expertise and entertainment for each of their streamed categories.

Data cleaning began with the first stages of identifying nonanemic channels: viewer count and viewer consistency. Pulling all channel instances from the stream log table in which the channel had at least 75 viewers identified channel names and basic viewer statistics. Because Twitch requires an average of 75 viewers to earn partnership (Twitch, 2017), this dissertation limited chat logs to ones in which a chat log had at least 75 viewers. The channel instances were then imported into R and counted. This dissertation then removed all channels with fewer than ten instances of 75 or more viewers from the master list of channels for further inclusion. With the bot rechecking for channels every 120 seconds, data cleaning removed all channels which failed to have 75 viewers for at least 20 minutes throughout the month of data collection.

After reducing the number of channels, data cleaning required using the New Twitch API to identify which of the remaining channels have partnership status. The API returned a JSON with channel data for each of the remaining channels. All the channels not listed as a partner were then removed from the list of channels to be analyzed. Because channel names were used instead of the channel number, the New Twitch API defacto removed partnered channels from the dataset that changed their names between data collecting and data cleaning as the API request for those channels would not have returned channel information.

Following the identification of partnered channels, manual coding analyzed the remaining channels for English-speaking and expertise. English-speaking streams include both English broadcast and English chat. Coding involved visiting each channel, loading an arbitrary broadcast (video-on-demand or VOD) at an arbitrary point in the broadcast, and checking the video and its accompanying chat for its language. Though other languages might appear in either a spoken or written format, English had to be the dominant language within the VOD. English was essential since the developers of the analytical tools and the researcher for this dissertation are English-speakers.

While coding for English-speaking, coding analyzed channels for their expertise regarding the games played. Channels self-identified as experts as opposed to relying on more subjective researcher coding. Namely, channels could identify as experts according to the information provided through the panels on the channel's page or through associated social media (e.g., Twitter and Instagram). The simplest identifier for expertise is team affiliation as a pro or history of competitive success, including speedrun rankings. Team affiliation as a streamer was not considered a signal of expertise unless a previous history as a professional or record of being highly ranked accompanied it. Channel panels and social media also provided information about company associations and if there is any history of casting for a game. For each channel, analysis coded all non-expertise games into the general entertainment category. Though not entirely accurate as expertise and entertainment exist more precisely along a spectrum, all non-expertise streamers are identified as entertainment.

The final stage of cleaning was to exclude data subsets that produced insufficient data. SQL calls from the various chat logs tables stored in the PostgreSQL database for this dissertation produced the data subsets necessary for identifying which conditions had insufficient data. All hypotheses split the chat logs into streamer-targeted messages (i.e., messages that tag the streamer using their channel name and the at, @, symbol) and other-targeted messages (i.e., messages that tag another user and does not tag the streamer). The PostgreSQL specific command ILIKE was used to identify streamer-targeted messages and other-targeted messages, with NOT ILIKE being used to exclude streamer-targeted messages from the other-targeted messages data subset since a message being streamer-targeted supersedes it also being other-targeted. LIWC then conducted a word count on each of the .txt files produced by the SQL call. Hypotheses two and three further force the division of chat logs by category. The SQL select function allowed for the filtering of chat logs according to stream category.

Each hypothesis test requires a minimum amount of data from each channel for each condition. Borrowing from past research which required 25 to 50 tweets per user (Coppersmith, Harman, & Dredze, 2014; Fink, Kopecky, Morawski, 2012) and the best case for average tweet length being 15 words (Arnoux, Boyette, Mahmud, Akkiraju, & Sinha, 2017), data subsets were required to have a minimum of 375 words. A condition is the combination of channel, category, and message target and results in each channel producing two data subsets for each category under which they stream. For example, Northernlion would produce two data subsets from their *Minecraft* streams and two data subsets from their *PLAYERUNKNOWN'S BATTLEGROUNDS* (*PUBG*) streams. If one of these four conditions produced a dataset with fewer than 375 words, then its paired subset (i.e., the subset with the same category but different message target) would also be removed. Returning to the Northernlion example, if the streamer-targeted messages from their *PUBG* streams were the only data subset to produce fewer than 375 words, then further data cleaning would remove both *PUBG* data subsets and both *Minecraft* data subsets would remain.

Data Analysis

There were two phases of analyses. First, LIWC analyzed each of the data subsets for their verbal immediacy. Verbal immediacy was used to test for the parasocial relationship claims made by the hypotheses. Second, n-grams were run on the streamer-targeted messages from gameplay streams to identify instances of gameplay engagement. Gameplay engagement is the central focus of the research questions.

Parasocial relationships. The first phase of analysis is testing for parasocial relationships by identifying verbal immediacy variance across a range of condition sets. Hypothesis one compares all streamer-targeted messages against all other-targeted messages. Hypothesis two splits messages according to whether the stream content was gameplay or non-gameplay. Hypothesis three removes streams from nongame categories and splits game streams between expertise and entertainment. This dissertation repurposes past work from Bazarova, Taft, Choi, and Cosley (2013), which used LIWC, a tool that systematically counts instances of words from specially developed and validated dictionaries (Pennebaker, Boyd, Jordan, & Blackburn, 2015), to score verbal immediacy as a predictor for partner familiarity, to identify parasocial relationships. The argument extension here is that the presence of greater verbal immediacy in the streamer-targeted messages - messages between the user and the streamer - than in the other-targeted messages - messages between the user and another user - suggests a stronger perceived closeness with the streamer than with the average user.

Verbal immediacy was calculated using the arithmetic score of a set of LIWC scores calculated from the full subset from each chat log, not from individual messages. This approach to verbal immediacy expands the approach used by Bazarova, Taft, Choi, and Cosley (2013) but with the updated LIWC dictionary. The selected LIWC variables are first-person singular pronouns, present focus, discrepancies, words greater than six letters, and articles. The verbal immediacy score employed the additive inverse scores for words greater than six letters and articles.

Words greater than six letters required an additional step as emotes constitute a significant part of Twitch chat and they appear in text format within the data set. The newly calculated score for words over six letters was the prior score minus the score from emotes longer than six letters. Long emotes were identified by pulling all emotes on Twitch through the API and then counting the length of each emote. Despite filtering for long emotes, it is still possible that inverse score for words greater than six letters may far outscore the remaining variables and, as a result, produce an overall negative score. This negative score should still provide the necessary insights into communication trends among livestream viewers.

Following the calculation of verbal immediacy scores, a paired-sample t-test and mixed-model ANOVAs analyzed the variance between the separate variables. A paired-samples t-test was used for the first hypothesis because it involved the same groups of people within both variables. For similar reasons, mixed-model ANOVAs examined the second and third hypotheses. Streamer-targeted and other-targeted messages include the same communicators and channels (i.e., within-subject); however, the category comparisons split the participants into separate groups (i.e., between-subject).

Gameplay engagement. The second phase of analysis identified the attempts at gameplay engagements made by viewers. Both research questions were concerned with how viewers are attempted to engage with gameplay when they were communicating with the streamers. Instead of identifying general communication trends about gameplay, this dissertation distinguished streamer-targeted messages from gameplay streams as the relevant data set. Due to the impracticality of manually coding the full subset of data, n-grams filtered the data according to message commonality. After uploading the relevant data into R, tools from the tidyverse package identified the most common quintgrams, a recurring set of five contiguous words, that only include dictionary words and at least one discrepancy word (e.g., could, would, should). Discrepancies were an essential inclusion because it is a component of identifying parasocial relationships, which should make it easier for users to engage with streamers on a less superficial level, and the inclusion of a discrepancy suggests the desire to engage actively with the streamer.

Once identified, methods from grounded theory analyzed the developed corpora. Grounded theory is a qualitative methodology that uses multiple stages of coding to systematically examines text with the intention of developing new theories (Glaser & Strauss, 1967; Strauss & Corbin, 1994). Analyzed text can either be elicited through surveys and interviews or extant. This dissertation uses a form of extant texts as it compiles messages that exist naturally without research interference (Charmaz, 2006). Coding is traditionally accomplished through three steps: initial, selective, and axial (Glaser, 1978; Glaser & Strauss, 1967). Charmaz (2006) argued that open coding's role was to break apart data into separate codes. N-grams replaced the open coding stage. Selective coding takes a more directed look at the developed codes (Glaser, 1978). Due to its focus on gameplay engagement, this dissertation required the exclusion of messages that did not attempt to engage with gameplay or gamespace. Therefore, selective, or focused, coding was used to synthesize the remaining messages into the most significant groups (see Charmaz, 2006). Selective coding grouped the full messages for these excerpts according to its content. The final step of coding is axial coding, and it serves to reconnect data (Strauss & Corbin, 1998). Axial coding identified the final categories of gameplay and gamespace.

RESULTS

From December 17, 2018, through January 17, 2019, a series of Python bots systematically checked a defined set of Twitch categories every two minutes for the most popular channels at that time. Due to a few issues that arose with the deprecation of the older Twitch API, the bots logged more than the intended number of channels. In total, the bots collected data from 117, 943 channels. 6,564,307 viewers of these channels sent 321,189,309 messages over that month. The data cleaning process to limit the dataset to targeted messages, as outlined in the methodology chapter, drastically reduced each number (i.e., 3,127 channels, 1,298,148 senders, 224,942 messages). The hypotheses further divided the remaining messages into streamer-targeted and other-targeted messages prior to dividing it for two stream content comparisons: gameplay versus non-gameplay and entertainment versus expertise. The proposed research questions then examined just the streamer-targeted messages from gameplay streams for their attempts to engage with gameplay and gamespace.

Research Hypotheses

The research hypotheses examined targeted messages for their verbal immediacy. Verbal immediacy is a strong predictor for perceived closeness and, therefore, a predictor of parasocial relationships. A paired-sample t-test analyzed the variance between streamer-targeted and other-targeted messages from all channels. Mixed-model ANOVAs continued the analysis of variances between message target and stream content.

