NOVA IMS Information Management School

MGI

Mestrado em Gestão de Informação
Master Program in Information Management

SENTIMENT ANALYSIS: THE CASE OF TWITCH

Mining user feedback from livestream chats

Jaime Mendes Gouveia Batalha Reis

Project Work presented as the partial requirement for obtaining a Master's degree in Information Management

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

SENTIMENT ANALYSIS: THE CASE OF TWITCH CHAT

Mining user feedback from livestream chats

Jaime Mendes Gouveia Batalha Reis

Project Work presented as the partial requirement for obtaining a Master's degree in Information Management, with a specialization in Information Systems and Technologies Management.

Advisor: Roberto André Pereira Henriques, Phd

Advisor: Aleš Popovič, PhD

November 2019

ABSTRACT

In a world where users often share their thoughts and opinions through online communication channels, applications that can tap into these channels as to extract consumer feedback have become increasingly valuable. Traditional marketing research techniques such as interviews or surveys offer results that pale in comparison to sentiment analysis applications that can extract organic feedback from an extremely large selection, with very little resources and in real-time. This thesis focuses on proposing and developing one of these tools that targets livestreams, which have, over the years, seen a massive increase in popularity from both a user-base standpoint as well as brand involvement. We chose the livestreaming platform “Twitch” as the target of research and developed a sentiment analysis model, using rule-based approaches, capable of interpreting user chat messages and identifying whether those messages are negative, positive or neutral. Additionally, an application was developed to better view and analyze the results of the model. By segmenting our results by product reveal, we also exhibit how the application allows for the extraction of various insights about the public’s opinion of that product.

KEYWORDS

Sentiment Analysis; Opinion Mining; Livestreams; Twitch; Text Mining;

INDEX

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

BTTV	Better Twitch TV
BC18	Blizzcon 2018
CSS	Cascading Style Sheets
ESPORTS	Electronic Sports
FFZ	FrankerFaceZ
HTML	HyperText Markup Language
MSG	Message
POS	Part-of-Speech
SAS EM	SAS® Enterprise Miner
SAS AS	SAS® Sentiment Analysis Studio
VOD	Video-On-Demand

1. INTRODUCTION

In the age of internet connectivity, consumers commonly voice their thoughts and opinions through various online channels. Being able to tap into these channels and extracting these thoughts and opinions offers valuable feedback for companies, especially if this feedback is expressed via a public forum such as Twitter or a livestream chat. Consumers play a very important role in influencing another consumers' purchase decisions, (Litvin, Goldsmith, & Pan, 2008; Liu, 2012) thereby increasing the impact these types of messages have depending on the size of exposure.

Various companies have started mining this information, focusing on social media as it is a common place consumers use to express opinions about different companies or products and provides users with an audience, in turn increasing the exposure and influence that this feedback might have. However, as the internet grows and new technology develops, new channels have opened that have equal or greater potential of providing companies with valuable feedback about their products.

One of these channels is livestreams. Livestreaming technology has existed for a long time, but has gained massive popularity over the recent years due to improvements in technology and the development of new platforms, allowing anyone to start their own livestream with relative ease. A parallel could be drawn to video content creation, which, much like livestreaming, saw a massive increase in popularity in previous years (Cha, Kwak, Rodriguez, Ahn, & Moon, 2007). Nowadays, video sharing platforms like YouTube are a giant industry with 1.9 billion monthly active users (Statista, 2019) and companies often use these platforms to advertise their products and collect consumer feedback. Conversely, companies and brands have also started to use livestreams as a means to release or announce their products, and often use them in their marketing strategies (Hackl, 2016).

The focus of this thesis will be on Twitch, one of the largest livestreaming platforms (Gandolfi, 2016). Twitch offers a livestreaming platform where any user from anywhere in the world can broadcast any gaming, lifestyle or creative content from either their computer or mobile phone. These livestreams, as is common with other livestreaming platforms, include a chat, where users can communicate and interact with other viewers, discussing the events of the livestream. This communication channel is filled with valuable information as consumers use the chat to express feedback, in real-time, to what they are being shown on the livestream.

The problem with this channel is that manual extraction of opinion is unfeasible for two reasons. First, depending on the number viewers, the chat can be extremely hard to read due to the waterfall of messages being sent each second. Secondly, these livestream platforms usually develop their own language and users often use platform-specific slang or terms that companies might not be aware of, or understand.

One of the solutions that have been used for other social media platforms like Twitter, involve building automatic tools that parse these large number of Tweets using different text mining techniques like sentiment analysis. These tools identify Tweets that contain an opinion, the target of this opinion, as well as the sentiment expressed by this opinion, thereby allowing the extraction of valuable information automatically and in-real time.

This thesis' objective is to develop a similar tool, one that companies can use to tap into this new channel of communication and perform an analysis which is able to extract valuable consumer feedback that would otherwise be unfeasible to do manually. We propose a tool, that can intake the chatlogs of a particular stream, run a sentiment analysis model that will attempt to classify each message based on its sentiment polarity (positive, negative or neutral message) and then showcase the output, in a way that can allow for the exploration of the results. This tool has to be general enough that can be used for any type of livestream on the platform and is able to work with very large datasets. The following goals were set:

- Explore, analyse and understand the type of communication found on Twitch.
- Build a Sentiment Analysis model that can receive raw messages sent by users and output, if found, the sentiment expressed by the user who sent the message. (Positive, Negative or Neutral)
- Build an easy-to-use website where the output of the model can be explored and cross-referenced with the livestream video, in turn allowing for the exploration and analysis of the results.