Research Hypothesis One

The first stage of hypothesis testing was meant for establishing a baseline understanding of verbal immediacy variance on Twitch, regardless of Twitch content, between streamer-

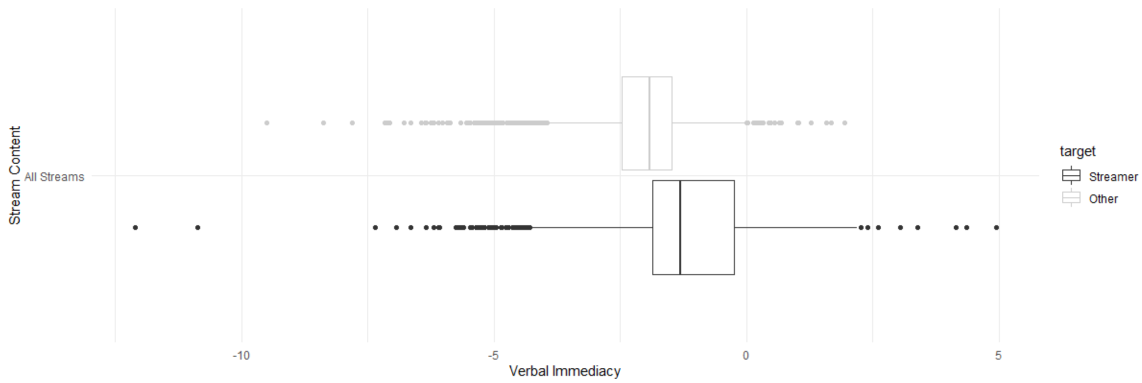

Figure 7. Streamer-Targeted vs. Other-Targeted Messages from All Channels

targeted and other-targeted messages (see Figure 7). Message variance was tested by defining verbal immediacy scores through the arithmetic score of LIWC scores for each of the previously designated message groups for each channel. Results were tested using a paired samples t-test, and the effect size was calculated using Cohen's *d*.

RH 1: *Streamer-targeted messages* from the chats from *all streams* will have a greater verbal immediacy score than *viewer-targeted messages* from the chats of *all streams*.

Analysis found that streamer-targeted messages from all chats (M = -1.13, SD = 1.24) produced a greater verbal immediacy score than viewer-targeted messages (M = -2.06, SD = .93), $t(4763) = -49.74$, $p < .001$ (see Tables 4, 5). The analysis has a large effect ($p <.001$, d = .85). Effect size is especially significant given the quantity of data collected often over producing statistically significant results.

Table 4

Hypothesis One Descriptive Statistics

	Mean	N	Std. Deviation	Std. Error Mean
Other	-2.0581	4764	.93456	.01354
Streamer	-1.1298	4764	1.23698	.01792

Table 5

Hypothesis One Paired Samples t-Test

	Paired Differences								
				95% CI of the Difference					
	Mean	Std. Deviation	Std. Error Mean	Lower	Upper	t	df	*p*	*d*
Other - Streamer	-.928	1.288	.019	-.965	-.892	-49.738	4763	.000	.847

Research Hypothesis Two

The second hypothesis continued examining the verbal immediacy scores of Twitch chat messages. Hypothesis two divided the all chat category between game streams (e.g., *Hearthstone* and *League of Legends*) and nongame streams (e.g., Just Chatting and Music & Performing Arts) while still dividing messages between streamer-targeted messages and viewer-targeted messages (see Figure 8). A mixed-model ANOVA was used to test for these relationships.

RH 2a: *Streamer-targeted messages* from the chats from *nongame streams* and *game streams* will have a greater verbal immediacy score than *viewer-targeted messages*.

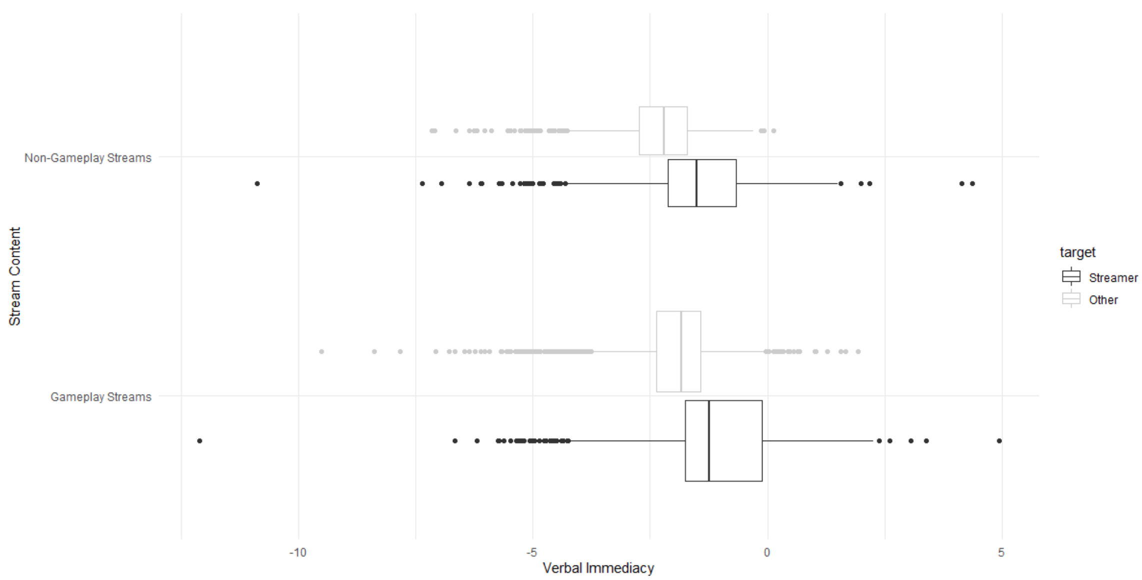

Figure 8. Streamer-Targeted vs. Other-Targeted Messages

from Non-gameplay vs. Gameplay Streams

Table 6

Hypothesis Two Descriptive Statistics

	Type	Mean	Std. Deviation	N
Viewer	Game	-1.9689	.91732	3542
	Nongame	-2.3165	.93638	1222
	Total	-2.0581	.93456	4764
Streamer	Game	-1.0321	1.19350	3542
	Nongame	-1.4128	1.31530	1222
	Total	-1.1298	1.23698	4764

The mixed-model ANOVA first examined the within-subject condition of message target which separates streamer-targeted and viewer-targeted messages. The test of within-subjects contrasts found a statistically significant difference between message targets [F (1, 4762) = 1854.44, p < .001] with a large effect size (p < .001, η^2_p = .28; see Table 7). This finding supports RH 2a which posited that streamer-targeted messages (M = -1.13, SD = 1.24) would produce a greater verbal immediacy score than viewer-targeted messages (M = -2.06, SD = .93; see Table 6).

RH 2b: *Streamer-targeted messages* and *viewer-targeted messages* from the chats from *nongame streams* will have a greater verbal immediacy score than from *game streams*.

A mixed-model ANOVA also allows for an additional test for between-subjects effects. The between-subjects effect for the second hypothesis is the stream type variable contrasting nongame streams (M = -1.87, SE = .03) and game streams (M = -1.50, SE = .02). The tests of between-subjects effects found a statistically significant difference for stream types [F (1, 4762) = 158.17, p <.001] with a small effect size (p < .001, η^2_p = .032; see Table 8); however, the relationship is inverted from the one posited by the hypothesis.

Table 7

Hypothesis Two Tests of Within-Subjects Contrasts

Source	Sum of Squares	df	Mean Square	F	p	η^2_p
Target	1538.880	1	1538.880	1854.440	.000	.280
Target * Type	.496	1	.496	.597	.440	.000
Error(Target)	3951.677	4762	.830			

Table 8

Hypothesis Two Tests of Between-Subjects Effects

Source	Sum of Squares	df	Mean Square	F	*p*	η^2_p
(Intercept)	20577.318	1	20577.318	13506.652	.000	.739
Type	240.963	1	240.963	158.165	.000	.032
Error	7254.884	4762	1.523			

Research Hypothesis Three

The third hypothesis further divided Twitch chat messages. Hypotheses three removed the messages from nongame streams and divided the game streams messages according to the streamer specializations (i.e., expertise and entertainment). Message target remained (see Figure 9). A mixed-model was again used to test for these relationships.

RH 3a: *Streamer-targeted messages* from the chats from *entertainment streams* and *expertise streams* will have a greater verbal immediacy score than *viewer-targeted messages*.

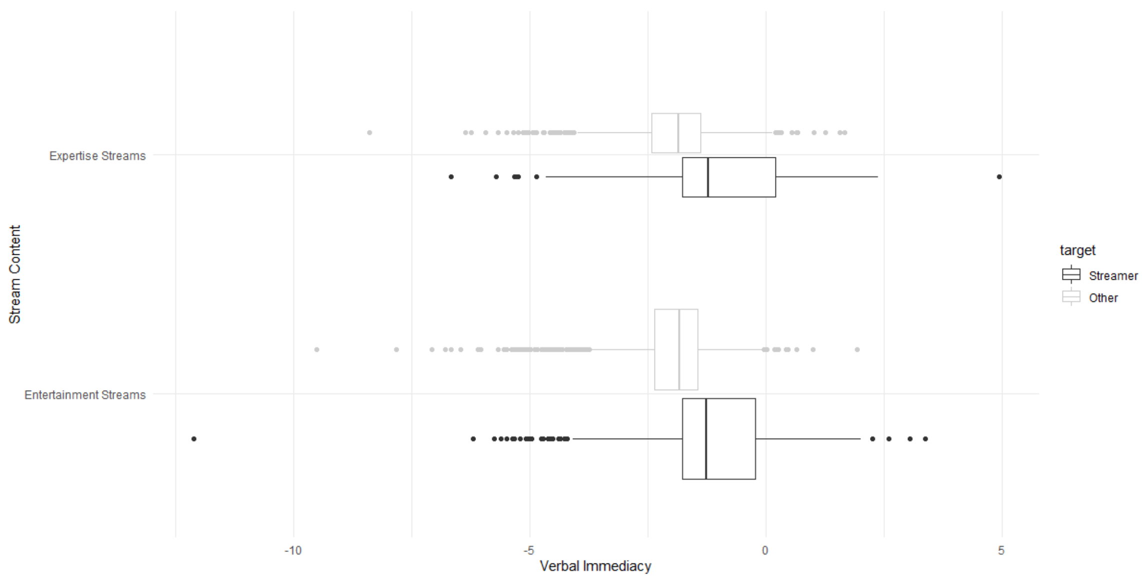

Figure 9. Streamer-Targeted vs. Other-Targeted Messages

from Entertainment vs. Expertise Streams

Table 9

Hypothesis Three Descriptive Statistics

	Type	Mean	Std. Deviation	N
Viewer	Entertainment	-1.9633	.86535	2859
	Expertise	-1.9922	1.10919	683
	Total	-1.9689	.91732	3542
Streamer	Entertainment	-1.0525	1.16780	2859
	Expertise	-.9467	1.29297	683
	Total	-1.0321	1.19350	3542

Another mixed-model ANOVA was used to examine the third research hypothesis. The tests of within-subjects contrasts found a statistically significant difference between message targets [F (1, 3540) = 1293.67, $p < .001$] with large effect size ($p < .001$, $\eta^2_p = .268$; see Table 10). This finding supports RH 3a which posited that streamer-targeted messages ($M = -1.03$, $SD = 1.19$) would produce a greater verbal immediacy score than viewer-targeted messages ($M = -1.97$, $SD = .92$; see Table 9).

> RH 3b: *Streamer-targeted messages* and *viewer-targeted messages* from the chats from *entertainment streams* will have a greater verbal immediacy score than from *expertise streams*.

The secondary test of the mixed-model ANOVA, the test of between-subjects effects, was used to examine the second part of the third research hypothesis. In comparing the entertainment streams ($M = -1.51$, $SE = .02$) and expertise streams ($M = -1.47$, $SE = .02$), the tests of between-subjects effects failed to find a statistically significant difference for stream types [F (1, 3540) =

Table 10

Hypothesis Three Tests of Within-Subjects Contrasts

Source	Sum of Squares	df	Mean Square	F	*p*	η^2_p
Target	1054.884	1	1054.884	1293.674	.000	.268
Target * Type	4.993	1	4.993	6.123	.013	.002
Error(Target)	2886.578	3540	.815			

Table 11

Hypothesis Three Tests of Between-Subjects Effects

Source	Sum of Squares	df	Mean Square	F	*p*	η^2_p
(Intercept)	9774.178	1	9774.178	6744.202	.000	.656
Type	1.632	1	1.632	1.126	.289	.000
Error	5130.420	3540	1.449			

1.13, $p = .289$; see Table 11]. The lack of statistical significance suggests no difference between verbal immediacy scores for the chats from entertainment streams and expertise streams.

Research Questions

The proposed research questions continue the analysis of affective messaging by examining the Twitch chat messages for intentionality. Game engagement is the intentionality of interest for this dissertation. Theoretically, game engagement can occur at both the gameplay and gamespace level. Attempted gameplay engagement is the messages from viewers in which they intended to discuss the game as opposed to other content. Chat messages are restricted to streamer-specific messages as they explicitly identify the streamer within the message.

RQ 1: Are *streamer-specific messages* for gameplay streams attempting to engage with the gameplay?

Gamespace engagement concerns a far more limited messaging. For chat to be included within gamespace entirely because of their messages, the messages must assume a traditional gamespace role. Viewers, when they specifically identify the streamer in their message, partially meet this condition immediately because Twitch will highlight the message in the chat for the person that the chatter is targeting. This process is like the player-to-character interactions from the Player Interaction Framework (Wainess, Kerr, & Koenig, 2011), which is characterized by the highlighting of a character is of importance. The elevation of a viewer to a gamespace agent is by identifying instances in which they fulfill the role of either a presentation or background object.

RQ 2: Do these messages elevate the senders to inclusion within gamespace?

This dissertation took a mixed-method approach to identify and code relevant messages. Because the dissertation is more concerned with how gameplay and gamespace engagement is occurring, instead of how much, n-grams were identified and then manually coded. The n-gram size for this dissertation was five (quintgrams). Grounded theory methods identified two categories of relevance: game questions and game suggestions. Game questions are viewers attempting to convert players into presentation objects while game suggestions are viewers attempting to elevate themselves to the role of a presentation object.

Streamers as Objects

The first example of gameplay engagement occurs when viewers attempt to acquire specific information from the streamer. These interactions generally occur as either implicit or explicit questions. Viewer questions can include questions about the game, what the streamer is doing, or what the viewer should do. Each question assists in defining the player within the gamespace.

After Blizzard announced that one of its primary characters is gay, sonicspeed5 wrote, "@redshell what do you think about making s76 gay, they should have made doomfist gay :/" in the chat for Redshell. Viewers can also take advantage of chat to find further information about in-game changes. In the chat for the channel Gorgc, cthetrees wrote, "@Gorgc what do you think of the dusa huskar matchup? shouldn't dusa win it now because huskar has no magic resist?".

Viewers are not limited to asking questions about the current state of the game as they can also ask for some input on possible future changes. itsaBella posed a potential change to

Geistra: “@Geistra i think it would be cool if the prestige blood was a layer you could wear over any outfit.” Another suggested change was offered to TeftyTeft by gorthanlande1 when they wrote, “@TeftyTeft Ideally what would you like to see out of a skill tree in Destiny.” Le_VIP took a more cosmetic approach to proposed changes when they wrote in the chat for the channel dasMEHDI “they should make a plate and put it inside the galleon with their name on it @dasMEHDI.”

Viewers as Objects

Along with elevating and altering the roles of the streamers within gamespace, viewers are also able to position themselves as something akin to traditional gamespace objects. The most obvious method for this occurring in Twitch chat is through viewers offering information or suggestions to the streamer. Gamespace inclusion occurs most frequently through viewers serving as additional tutorial systems for the streamers. Viewers are also able to provide unique information or suggest different play styles.

Most gamespace engagement from viewers comes from tutorial-like statements. TheKoztos wrote to michaelalfox, “@michaelalfox you need to make a 3x3x3 cub of coke brick, the center block needs to be empty”. Similarly, Cru3Lmotiv wrote to Mathil1, “@Mathil1 I think you need to update you loot filter”, while they were streaming Path of Exile. While also streaming Path of Exile, cooxi messaged Quin69 stating, “@Quin69 you have like 50ex of gear, except no leech/ref you should be able to do every single map in this game (expect grandmaster maybe)”. Prior to offering tutorial-like information, viewers can offer streamers the choice to receive the new information. Sanfist_Mage offered such an opportunity when they wrote, “@Aurora_Peachy do you want me to tell you about night mode or do you prefer to experience it your self?” to Aurora_Peachy. More unique approaches also exist. Streamers can turn to the chat for updated information about game bugs. WINTERFR0STY attempted to inform BurkeBlack when they shared, “@BurkeBlack it might make you update your bnet settings but you should be able to.”

DISCUSSION

New media sites, such as Twitch, continually provide media figures and users the opportunity to communicate openly. Streamers can communicate with their viewers through their streams as well as through Twitch chats. Viewers can communicate in the Twitch chat with streamers and other viewers. This dissertation is particularly interested in the user-figure relationship from the perspective of the media user. Namely, the proposed research hypotheses and questions examined the implications of media user communicative tendencies. The research hypotheses analyzed the variation of chat messages, and the research questions analyzed the significance of chat messages regarding gameplay and gamespace.

Research Hypotheses

Natural language processing produced two broad categories of findings. First, streamer-targeted messages consistently produced greater verbal immediacy than other-targeted messages, regardless of stream content. Second, stream content is a generally poor determiner of verbal immediacy. These findings provide a foundation for the proposed research questions and potential directions for future research.

When comparing the variance of verbal immediacy expressed within streamer-targeted and other-targeted messages, verbal immediacy was consistently greater for streamer-targeted messages than other-targeted messages. This finding supports the presence of parasocial relationships (PSRs). PSR is the perceived closeness developed by media users that is comparable to that of other friends and peers. By perceived a closer relationship, viewers are more comfortable communicating with streamers as they would with their friends.

Unlike message specificity, stream content proved to be less of a determiner of variance. When comparing gameplay and non-gameplay streams, the mixed-model ANOVA identified an inverted relationship between content and verbal immediacy than posited by the second research hypothesis. Namely, there was a greater verbal immediacy in messages from gameplay streams than from non-gameplay streams. The effect size of this variance was small, but this does suggest that Twitch, as a platform, provides users sufficient tools, such as time, to overcome the barriers created by gameplay. As argued by most cues-filtered-out theories of computer-mediated communication (CMC), media users often overcome the barriers created by mediation over time (see Walther & Parks, 2002). With streamers regularly spending more than 40 hours a week streaming, the type of content becomes less critical in determining the developed dispositions and relationships. Since Twitch is a platform predominantly built for gamers, it is possible that the motivations for viewers may overshadow the inherent limitations of the content. The comparison of the two gameplay categories – expertise and entertainment - proved no statistical significance. The lack of significant variance further supports the argument that viewers, due to their motivations and allotted time, are able to overcome the inherent limitations of the content. The significance of the variance between nongame and game streams further supports this as the relationships between streamer and viewer will remain mostly unchanged, but the viewers will recognize nongame streams as opportunities to engage with the streamers.

Research Questions

The discovery of greater verbal immediacy for streamer-targeted messages than for other-targeted messages provides insight into the perception viewers hold toward the streamers they watch. The increased immediacy suggests that viewers have a great enough perceived closeness that they would be comfortable discussing various topics with streamers as they would with more traditional relationships. The synchronous nature of chat also allows users to discuss any unbanned topic they wish, including directly affecting the streaming content. For Twitch, most of the streamed content is gameplay. From this position, the research questions examine the methods in which viewers attempt to engage with gameplay.