The data used was chosen based on different influencing factors such as the availability of data, the size of the dataset and the type of event. The aim was to find streamed events on Twitch that released one or more products into the market. Another factor considered was getting data from different types of audiences or communities which meant using different events and from different years, attempting to preserve external validity allowing our tool to be generalized for different types of streams.

With these factors in mind, three events were chosen:

- Blizzcon 2018 (Opening Ceremony Stream)
- Nike Smart Shoes Launch
- E3 2018 (June 10 stream)

Blizzcon is an annual convention hosted by Blizzard Entertainment, a gaming company. This event is held to promote their games and to announce upcoming titles. This thesis will be using the data from the opening ceremony of the 2018 stream as it is the day in which most products are revealed to the public. This stream lasted 2 hours and 45 minutes and contained 75063 chat messages.

Nike Smart Shoes Launch was an event hosted by Nike in which they revealed for the first time a new pair of shoes, dubbed Smart Shoes. This event went on for 1 hour and 39 minutes but only 5345 chat messages were sent over the course of the event. This considerably smaller stream was a perfect contrast to the other two as it was unrelated to gaming and had a much smaller data set.

The final event that was chosen was the **E3 2018** event. E3 stands for Electronic Entertainment Expo and is an event in which multiple video game studios reveal their products to the public. This event is a massive event, with many product launches, making it the perfect fit for our thesis. We chose the June 10th stream that lasted 9 hours and 22 minutes.

The first section of this thesis “Livestream Textual Analysis” addresses the past research done in the area, introduces information regarding the platform and the type of communication found on it. Next, in the “Methodology” section, we present the methodology used to develop this tool, including the way we collected, treated and analyzed the data, the design of the sentiment analysis model and the design of the website. Then, in the “Results and Discussion” section, we present the results of our model and tool. Finally, in the last two sections we discuss our conclusions as well as the limitations and recommendations of this thesis.

2. LIVESTREAM TEXTUAL ANALYSIS

2.1. TEXT MINING

Text mining can be defined as the process of deriving new, previously unknown information from various written sources through the use of digital means (Hearst, 2003). It is a sub-field of data mining that has received increasingly more attention these past years due to the large amount of available textual data that is created from all the new technology developments in both software and hardware (Aggarwal & Zhai, 2012). The unstructured nature and the extent of the data makes it unfeasible to manually glean any meaningful information and so new techniques like text mining had to be developed. This technique has been especially sought after by customer-centric companies due to its powerful value in decision-making information (He, Zha, & Li, 2013; Markham et al., 2015), ranging from analyzing customer feedback to predicting trends (Liau, Tan, & Pei, 2014).

Text mining combines different techniques from machine learning, data mining, natural language processing (NLP), information retrieval (IR) and knowledge management (Feldman & Sanger, 2006)

2.2. SENTIMENT ANALYSIS

Text mining can also be used to uncover sentiment. With the explosion of social media, the number of consumers willing to share their opinion and feedback has increased, which resulted in a stream of valuable data that can give companies real-time feedback and aid in decision-making. As to harvest this value, companies prefer using some text mining techniques over traditional analysis methods which are more time consuming (Liau et al., 2014). One of these techniques is called sentiment analysis, or sometimes called opinion mining which can be defined as the field of study used to analyze people's sentiment, attitude, or emotion expressed from textual language (Liu, 2012). A common way of classifying this data is to translate these sentiments into either positive, negative or neutral, through sentiment classification.

Consumers opinions play a major role in influencing the behavior and choices of other consumers as humans often seek the judgement of their peers when making a decision (Litvin et al., 2008; Liu, 2012). Word-of-mouth, for example in form of reviews, has been shown to have a significant impact on the sales of a product (Chevalier & Mayzlin, 2006; Nguyen Thi Ngoc, Nguyen Thi Thu, & Nguyen, 2019). This has led many companies to adopt sentiment analysis systems to capture this public opinion and use it for various purposes, from predicting sales to managing brand reputation (Liau et al., 2014; Liu, 2012).

Sentiment Analysis can be done in three different levels of scope in which the granularity will depend on the target of analysis. Research has been done in all different levels of granularity, varying from document level classification analysis (Pang & Lee, 2004; Turney, 2002), sentence level classification analysis (Hu & Liu, 2004; Kim & Hovy, 2004; Wilson, Wiebe, & Hoffmann, 2005) and feature level classification analysis. When the scope is the document, the analysis will classify the sentiment of the entire document, assuming each document as one opinion and attributing a polarity value (positive, negative, neutral) for example a product review. This is the most complicated of all three scopes as the analysis will have to take into account the relation between both sentences and words (Liu, 2012). When the scope is the sentence, the analysis will classify the sentiment of each sentence, assuming each sentence as an individual opinion. Finally, when the scope is the feature, the analysis will first identify the features that are being discussed, for example, a feature could be the "camera focus" or the "price" (including their synonyms), then it will classify these features by attributing a polarity measure (positive, neutral, negative).

2.3. LIVESTREAMING

Livestreaming has existed for a long time, it is an old technology that has been gaining massive popularity over the last few years (Haimson & Tang, 2017; Raman, Tyson, & Sastry, 2018; Scheibe, Fietkiewicz, & Stock, 2016) due to new advancements made in technology like larger bandwidth, better processing power in personal computers, new web services and better streaming platforms, allowing anyone to stream anything they want at any time, with little effort. A parallel could be drawn to video content creation, which, much like livestreaming, saw a massive increase in popularity in previous years (Cha et al., 2007), that was made possible due to technology advancements allowing any user to publish a video that could be seen by anyone, at any time. Nowadays, video sharing platforms like YouTube are a giant industry with 1.9 billion monthly active users (Statista, 2019).
Livestreaming has been used by individuals for multiple purposes over the recent years – In 2012, a landowner started a livestream of webcams pointed at a bald eagle nest on a livestreaming platform called Ustream, now owned by IBM, gaining over 16 million views over one year (Dick Pritchett Real Estate, n.d.). In 2016, Periscope, a popular livestreaming platform owned by Twitter, was used to livestream the sit-in protest in the US House of Representatives by democrat lawmakers to protest in favor of gun control (Guardian, 2016). In the same year, a woman used Facebook Live to stream an encounter with the police which lead to the death of her boyfriend (The New York Times, 2016).