Though most messages do not involve gameplay, gameplay-related messages are still commonly occurring. Among these messages, viewers often attempt to engage with the gameplay through the streamer. The analyses found that viewers take two approaches to gameplay engagement. First, viewers will attempt to redefine the role of the streamer within gamespace by asking them questions. By answering questions, streamers begin to take a role commonly limited to non-player characters (NPC). Second, viewers will attempt to elevate themselves to the NPC role. Instead of inviting streamers to provide information to them, viewers will take on the NPC role by providing streamers the information that is traditionally done by in-game characters.

Though streamers are naturally introducing themselves into the gamespace (see Newman, 2002a, 2002b), viewers provide streamers an opportunity to redefine themselves within gamespace as they pose questions through chat. In these instances, streamers can momentarily cease being the primary human agents in games and, instead, serve as presentation objects for the viewers. This dual nature provides a unique position for gameplay streamers that would nominally occur in traditional gameplay settings. The closest example would be players in an arcade (Taylor, 2012). Arcade players are not expected to interact with non-players extensively they do not know; whereas, streamers are generally expected to interact with their viewers. Despite viewers engaging with gameplay in these instances, these actions would not justify defining viewers as human agents within gamespace.

Viewers are attempting to elevate themselves into human agents of gamespace when they provide information to the streamers. Though streamers ultimately have control over what role viewers have within gamespace, analysis suggests that these viewers are actively attempting to affect gameplay. Even if streamers are not actively reading or taking suggestions from the chat, viewers can still serve as background objects as long as the streamers are not entirely removing chat from their streaming environment. Most of the examples of gameplay engagement from this category involved general information traditional gamers would get from in-game tutorials, NPCs, or player manuals. Viewers would disseminate information through either making a statement or reshaping a statement into a question. By streaming their gameplay, streamers can use the collective knowledge of Twitch chat to address bugs or glitches instead of leaving the game and searching the internet to find a solution.

Based upon the quintgrams, messages from entertainment and expertise streams appeared to differ in their preferred method for gameplay engagement in a predictable manner. Expertise streamers are streamers with uniquely advanced skills in or knowledge of the game they are playing. Viewers, presumably recognizing the knowledge and skills difference, produce myriad gameplay questions and few gameplay suggestions. Chat messages from entertainment streamers, which are predominantly identified by their lack of expertise, were the primary source for gameplay suggestions. Though further research is necessary to identify how significant these communicative differences are, it does suggest that individuals who are interested in becoming more integral elements of the streamer's gamespace are more likely to interact in the chat of entertainment streamers while individuals who are interested in learning more specialized knowledge of a game will choose expertise streams. It is necessary, however, to recognize that gameplay engagement styles are not mutually exclusive since entertainment streamers can still provide unique information, despite expertise not being their specialty, and expertise streamers can still receive suggestions.

Another type of gameplay engagement is of the more troll variety. Instead of providing accurate information, viewers will occasionally provide information that is either incorrect or counterproductive. A famous example of trolling Twitch streamers occurred in July of 2018 when a user introduced a copypasta in the stream of Ninja, the most popular streamer at the time, when he wrote, “What the fuck I though you died from ligma” (Alexander, 2018c). “Ligma” is the set up for a crude joke, which prompts the streamer to ask, “what’s ligma?” Viewers would then reply, “ligma balls.” Gameplay related troll suggestions can get players to do things that poorly affect their gameplay, such as viewers suggesting that girlwithyellowspoon (2018), who was playing *Minecraft*, throw her pickaxes into the lava. Another common troll suggestion is for viewers to suggest Alt+F4, a key combination that will force quit a program in Windows operating systems, as a solution to in-game problems (FortyOne, 2019).

Limitations

Though n-grams are an efficient approach to identifying common messaging, Twitch chat provides a unique set of issues that will require special attention if applied to future research. Streamers regularly use bots to assist with commonly occurring chat needs, such as banning specific words (e.g., racial slurs) or hyperlinks. Many bots have built-in functions that allow them to respond to command prompts in chat, such as how long the streamer has been live or what song the streamer is playing. Because the bots produce nearly identical responses to each command, the n-grams from these messages repeat regularly. It would be easy to remove the most common Twitch bots from the dataset (e.g., nightbot, streamlabs); however, these bots are regularly forked – the code is altered into a new bot to meet the needs of the user and given a new name – and, therefore, difficult to identify and remove en masse. When new memes or copypastas arise on Twitch, viewers are likely to duplicate them continually with little to no alterations. These messages from viewers can occur on a channel, on a collection of channels, or across the entire platform. Both types of messaging affect the effectiveness of n-grams negatively and must be compensated through manual coding.

A minor technical limitation that arose throughout the completion of this dissertation is Twitch upgrading to a new API system and removal of some functionality of the previous API. With the introduction of the new Twitch API, changes to the infrastructure that break elements of the previous API are either removed or not fixed. The bots for this dissertation were coded to only include partnered streamers with English-speaking chats but, because of errors, this information was no longer accessible through an API request. This dissertation fed the full list of logged channels through the new API to receive streamer status and remove all non-partners from the logs, though retained them in the original database. Because data analyses made API calls following partnered channels changing their names, it was necessary for coding to remove such channels as it was impractical to identify partnership status following the name changes. Manual coding identified English-speaking channels during the entertainment/expertise coding process. To code for an English-speaking chat, past VODs were loaded and briefly scanned to see if the streamer and chat were communicating in English. A chat did not need to be entirely English-speaking to be included in the analysis, just predominantly English-speaking. The language restriction is to limit the issues that can arise when attempting to use machine learning tools for unintended languages.

Differentiating entertainment and expertise also provided issues that limited the dissertation. Because the line of demarcation between entertainment and expertise are fuzzy, coding erred toward coding entertainment in the negative. Coding entertainment in the negative means that it included any channel which did not clearly identify itself as an expert through their stream, channel, or social media. A more accurate approach to identifying the differences between the two types of streams would be to identify a smaller subset of channels that heavily identified toward one of the extremes.

CONCLUSION

Twitch represents a growing subset of media in which media figures have nearly complete control over what content they create and when they want to broadcast. Gameplay live streamers are especially unique since they are repurposing third-party media (i.e., games) into something they perform live to spectators who can communicate directly with the streamers. This developing media ecosystem provides insight into both how the media landscape has changed and what media may look like in the future. A series of bots logged tens of thousands of total channels from over 25 stream categories for 30 days to provide a complete perspective on Twitch chat messaging. Data cleaning then reduced the logs to only partnered streamers with English-speaking chats. Chat logs were then divided based on target specificity (i.e., streamer-specific and non-streamer specific messages) and stream content (i.e., game, nongame, entertainment, expertise).

The research hypotheses and questions addressed two separate, broader concepts that build upon each other. The research hypotheses examined how verbal immediacy varied among different sets of Twitch chat messages, and the research questions examined how viewers are attempting to engage with and affect the streamed gameplay. The data supported hypothesis one found, that streamer-targeted messages produced greater verbal immediacy than other-targeted messages. The data also supported hypotheses two and three, that stramer-targeted messages produced greater verbal immediacy than other-targeted messages. Stream content type provided little significant impact, if any, to message content. The variance between streamer-targeted messages from entertainment and expertise was not statistically significant. Therefore, the intended recipient of the chat messages is a better predictor for verbal immediacy than stream content. That is not to say, however, that stream content is entirely insignificant.

Future Research

Beyond its current findings, this dissertation presents the data and framework necessary for examining a range of other issues. For example, despite its growing diversity, Twitch can still be broadly defined as a site populated by straight, white, cis-gendered, young adult men. Therefore, Twitch chat logs can prove an excellent tool for examining how a mostly homogeneous group discusses out-groups. Common out-groups would include people of color, women, LGBTQ+ individuals, or those with mental health issues.

Twitch chat has a lurid history of racially motivated content and language. Global emotes, which are emotes available to every user, have several that are people's faces. These emotes are often used to represent that entire race. Trihard, an emote using the face of a famous speedrunner called Trihex, is used as a substitute for making jokes about black men. Similarly, MingLee references the appearance of Asians, and ANELE references the appearance of Middle-Easterners. This tactic is so prolific that viewers quickly adopt new global emotes into this routine. For example, KFC and Snickers ran ad campaigns on Twitch that included emotes. KFC had an emote of a bucket of fried chicken, and Snickers had an emote with a Snicker's bar. These emotes, as predicted by some streamers, were quickly co-opted as racial slurs toward black men and women (Alexander, 2018b). Twitch has attempted to correct for this tendency by banning streamers when they use derogatory terms. The attempts to correct for this issue is strict enough that an Iranian-born *League of Legends* streamer was banned when, presumably due to his accent, viewers and Twitch staff mistakenly heard "n****r" when he actually said "idiot." His ban, which was initially 30 days, was first lowered to 7 days and then completed lifted after several reviews. This dissertation provides a set of tools to identify and analyze how individuals discuss race and how it affects the larger conversation.

Women are a rapidly growing demographic for Twitch in both their non-gameplay and gameplay sections. Despite their increased presence, women are also regularly attacked online through their chat or their social media (e.g., Instagram or Twitter). Much of their poor treatment occurs at the hands of trolls, individuals who are intentionally abrasive or offensive online in order to provoke a response. One method of attack is through mass reporting them, or "tattle," to get them suspended (Katzowitz, 2019). Mass reporting is not the only possible outcome for women on Twitch. Several women have also reported death threats and being doxxed – having private information spread widely through web or print. Adult actresses, a growing subset of women streamers, are especially targeted on the platform given their celebrity or past exposure (Snow, 2019). By analyzing the variance between the chats of men and women, future research would be able to identify exact variations between the two groups of streamers and then draw further theoretical implications that can produce more equitable environments.

Despite the LGBTQ+ community having a constant presence throughout its history, Twitch has not always been great at protecting its members. Like its problems with racial slurs, Twitch streamers often use homophobic slurs and, in turn, get suspensions. For example, Deadmau5, an EDM DJ, was suspended in February of 2019 for spouting a slur against an opponent while streaming. Racial and homophobic terms are so commonly used and discussed across the platform that streamers began colloquially referring to these words as "gamer words" (Martin, 2019). Twitch even has a global emote (KappaPride) that serves an explicitly "gay" companion to another global emote (Kappa). Both emotes serve to mean that something is a joke. The problem with KappaPride is that it can be used for both in-group and out-group jokes. Out-group jokes can often come at the expense of the LGBTQ+ community. Natural language processing allows for the examination of how Twitch chat communicates LGBTQ+ issues.