In addition to individual generated content, livestreaming has also become a great way to promote corporate brand and more companies have started to adopt it in their marketing campaigns (Hackl, 2016). Famously in 2012, Redbull broke records by livestreaming a space stunt on YouTube Live, Google's streaming platform, which reached 8 million concurrent viewers (Youtube, 2012). In 2016, Buzzfeed used Facebook Live to livestream an exploding watermelon with more than 730,000 viewers watching for 30 minutes (Mosher, 2016). In the same year, Martha Stewart and The Home Depot collaborated to create a Facebook live stream of a Christmas ornament do-it-yourself (DIY) tutorial ("Facebook," 2016).

2.3.1. Streaming platforms and Twitch

Multiple livestreaming platforms have appeared over the years, Scheibe, Fietkiewicz, & Stock (2016) differentiate between two types of platforms, the first are general live streaming services which aren't limited to any theme, the second are topic-specific livestreaming services. In the first category, Scheibe, Fietkiewicz, & Stock (2016) names services like YouNow, Ustream, Periscope, Meerkat and YoutubeLive. And in the second category, they name Twitch for gaming content and Picarto for art content. Nowadays however, some of these platforms have expanded to more broad markets. Amazon's Twitch, for example, has been tapping into more creative avenues (Shaw, 2018) while YouTube decided to tap into more topic specific content by creating a separate app in 2015 exclusively for gaming content, then implemented into the main website in 2018.

The focus of this thesis will be on Twitch, one of the largest social streaming platforms (Gandolfi, 2016). The platform made its debut in June 2011 as a spin-off of another streaming platform called Justin.tv. Unlike Justin.tv, a general live streaming service, Twitch at the time was a topic-specific service, focusing solely on gaming content. Its popularity rapidly grew and by 2013, it had raised 45 million users, each watching, on average, 100 minutes of streams a day (M. Ewalt, 2013). The company was eventually bought by Amazon in 2014 for $970 million (Weinberger, 2016) and in 2018, according to Twitchtracker - a website that records data about the platform, it was estimated that Twitch had 560 billion minutes of streams watched, it had on average 3.4 million unique broadcasters every month, around 1 million average concurrent viewers and 41 thousand average concurrent channels (TwitchTracker, n.d.).

Twitch offers a livestreaming platform where any user from anywhere in the world can broadcast any gaming, lifestyle or creative content from either their computer or mobile phone. These users, called streamers, can create their own channel and stream under the thousands of different categories allowed by Twitch directly and in real-time to their audience, which are called viewers. These channels, as is common with other livestreaming platforms, come with a chat, where users can communicate and interact with other viewers, discussing the events of the stream. Twitch also offers a video-on-demand service, where if the streamer chooses so, the broadcasts streamed on their channel, along with the chat, will be saved on the platform, where it can be consumed on-demand. A feature we will be taking advantage of later on in our thesis.

One of the key unique features of this platform, compared to others, is their ability to give streamers multiple ways to monetize their content. Once streamers meet certain conditions they are made partners, giving them access to monetize their streams in 3 different ways. First, they allow viewers to subscribe to their channel by paying a certain amount each month, part of which goes to the streamer, the other to Twitch, giving the viewer certain benefits like subscriber-only emotes to use in the chat and a special badge. Secondly, the streamer earns a share from the advertisement videos that are played on their channel. Finally, Twitch gives its partners the ability to receive bits, a virtual good currency viewers can buy in order to send "Cheers" on your channel. Twitch will then share part of that revenue with the partner (Twitch, n.d.).

In addition to these 3 monetization avenues the platform employs, companies have realized the tremendous potential of livestreaming and often offer sponsorships to streamers. These brand deals vary from displaying their logo on the streamers' broadcast, product placement, or playing a game for certain amount of time, a common type of brand deal game developers use to promote a new upcoming game. This type of brand deal was so attractive and common that Twitch, in 2018, announced a new program called the 'Bounty Board Program' in which accepted streams can browse and accept different paid sponsorships opportunities, called bounties, directly from the platform. This offered a seemless way to connect sponsors with streamers and facilitate these types of brand deals. Ninja, the biggest streamer on the platform in 2018, made close to $10 million that year, according to an interview he did with CNN (Briggs, 2018).

Twitch is not only a platform for entertainers and gamers, the platform is also widely used to cover a variety of events like charity streams, esports (Eletronic Sports) tournaments and conventions.

2.3.2. Twitch User Communication

As previously mentioned, Twitch streams have a chat associated with them. This chat utilizes an old and fairly known application layer protocol called Internet Relay Chat (IRC) in which users can freely communicate with each other, in a chat box alongside the stream. Analysing this communication can prove very useful for companies, as it can contain valuable feedback, portraying the community's opinion towards a product, industry or service (Kaytoue, Silva, Cerf, Meira, & Raïssi, 2012).

Previous research into Twitch chat's communication is scarce. Of note, studies show that the size of the audience plays a big role on the type of communication of the audience (Ford et al., 2017; Hamilton, Garretson, & Kerne, 2014). Hamilton, Garretson, & Kerne (2014) argue that as the audience grows, the communicative nature of the chat transforms from a "meaningful medium of discussion" into an "illegible waterfall of text" that is too fast to be read. Adding that, due to the pace of messages being too fast to be read, the audience is no longer able to have one-on-one conversations with each other. More recent studies validate this notion, uncovering that messages in larger audience streams are shorter, have a higher number of emotes and are more repetitive in nature (Olejniczak, 2015). However, Ford et al. (2017) disagrees with the notion that the nature of larger audience chats is not meaningful, claiming that these types of chats "can still be examined as successful communication spaces in their own right". This dissertation will examine this type of chat, where it is less about the one-on-one conversation and more about the reactionary nature of the chat in response to the broadcast, which entails more inside jokes, shared references and twitch emotes (Ford et al., 2017).

One of the things that makes Twitch chat communication unique, is their use of emotes. The platform offers a wide variety of emotes for viewers to use at their disposal, these emotes can be global emotes, usable by any viewer, or subscriber-only emotes, which are emotes partnered streamers may upload to the platform that can only be used by viewers who subscribe to those channels but can be seen by anyone. In addition to the emotes provided by Twitch, users of the platform may also use third party extensions, which add a plethora of other emotes. The two main third-party emote extensions are "BetterTTV" (BTTV) and "FrankerFaceZ" (FFZ) (StreamElements, n.d.) which work in a similar fashion: each user that uses the extension can view and use the global emotes provided by the extensions and each channel is able to, by default, add 20 emotes (in the case of FFZ) and 5 emotes (in the case of BTTV) from a selected list that each extension provides, a list that is comprised of user-submitted approved emotes, which can then be seen and used in their channel.

The emotes Twitch uses on their platform differ slightly from the common emoji found in todays' modern services. Twitch emotes are used to convey an emotion or feeling but contrary to modern emoji, these emotes' meaning aren't easily decoded by looking at the graphical image, instead, the different meanings of each emote are based on the context and history of the emote (van der Weijden, 2017). Additionally, Twitch emotes don't use Unicode as most other platforms do.

Due to the large amount of emotes used in big audiences streams (Ford et al., 2017; Olejniczak, 2015) coupled with the fact they represent an emotion (van der Weijden, 2017) makes their interpretation essential when it comes to understanding messages and by extension, our sentiment analysis study. Later in the dissertation we'll analyze emotes and their meaning more in depth.

3. METHODOLOGY

This project was conducted following the flowchart depicted in Figure 1 below:

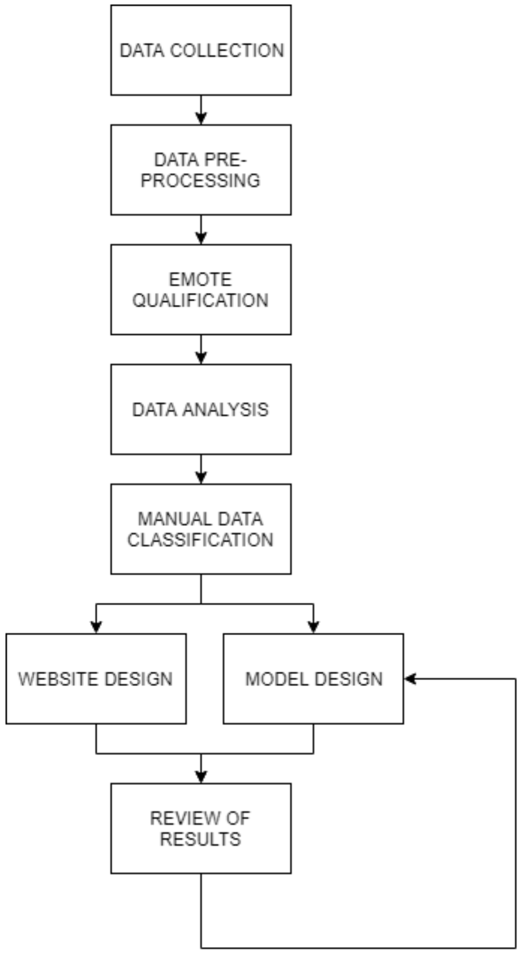


Figure 1 – Project Flowchart Diagram

We started off by collecting all the datasets that were deemed necessary for the execution of this thesis, then treated the data using pre-processing methods as to make it viable, useful and efficient for our tool. Next, we conducted a set of interviews to build a dictionary of emotes used on Twitch so that we could integrate them into our models. The next step involved a series of data exploration methods used to extract information about the kind of communication that is present on Twitch by analysing our datasets and transposing previous tacit knowledge on the subject, into explicit knowledge. Following that, we design our website and our sentiment analysis models through multiple iterations by analysing the results and consequently improving our model.

3.1. Data Collection

3.1.1. Twitch VODs and Chat Logs

As mentioned in 2.3.2, Twitch uses an application layer protocol called Internet Relay Chat (IRC), which we can tap into, using Twitch's own API. On February 2016, Twitch started recording chat messages that go alongside their Video-On-Demand (VOD) feature (McInnis, 2016). With this feature, we are able to use old VODs to train and test the sentiment analysis models. Through a tool called "RechatTool" (Purcell, 2019), we extracted the chat logs from the three aforementioned VODs. This tool taps into Twitch's API to extract the chat logs from past videos into two files, a JSON file and a .txt file. The latter being the one we use, which comes in the following format: {[H:M:S.ms]} {NICKNAME}: {MESSAGE} or, for example: [01:32:05.372] EricSmith: Hello!

To extract the stream video files, a tool called "Twitch Leecher" (Rebitzer, 2019) was used which allowed the extraction of the videos in .mp4 format, which will be useful in both the analysis part as well as the website later on.