Mental health is a final out-group issue that can lead to destructive online communication. Topics like autism and depression commonly arise from Twitch streamers and their chat. Many Twitch streamers use terms like autism and autistic as slurs against neurotypical others. Though these terms can lead to a streamer's suspension (Asarch, 2019), they are still regularly used. Because of its regular use, textual analysis could produce a wide range of information about how mental health is freely discussed on the platform. The topic of mental health has become more acceptable as YouTube and Twitch content creators began opening up about their burnout (Alexander, 2018a).

Besides communications about out-groups, Twitch is also largely defined by its cultural construction. Memes, though not unique to Twitch, are a considerable part of its culture. Russian troll farms weaponized memes to influence the 2016 presidential elections (Thompson & Lapowsky, 2018). What makes Twitch unique is the ever-changing landscape of sub-communities on the platform. Viewers can follow hundreds of channels across the platform. Some memes are developed within a single channel and never spread to other channels, while others spread across thousands of channels. A preliminary review of chat logs suggests that most memes that spread past a single channel are likely to be contained with a sub-community. Sub-communities can develop from several premises. Many viewers that watch more than one channel can still limit their viewing to a single game or game genre. Fans of a single streamer may also choose to watch streamers with whom their favorite streamer interacts. This approach to community building is how esports teams build up lesser-known players on their teams and how groups like Offline TV or The Derp Crew. A meme that gains traction in one channel will likely spread throughout the remainder of the sub-community.

APPENDIX

Dissertation Logger: `settings.py.example`

```
IRC = {
    'SERVER': 'irc.chat.twitch.tv',
    'NICK': 'USERNAME',
    'PASSWORD': 'oauth:YOUR_KEY_HERE',
    'PORT': 6667,
}

# If you're using a local database the host is 'localhost'
DATABASE = {
    'NAME': 'YOUR_DATABASE',
    'USER': 'YOUR_USERNAME',
    'PASSWORD': 'YOUR_PASSWORD',
    'HOST': 'localhost',
}

API = {
    'CLIENTID': 'YOUR_CLIENT_ID'
```

Dissertation Logger/CATEGORY: `bot.py`

```
import settings
import threading
import Queue
import socket
import logging

from irc import IRCBot, run_bot, DisconnectedException


class TwitchBot(IRCBot, threading.Thread):
    """
    A threaded IRC bot that automatically reconnects and rejoins channels if
disconnected. Communication happens over the queue command_queue. The
accepted commands through the queue are 'join' and 'part'.
    """
    MESSAGES_TO_IGNORE = ['/join', '/part']

    def __init__(self, name, conn, chat_logger, command_queue,
log_filename=None, *args, **kwargs):
        super(TwitchBot, self).__init__(conn, *args, **kwargs)

        self.name = name
        self.chat_logger = chat_logger
        self.command_queue = command_queue
        self.disconnect = threading.Event()

        self.logger = self.conn.get_logger(name, log_filename)

    def run(self):
        """
        Receives data in a loop until the event for disconnecting is set.
Checks the command queue for actions to be taken (joining or leaving
channels).
        """
        patterns = self.conn.dispatch_patterns()
```

```
        while not self.disconnect.is_set():
            try:
                data = self.conn.get_data()  # returns empty string if times
out
                if data:
                    self.conn.dispatch_data(data, patterns)

                command = self.command_queue.get_nowait()
                self.process_command(command)
            except DisconnectedException:
                self.logger.info('Disconnected from server. Reconnecting.')
                self.conn.close()
                self.connect_and_join_channels(self.channels)
                continue
            except Queue.Empty:
                continue

    def join(self, timeout=None):
        """
        Forces the bot to disconnect, close its socket and database
        connection.
        """
        self.conn.close()
        self.chat_logger.close()
        self.logger.info('Closed connection.')
        self.disconnect.set()
        super(TwitchBot, self).join(timeout)

    def connect_and_join_channels(self, channels):
        if not self.conn.connect():
            raise RuntimeError("Failed to connect to IRC channel.")

        for channel in channels:
            self.conn.join(channel)
        self.channels = channels

    def process_command(self, command):
        """
        Processes a command, either joining or leaving channels.
        command is expected to be a tuple of a string and a list.
        Valid commands are 'join' and 'part'.
        """
        if not (type(command) is tuple and len(command) == 2):
            raise ValueError("Expected command to be a tuple of a string and
a list")

        action, channels = command
        self.logger.info("Received command %s (%s)" % (action,
','.join(channels)))
        if action == 'join':
            for channel in channels:
                self.conn.join(channel)
            self.channels += channels
        elif action == 'part':
            for channel in channels:
                self.conn.part(channel)
            self.channels = [c for c in self.channels if c not in channels]
```

```
    def log(self, sender, message, channel):
        if sender == settings.IRC['NICK']:
            self.logger.info("%s, %s: %s " % (channel, sender, message))
            return
        if message in self.MESSAGES_TO_IGNORE:
            return

        self.chat_logger.log_chat(sender, message, channel)

    def command_patterns(self):
        return (
            ('.*', self.log),
        )
```

Dissertation Logger/CATEGORY: `create_tables.sql`

```sql
--
-- PostgreSQL database dump
--

SET statement_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
SET check_function_bodies = false;
SET client_min_messages = warning;

SET search_path = public, pg_catalog;

SET default_tablespace = '';

SET default_with_oids = false;

--
-- Name: chat_log; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
--

CREATE TABLE chat_log (
    id integer NOT NULL,
    channel character varying(64) NOT NULL,
    sender character varying(64) NOT NULL,
    message character varying(512) NOT NULL,
    date bigint NOT NULL
);


--
-- Name: chat_log_id_seq; Type: SEQUENCE; Schema: public; Owner: postgres
--

CREATE SEQUENCE chat_log_id_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;
```

```
--
-- Name: chat_log_id_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner:
postgres
--

ALTER SEQUENCE chat_log_id_seq OWNED BY chat_log.id;


--
-- Name: stream_log; Type: TABLE; Schema: public; Owner: postgres;
Tablespace:
--

CREATE TABLE stream_log (
    id integer NOT NULL,
    channel character varying(64) NOT NULL,
    title character varying(128),
    game character varying(64),
    viewers integer NOT NULL,
    date bigint NOT NULL
);


--
-- Name: stream_log_id_seq; Type: SEQUENCE; Schema: public; Owner: postgres
--

CREATE SEQUENCE stream_log_id_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;

--
-- Name: stream_log_id_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner:
postgres
--

ALTER SEQUENCE stream_log_id_seq OWNED BY stream_log.id;


--
-- Name: id; Type: DEFAULT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY chat_log ALTER COLUMN id SET DEFAULT
nextval('chat_log_id_seq'::regclass);


--
-- Name: id; Type: DEFAULT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY stream_log ALTER COLUMN id SET DEFAULT
nextval('stream_log_id_seq'::regclass);
```

```
--
-- Name: chat_log_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres;
Tablespace:
--

ALTER TABLE ONLY chat_log
    ADD CONSTRAINT chat_log_pkey PRIMARY KEY (id);

--
-- Name: stream_log_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres;
Tablespace:
--

ALTER TABLE ONLY stream_log
    ADD CONSTRAINT stream_log_pkey PRIMARY KEY (id);

CREATE INDEX chat_log_date_idx ON chat_log (date);
CREATE INDEX chat_log_channel_idx ON chat_log (channel);

--
-- PostgreSQL database dump complete
--
```

Dissertation Logger/CATEGORY: `db_logger.py`

```
import psycopg2

from utils import current_time_in_milli


class DatabaseLogger:
    conn = None
    cursor = None

    def __init__(self, host, name, user, password):
        self.conn = psycopg2.connect(host=host, dbname=name, user=user,
password=password)
        self.conn.autocommit = True
        self.cursor = self.conn.cursor()

    def close(self):
        self.cursor.close()
        self.conn.close()

    def log_chat(self, sender, message, channel):
        if len(message) > 512:
            message = message[:512]

        if self.cursor.closed:
            return

        try:
            self.cursor.execute("INSERT INTO CATEGORY_chat_log (sender,
message, channel, date) VALUES (%s, %s, %s, %s)",
                                (sender, message, channel,
current_time_in_milli()))
```

```
        except psycopg2.DataError as e:
            print e
            print message

    def log_stream_stats(self, stream):
        if 'status' not in stream['channel']:
            stream['channel']['status'] = None
        elif stream['channel']['status'] and
len(stream['channel']['status']) > 128:
            stream['channel']['status'] = stream['channel']['status'][:128]
        if 'game' not in stream['channel']:
            stream['channel']['game'] = None

        if self.cursor.closed:
            return

        self.cursor.execute("INSERT INTO stream_log (channel, title, game,
viewers, date) "
                            "VALUES (%s, %s, %s, %s, %s)",
                            (stream['channel']['name'],
                             stream['channel']['status'],
                             stream['channel']['game'],
                             int(stream['viewers']),
                             current_time_in_milli()))
```

Dissertation Logger/CATEGORY: `main.py`

```
import argparse
import sys
from manager import TwitchManager


def main():
    parser = argparse.ArgumentParser()
    parser.add_argument("-n", "--streams-to-log", dest="channels_amount",
type=int,
                        help="the number of streams to log", default=50)
    parser.add_argument("-f", "--log-filename", dest="log_filename",
                        help="the filename to log to", default=None)
    parser.add_argument("-c", "--channels", dest="channels", type=str,
nargs='+',
                        help="the specific channel names to log",
default=[])
    args = parser.parse_args()

    manager = TwitchManager(channels_amount=args.channels_amount,
channels=args.channels, log_filename=args.log_filename)
    try:
        manager.run_log_loop()
    except KeyboardInterrupt:
        print 'Exiting gracefully...'
        manager.stop_bot()

if __name__ == "__main__":
    main()
```

Dissertation Logger/CATEGORY: `manager.py`

```python
import Queue
import settings
import time

from irc import run_bot, IRCConnection
from bot import TwitchBot
from utils import get_top_streams, get_channel_names
from db_logger import DatabaseLogger


class TwitchManager:
    """
    Manages a series of irc chat bots that log the most popular channels on
    Twitch.TV. The list of most popular channels is updated every 60
seconds.
    """
    CHANNELS_PER_BOT = 25
    SECONDS_BETWEEN_UPDATE_STREAMS = 120
    SECONDS_BETWEEN_CREATE_BOTS = 30

    def __init__(self, channels_amount, channels, log_filename=None):
        self.bots = []
        self.channels_amount = channels_amount
        self.log_filename = log_filename
        self.channels = channels
        self.db_logger = DatabaseLogger(settings.DATABASE['HOST'],
                                        settings.DATABASE['NAME'],
                                        settings.DATABASE['USER'],
                                        settings.DATABASE['PASSWORD'])

    def _create_bot(self, name, channels):
        conn = IRCConnection(settings.IRC['SERVER'],
                             settings.IRC['PORT'],
                             settings.IRC['NICK'],
                             settings.IRC['PASSWORD'],
                             self.log_filename)
        bot_db_logger = DatabaseLogger(settings.DATABASE['HOST'],
                                       settings.DATABASE['NAME'],
                                       settings.DATABASE['USER'],
                                       settings.DATABASE['PASSWORD'])
        bot = TwitchBot(name, conn, bot_db_logger, Queue.Queue(),
self.log_filename)
        bot.daemon = True
        bot.connect_and_join_channels(channels)
        bot.start()
        return bot

    def _create_bots(self, channels):
        """
        Creates bots to log the given list of channels.

        Twitch limits how many channels can be joined per connection, so we
        create just enough bots to log all the desired streams. There's also
        a timeout between opening connections otherwise Twitch disconnects
        all bots.
        """
        channels_joined = 0
```

```
        while channels_joined < len(channels):
            self.bots.append(
                self._create_bot('Bot %i' % len(self.bots),
                                 channels[channels_joined:channels_joined +
self.CHANNELS_PER_BOT]))
            channels_joined += self.CHANNELS_PER_BOT
            time.sleep(self.SECONDS_BETWEEN_CREATE_BOTS)

    def _update_bot_channels(self, bot, new_channels):
        channels_to_remove = list(set(bot.channels) - set(new_channels))
        channels_to_add = list(set(new_channels) - set(bot.channels))

        if channels_to_remove:
            bot.command_queue.put(('part', channels_to_remove))

        if channels_to_add:
            bot.command_queue.put(('join', channels_to_add))

    def _log_streams(self, streams):
        for stream in streams:
            self.db_logger.log_stream_stats(stream)

    def _run_popular_streams_loop(self):
        streams = get_top_streams(self.channels_amount)
        channels = get_channel_names(streams)
        self._create_bots(channels)
        self._log_streams(streams)

        while True:
            time.sleep(self.SECONDS_BETWEEN_UPDATE_STREAMS)
            streams = get_top_streams(self.channels_amount)
            channels = get_channel_names(streams)

            i, channels_joined = 0, 0
            while channels_joined < self.channels_amount:
                self._update_bot_channels(self.bots[i],

channels[channels_joined:channels_joined + self.CHANNELS_PER_BOT])
                i += 1
                channels_joined += self.CHANNELS_PER_BOT

            self._log_streams(streams)

    def _run_static_streams_loop(self):
        self._create_bots(self.channels)
        while True:
            time.sleep(self.SECONDS_BETWEEN_UPDATE_STREAMS)
            #streams = get_top_streams(self.channels_amount)
            #self._log_streams(streams)

    def run_log_loop(self):
        """
        Creates the logger bots and update which stream they log every 60
        seconds.
        """
        if self.channels:
            self._run_static_streams_loop()
```

```
        else:
            self._run_popular_streams_loop()

    def stop_bot(self):
        for bot in self.bots:
            bot.join()
```

Dissertation Logger/CATEGORY: `utils.py`

```
import json
import requests
import time
from settings import API

from requests.exceptions import ConnectionError, SSLError


def get_top_streams(n):
    twitch_api_url =
"https://api.twitch.tv/kraken/streams/?limit=%i&language=en&game=%s" % (n,
urllib.quote('CATEGORY'))
    headers = {'Client-Id': API['CLIENTID']}
    try:
        return json.loads(requests.get(twitch_api_url,
headers=headers).text)['streams']
    except (ValueError, ConnectionError, SSLError):
        time.sleep(5)
        return get_top_streams(n)


def get_channel_names(streams):
    return [stream['channel']['name'] for stream in streams]


def current_time_in_milli():
    return int(round(time.time() * 1000))
```

REFERENCES

REFERENCES

Alexander, J. (2018a, January 18). YouTubers, Twitch streamers are opening up about serious burnout, personal struggles: It's an easy job. *Polygon*. Retrieved from https://www.polygon.com/2018/1/18/16899532/youtube-twitch-burnout-h3h3-pewdiepie-lirik

Alexander, J. (2018b, March 26). Abuse of KFC emote on Twitch leads to more conversations about toxic chat culture. *Polygon*. Retrieved from https://www.polygon.com/2018/3/26/17163582/kfc-emote-twitch-trihex-forsen-trihard-xqc

Alexander, J. (2018c, July 23). Creator behind Ninja's death hoax addresses the viral ligma meme. *Polygon.* Retrieved from https://www.polygon.com/2018/7/23/17602586/ninja-death-hoax-instagram-ligma

Altice, N. (2015). *I am error: The Nintendo family computer/entertainment system platform*. Cambridge, MA: MIT Press.

Andersen, P. B. (1992). Interaktive væker. En katastrofeteoretisk tilgang. Per Aage Brandt and others (editors): *Almen semiotik*, *5*, 89-112.

Anderson, S. L. (2017). Watching People Is Not a Game: Interactive Online Corporeality, Twitch.tv and Videogame Streams. *Game Studies*, *17*(1). Retrieved from http://gamestudies.org/1701/articles/anderson

Arnoux, P.-H., Xu, A., Boyette, N., Mahmud, J., Akkiraju, R., & Sinha, V. (2017). 25 Tweets to Know You: A New Model to Predict Personality with Social Media. *Eleventh International AAAI Conference on Web and Social Media*. Presented at the Eleventh International AAAI Conference on Web and Social Media. Retrieved from https://www.aaai.org/ocs/index.php/ICWSM/ICWSM17/paper/view/15681

Asarch, S. (2019, May 2). Twitch streamer Kun0 banned for 7-days after bullying DreamNekoTV. *Newsweek*. Retrieved from https://www.newsweek.com/kun0-ban-twitch-streamer-bullying-dreamnekotv-art-neko-drama-livestreamfails-1413478

Auter, P. J., & Davis, D. M. (1991). When characters speak directly to the viewers: Breaking the fourth wall in television. *Journalism Quarterly, 68*, 165-171.

Bauer, R. A. (1963). The initiative of the audience. *Journal of Advertising Research*, *3*, 3-21.

Bazarova, N. N., Taft, J. G., Choi, Y. H., & Cosley, D. (2013). Managing impressions and relationships on Facebook: Self-presentational and relational concerns revealed through the analysis of language style. *Journal of Language and Social Psychology*, *32*, 121-141.

Brand, S. (1972). Spacewar: Fanatic life and symbolic death among the computer bums. *Rolling Stone*, *7*, 50-58.

Bohemer, J. H., & Leith, A. P. (2015, March). *Tweeting the World Cup: Soccer teams' use of dialogic communication during the 2014 World Cup*. Presented at the The Eighth Summit on Communication and Sport, Charlotte, NC.

Borelli, J. L., Sbarra, D. A., Mehl, M., & David, D. H. (2011). Experiential connectedness in children's attachment interviews: An examination of natural word use. *Personal Relationships*, *18*, 341-351.

Bryant, J., & Vorderer, P. (2013). *Psychology of Entertainment*. Routledge.

Charmaz, K. (2006). *Constructing grounded theory: A practical guide through qualitative analysis.* Sage: London.

Cohen, D. S. (2016, October, 18). OXO aka Noughts and Crosses – The First Video Game. *Lifewire*. Retrieved from https://www.lifewire.com/oxo-aka-noughts-and-crosses-729624

Colliander, J., & Dahlen, M. (2011). Following the fashionable friend: The power of social media weighing publicity effectiveness of blogs versus online magazines. *Journal of Advertising Research*, *51*, 313-320.

Consalvo, M. (2009). There is No Magic Circle. *Games and Culture*, *4*(4), 408–417. https://doi.org/10.1177/1555412009343575

Coppersmith, G., Harman, C., & Dredze, M. (2014). Measuring Post Traumatic Stress Disorder in Twitter. *Eighth International AAAI Conference on Weblogs and Social Media*. Presented at the Eighth International AAAI Conference on Weblogs and Social Media. Retrieved from https://www.aaai.org/ocs/index.php/ICWSM/ICWSM14/paper/view/8079

Damashek, M. (1995). Gauging Similarity with n-Grams: Language-Independent Categorization of Text. *Science*, *267*(5199), 843–848. https://doi.org/10.1126/science.267.5199.843

Day9TV. (2010, April 14). *My Life of Starcraft - Day[9] Daily #100*. Retrieved from https://www.youtube.com/watch?v=NJztfsXKcPQ

Donovan, T. (2010). *Replay: The history of video games*. East Sussex, England: Yellow Ant.

Dwyer, T. (2010). *Media Convergence*. McGraw-Hill Education (UK).