3.1.2. Chat Emotes

The emotes used in user communication are subject to constant changes, whether due to the addition of new emotes, removal of others, or at times, because they just fall off in popularity. This dynamic environment means it is necessary the analysis covers a wide variety of emotes.

There are four main sources of Twitch emotes:

- Twitch Global Emotes
- Twitch Subscriber Emotes
- Third-party "Better Twitch TV" (BTTV) Emotes
- Third-party "FrankerFaceZ" (FFZ) Emotes

Due to the nature of the Twitch Subscriber Emotes (explained in 2.3.2), not only are they essentially unfeasible to extract, they are also constantly changing, which, coupled with the fact only the subscribers of each channel can use those emotes, in turn crippling their popularity, lead to the decision of not using them in this thesis.

Third-party emotes have a similar problem, their extraction is made difficult by the lack of available libraries and the large quantities of available emotes that streamers can choose to add to their own channels which makes the extraction a difficult challenge. As such, for the purpose of this thesis we will only extract the top emotes used in most channels, this list can be found on Appendix 1- Emote Lexicon.

Although Twitch Global Emotes are easily extracted, some of the emotes are not used as often which in turn affects what they are meant to express. If an emote is not used very often then what the emote is supposed to express won't be well defined, making it very difficult to integrate the emote into our model. Therefore, we extracted the emotes based on usage/popularity and added them to the list (Appendix 1). This list will later (section 3.5.3.1) be used to classify these emotes, creating an emote lexicon which will be part of our sentiment analysis model.

3.2. DATA PREPARATION

The raw chat log data collected is essentially unstructured data and as such, a few steps were taken as to clean-up and prepare the data before further usage. The first step involved a clean-up of irrelevant data, these include messages with the special character “@”, which on Twitch serves a similar purpose to other common platforms like Twitter - to address a specific person. Messages that include the “@” symbol, followed by a profile, indicate that the message is part of a conversation with another user in chat, rather than an opinion or reaction towards the actual stream. For this reason, the decision was made to remove all messages that include the @ symbol. Furthermore, we also removed any hyperlinks contained within the messages, which again do not have any added benefit for the purpose of this thesis. Finally, we removed the columns pertaining to the username of the person who sent the message and the attached “:” symbols, leaving only the timestamp and the message columns.

Following the clean-up, we segmented our chat log datasets using randomized sampling with the aid of a Microsoft Excel VBA script to extract 1500 chat messages that would later be used to manually classify the polarity of each message and then fed into our statistical model corpus for training.

Because SAS Sentiment Analysis Studio has a specific input requirement, requiring each message to be an individual text document, we developed a Python script that converted each line of our dataset text files into individual text documents. In addition, this Python script removes the remaining timestamp column and moves it as the file name. The result is a grand total of 328,208 text documents containing only the message and their respective timestamp as the filename, ready to be used as input.

The following Table 1 showcases the outputs of the data pre-processing over the three different datasets:

Table 1 – Data Pre-processing Results

Dataset	# of messages before clean-up	# of messages removed containing the symbol “@”	# of messages removed containing a URL	Total messages after clean-up
Blizzcon 2018	75063	654	0	74409
Nike	5345	326	47	4972
E3 2018	250192	1974	45	248173

3.3. EMOTE QUALIFICATION

As previously discussed, emotes play a very important role on Twitch, more so than most other platforms. Emotes can express a feeling faster and more reliably than words can, especially in an environment that provides very little screen-time for others to read your message. This abundance of emotes used throughout Twitch chat messages warrants a way to build them into our model.

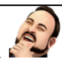
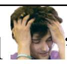
Indeed, using emotes in sentiment analysis research isn't necessarily a pioneer approach, however, the research conducted has been relatively limited. Pak et al. (2010) used emoticons like "☺" or "☹" as a reliable indicator of whether or not a tweet was positive or negative, thus using this method to collect positive and negative tweets and using them to train his model. Becker et al. (2013) used emoticons in a different way, their approach used the total number of positive and negative emoticons as features, using emoticons as part of his model as opposed to use them to train a model. Similarly, Selmer et al. (2013) used emoticons as features but through a different method that involved putting preprocessed emotes into binary, positive and negative, and then added that Boolean variable as a feature.

We propose a similar approach by using emotes, which are an extension of emoticons and emoji, as simple features for our Sentiment Analysis Model, which was also suggested by Guibon et al. (2016). Much like emoji, however, Twitch emotes don't represent a binary feeling, such as positive and negative. They can represent different feelings or at times not represent a feeling at all. Thus, it is important we dive into how emotes function on Twitch and then classify each emote per their linguistic purpose. Guibon et al. (2016) identifies three use cases for emoji in sentiment analysis:

- **Sentiment Expression**: In this case an emoji adds a sentiment to a message that would otherwise be considered neutral. An example would be the message "I can't go, I have to go to the dentist" versus "I can't go, I have to go to the dentist 😭". The former message conveys a neutral feeling towards having to refuse the offer due to having to go to the dentist while the latter conveys a feeling of sadness.
- **Sentiment Modification:** An emoji can be used to modify a sentiment that is present on the rest of the message. The author provides the following example "I'm so sad he's dead" versus "I'm so sad he's dead 😂". The laughing emoji modifies the feeling of sadness of the rest of the sentence and reveals the persons' real feelings.
- **Sentiment Enhancement:** Emoji can also be used to enhance a particular sentiment or feeling, consider the example "That's not cool" versus "That's not cool😡**",** while the first sentence does convey a feeling of discontent towards something, the second enhances this expression and makes it much more clear.

Despite Twitch emotes being different than emoji, our research reveals these three use cases identified by the author are also applicable in the context of Twitch chat and therefore will be used in our sentiment analysis model. In the following Table 2, we provide a few examples extracted from Twitch Chat messages which fall under these three different use cases.