Egliston, B. (2015). Playing Across Media: Exploring Transtextuality in Competitive Games and eSports. *DiGRA '15 - Proceedings of the 2015 DiGRA International Conference*, *12*. Retrieved from http://www.digra.org/wp-content/uploads/digital-library/122_Egliston_Playing-Across-Media.pdf

Fine, G. A. (2002). *Shared Fantasy: Role Playing Games as Social Worlds*. University of Chicago Press.

Fink, C., Kopecky, J., & Morawski, M. (2012). Inferring Gender from the Content of Tweets: A Region Specific Example. *Sixth International AAAI Conference on Weblogs and Social Media*. Presented at the Sixth International AAAI Conference on Weblogs and Social Media. Retrieved from https://www.aaai.org/ocs/index.php/ICWSM/ICWSM12/paper/view/4644

Fisher, B. A. (1978). *Perspectives on human communication*. New York: Macmillan.

FortyOne. (2019). John Archer presses ALT + F4. Retrieved from https://clips.twitch.tv/SplendidCrepuscularWebKappaClaus

Frederick, E. L., Lim, C. H., Clavio, G., & Walsh, P. (2012). Why we follow: An examination of parasocial interaction and fan motivations for following athlete archetypes on Twitter. International Journal of Sport Communication, *5*, 481-502.

Freitas, E. (2018, February 6). 2017 Twitch Year in Review: The Comic Book [Blog post]. Retrieved from https://blog.twitch.tv/2017-twitch-year-in-review-the-comic-book-ed1b08f8f116

Freud, S. (1960). *Jokes and Their Relation to the Unconscious*. W. W. Norton & Company.

Gantz, W. (1981). An exploration of viewing motives and behaviors associated with television sports. *Journal of Broadcasting*, *25*, 263-275.

girlwithyellowspoon. (2018). Jebaited Twice OLOLOLOLOLOL. Retrieved from https://clips.twitch.tv/StupidLittleCarrotPMSTwin

Glaser, B. G. (1978). *Theoretical sensitivity*. Mill Valley, CA: The Sociology Press.

Glaser, B. G., & Strauss, A. L. (1967). *The discovery of grounded theory*. Chicago, IL: Aldine.

Glaser, B. G., Strauss, A. L., & Strutzel, E. (1968). *The discovery of grounded theory: Strategies for qualitative research*. Piscataway, NJ: Rutgers.

Hall, E. T. (1966). A system of the notation of proxemic behavior. *American Anthropologist*, *65*, 1003-1026.

Hamari, J., & Sjöblom, M. (2017). What is eSports and why do people watch it? *Internet Research*, *27*(2), 211–232. https://doi.org/10.1108/IntR-04-2016-0085

Hamilton, W. A., Garretson, O., & Kerne, A. (2014). Streaming on Twitch: Fostering Participatory Communities of Play Within Live Mixed Media. *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, 1315–1324. https://doi.org/10.1145/2556288.2557048

Hartmann, T., Stuke, D., & Daschmann, G. (2008). Positive Parasocial Relationships with Drivers Affect Suspense in Racing Sport Spectators. *Journal of Media Psychology*, *20*(1), 24–34. https://doi.org/10.1027/1864-1105.20.1.24

Hefner, D., Klimmt, C., & Vorderer, P. (2007). Identification with the Player Character as Determinant of Video Game Enjoyment. In L. Ma, M. Rauterberg, & R. Nakatsu (Eds.), *Entertainment Computing – ICEC 2007* (pp. 39–48). Springer Berlin Heidelberg.

Hian, L. B., Chuan, S. L., Trevor, T. M. K., & Detenber, B. H. (2004). Getting to Know You: Exploring the Development of Relational Intimacy in Computer-mediated Communication. *Journal of Computer-Mediated Communication*, *9*(3). https://doi.org/10.1111/j.1083-6101.2004.tb00290.x

Horton, D., & Strauss, A. (1957). Interaction in Audience-Participation Shows. *American Journal of Sociology*, *62*, 579–587. https://doi.org/10.1086/222106

Horton, D., & Wohl, R. R. (1956). Mass Communication and Para-Social Interaction. *Psychiatry*, *19*, 215–229. https://doi.org/10.1080/00332747.1956.11023049

Iowa, J. D. P., A. Craig Baird Professor, The University of, & Simonson, P. (2004). *Mass Communication and American Social Thought: Key Texts, 1919-1968*. Rowman & Littlefield Publishers.

itmeJP. (2017a). *PC Specs*. Retrieved from http://itmejp.com/pc-specs/

itmeJP (Producer). (2017b, January 11). TwitchCon! British Person! Elite: Dangerous! [Episode 90]. *Dropped Frames*. Podcast retrieved from https://youtu.be/cFRnejw4iAM.

Kassing, J. W., & Sanderson, J. (2009). "You're the Kind of Guy That We All Want for a Drinking Buddy": Expressions of Parasocial Interaction on Floydlandis.com. *Western Journal of Communication*, *73*(2), 182–203. https://doi.org/10.1080/10570310902856063

Katzowitz, J. (2019, July 26). How the women of Twitch put up with the bullsh*t. *The Daily Dot*. Retrieved from https://www.dailydot.com/upstream/women-of-twitch/

Klimmt, C., Hartmann, T., & Schramm, H. (2006). Parasocial interactions and relationships. In J. Bryant & P. Vorderer (Eds.), *Psychology of entertainment* (pp. 291-313). Hillsdale, NJ: Erlbaum.

Klimmt, C., Rizzo, A., Vorderer, P., Koch, J., & Fischer, T. (2008). Experimental Evidence for Suspense as Determinant of Video Game Enjoyment. *CyberPsychology & Behavior*, *12*(1), 29–31. https://doi.org/10.1089/cpb.2008.0060

Korzenny, F. (1978). A Theory of Electronic Propinquity: Mediated Communication in Organizations. *Communication Research*, *5*(1), 3–24. https://doi.org/10.1177/009365027800500101

Lantz, F., & Zimmerman, E. (1999). Checkmate: Rules, Play and Culture. *Merge Magazine*, (5).

Lapowsky, N. T., Issie. (2018, December 17). How Russian Trolls Used Meme Warfare to Divide America. *Wired*. Retrieved from https://www.wired.com/story/russia-ira-propaganda-senate-report/

Leith, A. P. (2015, November). *Busking on a digital pitch: Investigating Internet-based gameplay broadcasting*. Paper presented at the Games as/with/through Communication: Introducing the Game Studies Division preconference of the National Communication Association 101st Annual Convention. Las Vegas, NV.

Leith, A. P. (2018). Playing games for others: Constructing a gameplay livestreaming typology. In R. Ratan, B. Winn, & E. LaPensée (Eds.), *Meaningful Play Proceedings 2018* (pp. 372 – 379). Pitssburg, PA: Carnegie Mellon University: ETC Press. doi: 10.1184/R1/9995969

Lemish, D. (1985). Soap opera viewing in college: A naturalistic inquiry. *Journal of Broadcasting & Electronic Media*, *29*, 275-293. https://doi.org/10.1080/08838158509386585

Lin, H., & Sun, C.-T. (2011). The Role of Onlookers in Arcade Gaming: Frame Analysis of Public Behaviours. *Convergence*, *17*(2), 125–137. https://doi.org/10.1177/1354856510397111

Link, D. (2012). Programming ENTER. Christopher Strachey's Draughts Program. *Computer Resurrection. The Bulletin of the Computer Conservation Society*, *60*, 23-31.

Loftus, G. R., & Loftus, E. F. (1983). *Mind at Play; The Psychology of Video Games*. New York, NY, USA: Basic Books, Inc.

Lowood, H. (2006). A brief biography of computer games. In P. Vorderer & J. Bryant (Eds.), *Playing video games: Motives, responses, and consequences* (pp. 25-42). Mahwa, New Jersey: Lawrence Erlbaum Associates, Inc.

Lueck, J. (2012). Friend-zone with benefits: The parasocial advertising of Kim Kardashian. *Journal of Marketing Communication*, *21*, 1-19.

Martinn, G. (2019, February 13). Deadmau5, a 38-year-old man, gets suspended on Twitch for using a homophobic slur. *Paste Magazine*. Retrieved from https://www.pastemagazine.com/articles/2019/02/deadmau5-a-38-year-old-man-gets-banned-by-twitch-f.html

Marwick, A. E., & boyd, d. (2011). I tweet honestly, I tweet passionately: Twitter users, context collapse, and the imagined audience. *New Media & Society 13*(1), 114–133.

Melnick, M. J. (1993). Searching for Sociability in the Stands: A Theory of Sports Spectating. *Journal of Sport Management*, *7*(1), 44–60. https://doi.org/10.1123/jsm.7.1.44

Mitchell, W. J. (1996). *City of Bits: Space, Place, and the Infobahn* (Revised ed. edition). Cambridge, Mass.: The MIT Press.

Newman, J. (2002a). In search of the videogame player: The lives of Mario. *New Media & Society*, *4*(3), 405–422. https://doi.org/10.1177/146144480200400305

Newman, J. (2002b). The Myth of the Ergodic Videogame: Some thoughts on player-character relationships in videogames. *Game Studies*, *2*(1), 1–17.

Newman, J. (2004). *Videogames*. Psychology Press.

Newman, J. (2008). *Playing with Videogames* (1 edition). London ; New York: Routledge.

Newman, M. Z. (2017). *Atari Age: The Emergence of Video Games in America*. MIT Press.

O'Neil-Hart, C., & Blumenstein, H. (2016). Why YouTube stars are more influential than traditional celebrities. Think with Google. Retrieved from https://www.thinkwithgoogle.com/consumer-insights/youtube-stars-influence/

Palmgreen, P., Wenner, L. A., & Rayburn, J. D. (1980). Relations between gratifications sought and obtained: A study of television news. *Communication Research, 7*, 161-192.

Park, J. H., & Lennon, S. J. (2004). Television apparel shopping: Impulse buying and parasocial interaction. *Clothing and Textiles Research Journal*, *22*, 135-144.