Table 2 – Sentiment Use-case Examples

Sentiment Expression	“what is this ”
	“Diablo 4 announcement next ”
	“Wacraft 3 ”
Sentiment Modification	“Good production ”
	“yaay so excited ”
	“so interesting ”
Sentiment Enhancement	“this doesn’t look good at all ”
	“damn those graphics look good though  ”
	“finally a good announcement ”

In addition to these cases, Guibon et al. (2016) identifies other uses for emoji that do not convey a sentiment or emotion. The paper identifies the following:

- **“Notifier”**: Emoji used as a way to keep the persons’ attention by sending a random emoji or emoticon to keep the conversation going. The author provided the example of when people greet each other in chat rooms through the use of a greeting emoji.
- **“Convenience”**: Using emoji to replace words or sentences as a means of transmitting information faster and more conveniently.
- **“For fun”**: Emoji used because the person sending it thinks it is a funny emoji and therefore sends it in hopes the addressee will also find the emoji funny. The author acknowledges that this type of usage is not a standard occurrence but does occasionally happen.

While the sentiment-expression related use cases are in fact transferable to Twitch emotes in the context of twitch chat, we found these other uses cases differ slightly in this new environment. We propose the following use-cases instead:

- **“Convenience”**: as opposed to the use-case Guibon et al. (2016) identified, in this context, Twitch users will often use emotes to substitute words not because it is a faster way to relay information but because of visibility. During the fast-paced chat environment, an emote will stand out more than a word or sentence. For example, the emote “” will be used to greet the streamer and the viewers as opposed to the standard “Hello” or “Hey”. This emote conveys no feeling or sentiment as it is purely used as a replacement to a greeting.
- **“Mirroring”**: Another particularity of how users use emotes on Twitch is that sometimes they will mirror what they see on stream and use an emote that mirrors either an object, person, feeling or activity being transmitted. This type of usage is not meant to express what the user feels but rather to mimic what they see on stream. For example,

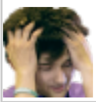
if a dog appears on stream, users often use a dog emote like "", or if country music starts playing users often use the emote "".

- **"Trolling"**: Another use-case relates to trolling. Sometimes the viewers act in bad faith to try and cause a reaction from the person streaming. For example, users might spam a "" emote, not because they want to let the streamer know they are bored, but instead to purposely try to get the streamer upset or feel bad as a way to get a reaction.

With these use cases in mind, the next step was to categorize and classify each emote that was extracted so they can properly be introduced into our model. As far as we understood, there hasn't been any research done in regards to classifying Twitch Emotes, let alone attributing a sentiment polarity. Emoticons are far easier as in most contexts, they are binary and a "☺" is positive and a "☹" is negative, while emoji classification proves more difficult. Currently, there is only one resource for emoji sentiment classification which can be found in a journal paper (Kralj Novak, Smailović, Sluban, & Mozetič, 2015) which queried 83 human annotators from different languages to be able to map and classify the sentiment of the emoji present in Unicode. However, the emotes used by Twitch only include some emotes that are equivalent to Unicode's emoji and the majority of the other emotes are not yet publicly classified. Therefore, due to the unavailability of secondary data we conducted interviews with 12 different frequent users of the platform.

These 12 interviews were conducted on a messaging platform similar to Skype called Discord. Each user was individually asked about each emote on our list and incentivized to translate the emote into one or more sentiment/feeling and then asked about the context in which these emotes are used. The results were then analysed, categorized, compiled, and added into a spreadsheet. Table 3 represents a small subset of the spreadsheet, which can be found in its entirety in Appendix 1.

Table 3 – Subset of the Emote Spreadsheet

Image	Emote Code	Meaning - Meme - Feeling	SE	SM	SEN	M	C	Binary Feeling	Source
	NotLikeThis	Conveys frustration or disappointment.	X		X			Negative	Twitch
	WutFace	Used to express that the viewer is scared or disgusted.	X		X			Negative	Twitch
	KappaPride	Mostly used as mirroring for anything related to lgbt, but can sometimes also be used as a sarcasm indicator.		X		X			Twitch
	DansGame	Mostly used to convey disgust.	X					Negative	Twitch
	JeBaited	Most often used when the streamer or the viewers were lead to believe something that wasn't true.				X	X		Twitch

3.4. DATA ANALYSIS

Our original corpus consisted of 330,600 documents or chat messages before cleanup. After we cleaned the data, we arrived at 328,208 documents ready to be analyzed, see Table 1 for a complete breakdown. Our goal was to find out more about the way users talk by analyzing the data and extracting insights. As stated in 2.3.2 there has been very limited research done in regards to analysing the sentiment of Twitch chat and most research involving sentiment analysis has been focused on either Tweets or online reviews which differ from the type of communication present in livestream chats. After studying hundreds of thousands chat messages these are the three main relevant differences identified:

- **"Video based context" vs "Stated Context".** While Tweets and reviews clearly state the context of their message, addressing the subject of their opinion in a sentence e.g. "*This movie is quite boring but the acting is brilliant, especially Leonardo DiCaprio*", "*Your customer service is terrible!! I had a bag badly damaged from a flight on 3 February and still no one has attended to my complaint*". The first example, a review found in a movie review website, clearly states the subject of the message, addressing the movie first and providing a general opinion, then later identifying a particular aspect of the movie "acting" and providing another opinion on this aspect. The second example, a tweet made to an airline company, also clearly states the subject of the complaint "customer service" and identifies what the problem is. In contrast, Twitch chat messages don't usually provide context, this context is usually deduced by what is happening on the livestream video and not by the message e.g. "*that looks horrible LUL*". In this example, there's no way to know what the user is addressing their opinion to, unless we look at the corresponding video timestamp, revealing the subject of the opinion is a game trailer for a video game called "Warcraft III Remastered".
- **Type of Communication**. When analysing reviews or tweets, you can expect multiple sentences with loaded statements that might consist of a problem, the reason for the problem and a sentiment on the problem. Conversely, Twitch chat messages are shorter on average and usually a sentence long, explained perhaps due to the nature of the chat itself, which moves relatively fast, discouraging long messages which are hard to read in a short span of time. The type of communication therefore is reduced to mainly short sentences, often loaded with a sentiment or an opinion.
- **"Shared Language".** While the average English-speaking person will probably be able to read and understand reviews or tweets written by different people, this might not be so obvious for Twitch chat messages. Twitch has, over the years, developed their own shared language, which is a combination of "memes" and emotes that have developed their own meaning. This "**Shared Language"** allows small messages to convey a feeling, sentiment or opinion in a condensed way, improving readability in a very fast-paced chat environment. For example, the one letter message "F", might not reveal at face-value any meaningful information but in Twitch's' "**Shared Language"** it translates to "I'm paying my respects", meaning one of two things, either the user is using its figurative meaning, for example in response to a product launch, signifying they're very discontent with the product and that it will not survive, or the user is using its literal sense, paying respects to someone who recently passed.

Next, to aid with this process of data exploration, we turned to SAS EM (SAS Enterprise Miner) which provides valuable text analytic and text mining tools which allow for the discovery of useful information inside a large dataset.

By first merging our three different datasets and importing them to SAS EM we are already able to extract meaningful information by looking at the document length by frequency graph. As depicted in Figure 2, most messages are short in length, confirming our initial assumption that the majority of chat messages are short, consisting only of one or two words. Another important piece of information we retrieved is that only 40.4% of the words in our dataset are recognized as English. The software uses dictionaries to try and categorize the language used in each document and in our case, it didn't recognize a language for 59.6% of the words in our dataset. This can be explained due to the previously mentioned "shared language" usage, which would confirm that this type of language is a major part of the communication found on Twitch, reinforcing its importance for our analysis.

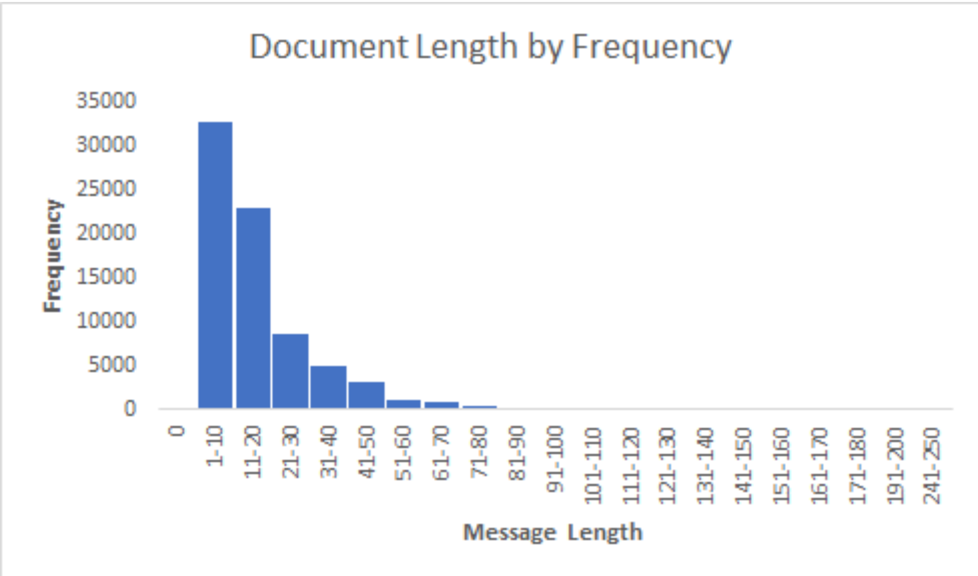


Figure 2 – Document Length by Frequency Graph

The next step involved series of different techniques used to improve the results of our analysis through filtering and parsing of our data. The first technique involved removing unnecessary words from our dataset. Some words, despite being often used in speech and writing, provide no significant information and are deemed "neutral" words. By removing these words, not only are we able to reduce the processing time but also remove any possible influence of these extraneous words over our analysis. This technique utilizes a built-in "**Stop Words**" list provided by SAS, consisting of these "neutral" words such as "at, as, are", and removing any matches. Additionally, as part of this removal, we included the removal of punctuation marks as well as numerical values.

Another technique used, denominated as **Part-of-Speech** (POS) Tagging uses linguistic technology to automatically detect the grammatical category of words in a sentence. Consider the following example: "*My conduct is extremely professional*". The word conduct can be both a verb and a noun depending on its sentence. This technology, however, will correctly identify "conduct" as a noun and "is" as the verb in the previous example. This technique categorizes each individual word in our dataset as one of the 15 possible grammatical categories thereby allowing the additional filtering of certain categories which are irrelevant to our analysis:

- Modal Auxiliary verbs. E.g. Can, could, must, shall.
- Conjunctions. E.g. yet, or, because, since, unless.

- Determiners. E.g. his, her, my, our.
- Prepositions. E.g. at, on, in.
- Pronouns. E.g. he, she, we.

Next, we use another technique called "**Term Stemming**". This technique further helps reducing processing time and improve the accuracy of this type of analysis by transforming words to their original root form. Using English dictionaries, SAS EM first takes words which are essentially the same, like "bigger" or "biggest" and replaces them with their root term "big", a process called "tail-chopping". Additionally, this technique takes into account Part-of-Speech (POS) tagging and will only replace words with their stem if they belong in the same grammatical category. Because text analytics is based on the relationships of terms across a certain dataset, by converging different variations of a term into one, the software is able to better find these relationships.