Pedersen, T., Banerjee, S., Joshi, M., Kohli, S., Liu, Y., McInnes, B., & Purandare, A., (2015). Ngram Statistics Package (Version 1.31) [Software]. Available from https://metacpan.org/pod/Text::NSP

Pennebaker, J.W., Boyd, R.L., Jordan, K., & Blackburn, K. (2015). *The development and psychometric properties of LIWC2015*. Austin, TX: University of Texas at Austin. DOI: 10.15781/T29G6Z

Perse, E. M., & Rubin, A. M. (1988). Audience Activity and Satisfaction with Favorite Television Soap Opera. *Journalism Quarterly, 65*, 368–375. https://doi.org/10.1177/107769908806500216

Perse, E. M., & Rubin, R. B. (1989). Attribution in social and parasocial relationships. *Communication Research*, *16*, 59-77.

Peters, J. D., & Simonson, P. (2004). *Mass communication and American social thought: Key texts, 1919-1968*. Lanham, MD: Rowman & Littlefield.

Peter, J., Valkenburg, P. M., & Schouten, A. P. (2005). Developing a Model of Adolescent Friendship Formation on the Internet. *CyberPsychology & Behavior, 8*(5), 423–430. https://doi.org/10.1089/cpb.2005.8.423

Pires, B. (2019). *A simple app for logging twitch's chat to a PostgreSQL database.: bernardopires/twitch-chat-logger* [Python]. Retrieved from https://github.com/bernardopires/twitch-chat-logger (Original work published 2015)

Ramirez, A., Jr., Zhang, S., McGrew, K., & Lin, S.-F. (2007). Relational communication in computer-mediated interaction: A comparison of participant-observer perspectives. *Communication Monographs*, *74*, 492–516.

Rasmussen, L. (2018). Parasocial interaction in the digital age: an examination of relationship building and the effectiveness of YouTube celebrities. *The Journal of Social Media in Society*, *7*, 280-294.

Ritterfeld, U., & Weber, R. (2006). Video Games for Entertainment and Education. In *Playing video games: Motives, responses, and consequences* (pp. 399–413). Mahwah, NJ, US: Lawrence Erlbaum Associates Publishers.

Rosengren, K. E., & Windahl, S. (1972). Mass media consumption as a functional alternative. In P. Vorderer & D. McQuail (Eds.), *Sociology of mass communications* (pp. 166–194). New York City, NY: Penguin.

Rubin, A. M., Perse, E. M., & Powell, R. A. (1985). Loneliness, parasocial interaction, and local television news viewing. *Human Communication Research*, *12*, 155-180.

Rubin, A. M., & Step, M. M. (2000). Impact of Motivation, Attraction, and Parasocial Interaction on Talk Radio listening. *Journal of Broadcasting & Electronic Media, 44*(4), 635–654. https://doi.org/10.1207/s15506878jobem4404_7

Salen, K., Tekinbaş, K. S., & Zimmerman, E. (2004). *Rules of Play: Game Design Fundamentals*. MIT Press.

Schiappa, E., Allen, M., & Gregg, P. B. (2007). Parasocial Relationships and Television: A Meta-Analysis of the Effects. In R. W. Preiss, B. M. Gayle, N. Burrell, M. Allen, & J. Bryant (Eds.), *Mass media effects research: Advances through meta-analysis* (pp. 301-314). Mahwah, NJ, US: Lawrence Erlbaum Associates Publishers.

Schmid, H., & Klimmt, C. (2011). A magically nice guy: Parasocial relationships with Harry Potter across different cultures. *International Communication Gazette*, *73*, 252-269.

Sherry, J. L., Lucas, K., Greenberg, B. S., & Lachlan, K. (2006). Video Game Uses and Gratifications as Predicators of Use and Game Preference. In *Playing video games: Motives, responses, and consequences* (pp. 213–224). Mahwah, NJ, US: Lawrence Erlbaum Associates Publishers.

Shoemaker, P., Schooler, C., & Danielson, W. A., (1989). Involvement With the Media: Recall Versus Recognition of Election Information. *Communication Research*, *16*(1), 78–103. https://doi.org/10.1177/009365089016001004

Silge, J., & Robinson, D. (2016a). tidytext: Text Mining and Analysis Using Tidy Data Principles in R. *The Journal of Open Source Software*, *1*(3), 37. https://doi.org/10.21105/joss.00037

Silge, J., & Robinson, D. (2016b, July 11). tidytext: Text Mining and Analysis Using Tidy Data Principles in R. https://doi.org/10.21105/joss.00037

Sniderman, S. (n.d.). Unwritten Rules. Retrieved April 12, 2019, from The Life of Games website: http://designblog.uniandes.edu.co/blogs/dise3349/files/2009/03/07-unwritten-rulesweb.pdf

Snow, A.. (2019, June 15). Many adult actors and actresses are now moonlighting as video game live streamers on Twitch—trolling be damned. *The Daily Beast*. Retrieved from https://www.thedailybeast.com/the-exciting-and-dangerous-lives-of-porn-star-gamers-the-harassment-was-hardcore

Schmid, H., & Klimmt, C. (2011). A magically nice guy: Parasocial relationships with Harry Potter across different cultures. *The International Communication Gazette*, *73*, 252-269.

Spitzberg, B. H., & Cupach, W. R. (2008). Fanning the flames of fandom: Celebrity worship, parasocial interaction, and stalking. In J. R. Meloy, L. Sheridan, & J. Hoffmann (Eds.), *Stalking, threatening, and attacking public figures: A psychological and behavioral analysis*, (pp. 287-321). New York, NY: Oxford University Press.

Strauss, A., & Corbin.J. (1994). Grounded theory methodology: An overview. In N. K. Denzin & Y. S. Lmcoln (Eds.), *Handbook of Qualitative Research* (pp. 273-285). Thousand Oaks, CA:Sage.

Strauss, A., & Corbin, J. (1998). *Basics of qualitative research: Grounded theory procedures and techniques* (2nd ed.). Thousand Oaks, CA: Sage.

SullyGnome (2019, July 9). *Twitch Statistics & Analysis - Games / Channels - SullyGnome*. Retrieved July 23, 2019, from https://sullygnome.com/

Taylor, T. L. (2009). The Assemblage of Play. *Games and Culture*, *4*, 331–339.

Taylor, T. L. (2012). *Raising the Stakes: E-Sports and the Professionalization of Computer Gaming*. Cambridge, MA: MIT Press.

Taylor, T. L. (2018). *Watch me play: Twitch and the rise of game live streaming*. Princeton, NJ: Princeton University Press.

Thorson, K. S., & Rodgers, S. (2006). Relationships between blogs as eWOM and interactivity, perceived interactivity, and parasocial interaction. *Journal of Interactive Advertising*, *6*(2), 5-44.

Twitch. (2017). *Twitch Blog*. Retrieved from https://blog.twitch.tv/

Twitchmetrics (2016, March 31). *Twitchmetrics - Streamers, Teams, & Games*. Retrieved July 23, 2019, from https://www.twitchmetrics.net/

Twitter. (n.d.). Clips now have Chat Replay right on the page! Relive the best moments on Twitch just as they happened [Twitter]. Retrieved from Twitter website: https://twitter.com/twitch/status/855111354197344257

Uszkoreit, L. (2017). *Video game live streaming and the perception of female gamers*. (Unpublished doctoral dissertation). University of Southern California, Los Angeles, CA.

Vorderer, P., Knobloch, S., & Schramm, H. (2001). Does Entertainment Suffer From Interactivity? The Impact of Watching an Interactive TV Movie on Viewers' Experience of Entertainment. *Media Psychology*, *3*(4), 343–363. https://doi.org/10.1207/S1532785XMEP0304_03

Wainess, R., Kerr, D., & Koenig, A. (2011). *Improving the Way We Design Games for Learning by Examining How Popular Video Games Teach. CRESST Report 798*. Retrieved from https://eric.ed.gov/?id=ED522827

Wainess, R., & Koenig, A. (2010). *Validation of a methodology for design and development of a game for learning with a multigroup development process*. Presented at the AERA 2010 Annual Meeting, Denver, CO.

Walther, J. B. (2010). Computer-Mediated Communication. In *The Handbook of Communication Science* (pp. 489–506). https://doi.org/10.4135/9781412982818.n28

Walther, J. B., & Burgoon, J. K. (1992). Relational Communication in Computer-Mediated Interaction. *Human Communication Research*, *19*(1), 50–88. https://doi.org/10.1111/j.1468-2958.1992.tb00295.x

Walther, J. B., & Parks, M. (2002). Cues filtered out, cues filtered in: Computer-mediated communication and relationships. In M. L. Knapp & J. A. Day (Eds.), *Handbook of interpersonal communication*, (pp. 529-563). Thousand Oaks, CA: Sage Publications.

Welbers, K., Atteveldt, W. V., & Benoit, K. (2017). Text Analysis in R. *Communication Methods and Measures*, *11*(4), 245–265. https://doi.org/10.1080/19312458.2017.1387238

Wenner, L. A., & Gantz, W. (1998). Watching sports on television: Audience experience, gender, fanship, and marriage. *MediaSport*, 233-251.

Wilson, J. M., Straus, S. G., & McEvily, W. J. (2006). All in due time: The development of trust in computer-mediated and face-to-face groups. *Organizational Behavior and Human Decision Processes*, *99*, 16–33.

Zillman, D., & Cantor, J. R. (1977). Affective responses to the emotions of a protagonist. *Journal of Experimental Social Psychology*, *13*(2), 155–165. https://doi.org/10.1016/S0022-1031(77)80008-5

Zillmann, D., & Paulus, P. B. (1993). Spectators: Reactions to sports events and effects on athletic performance. In R. N. Singer, M. Murphey, & L. K. Tennant (Eds.), *Handbook of Research on Sport Psychology* (First Edition edition). New York : Toronto : New York: Macmillan Pub Co.

Zillmann, D., & Sapolsky, B. (1989). Enjoyment from sports spectatorship. In J. H. Goldstein (Ed.), *Sports, Games, and Play: Social and Psychological Viewpoints* (2nd ed., pp. 241–278). Hillsdale, NJ: Erlbaum.