Finally, SAS EM offers a **spell-checking** option. This option will perform a spell-check to find spelling errors i.e. spelling "luve" as opposed to "love", and will add these errors to a synonym list, using the correct version as its parent. In addition, we took advantage of this synonym list and added our list of emotes, including a special role to the corresponding term, giving us a quick overview of how many emotes were used throughout the corpus, as seen in Figure 3. Unsurprisingly, out of the 1018971 words in our dataset (pre-filtering), 194653 were emotes, totaling a 20% of the total word count, confirming the assumption emotes play a huge role in this environment.

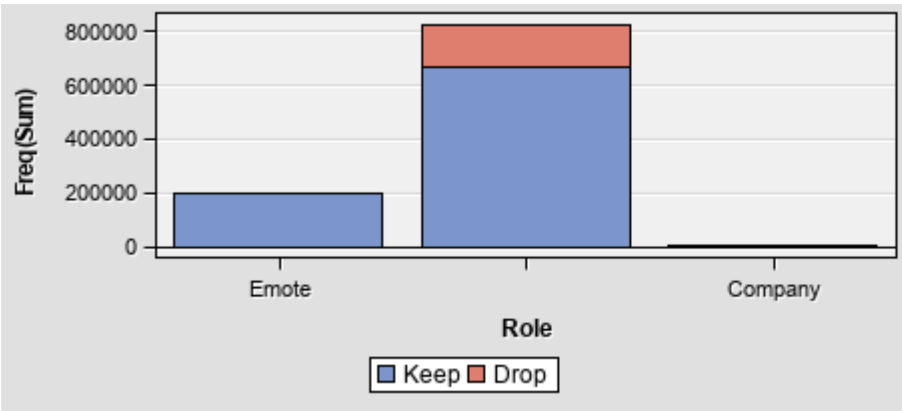


Figure 3 – Frequency by Role Graph

Following this pre-processing phase, the software generates a **term-by-document matrix**. This matrix is usually the starting point of any text analysis and is calculated through the following process:

1. Represent the text documents into vectors of words
2. Transform the vectors of words to numerical format by building a matrix in which each row translates to term vectors across all documents and each column translates to document vectors across all terms

Consider the following example:

Doc1: "I love my house"
Doc2: "I hate my house"
Doc3: "I love my husband"

The first step involves creating a word vector denominated "document vector". This vector will consist of every word in the document that wasn't selected to be discarded from our previous filtering options. Resulting in the following three vectors:

Doc1: (love, house)
Doc2: (hate, house)
Doc3: (love, husband)

The final step is to transform our word vectors into numerical format by building a term-by-document matrix. Table 4 depicts what the document matrix would look like, continuing our example.

Table 4 – Term-by-Document Matrix Example

Term	**Doc1**	**Doc2**	**Doc3**
love	1	0	1
house	1	1	0
hate	0	1	0
husband	0	0	1

If a term is present in a document, the value of the corresponding cell will be 1, if not, the value will be set to 0. This method is usually the starting point of any text mining technique, transforming our dataset into a large numeric matrix allowing the application of different statistical techniques. For our analysis, we first start by taking a look at Table 5, a table generated from this matrix applied to our dataset, a subset of which can be seen below.

Table 5 – Term Table

Term	# Docs	Freq	Role	Attribute	Status ▼
lul ...	17071	25343	Emote	Entity	Keep
+ residentsleeper ...	16308	60362	Emote	Entity	Keep
pogchamp ...	16213	32873	Emote	Entity	Keep
+ claim ...	6857	7061		Alpha	Keep
+ rip ...	6668	7154		Alpha	Keep
+ game ...	5707	8338		Alpha	Keep
+ hahaa ...	4341	6464	Emote	Entity	Keep
+ battle ...	4332	6452		Alpha	Keep
+ royale ...	4031	5961		Alpha	Keep
d: ...	3293	3789		Mixed	Keep
+ skate ...	2837	9100		Alpha	Keep
lol ...	2694	2764		Alpha	Keep
+ wutface ...	2616	3952	Emote	Entity	Keep
kappa ...	2607	3967	Emote	Entity	Keep
despacito ...	2556	6373		Alpha	Keep

This table, denominated the term table, is a table that contains every single term found in the document, after filtering and parsing. It is divided into seven columns:

- **“Term”:** The original root term with no spelling errors.
- **“#Docs”:** Number of documents that contain the term.
- **“Frequency”:** Number of times the word or term appears in the dataset.
- **“Role”:** Entity Classification of the term, in our case we only added one additional role “Emote”.
- **“Attribute”:** An indication of what type of characters are used to compose that term. I.e. Alpha if all the characters are letters or Mixed if they include a combination of letters and punctuation.
- **“Status”:** Whether we choose to keep this word or not, previously decided based on our filtering preferences.

Further analysis of this table reveals that by ordering by #docs, eighteen out of the top thirty terms are part of the aforementioned “Shared Language”. The first three terms: “lul”,”residentsleeper” and ”pogchamp” are all emote codes and can be distinguished as they have a substantial lead over the rest of the terms – about ten thousand more documents contain the third place term “pogchamp” than the fourth place “claim”. This information is very beneficial since two out of the three terms: “residentsleeper” and “pogchamp” have a very distinct sentiment polarity indication thus allowing their exploitation in our sentiment analysis model. Furthermore, the list indicates that most of the terms present in the most amount of documents are terms related to a sentiment, with the exception of the terms: ”claim”,”despacito”,”battle”,”royale”,”skate” and “game”. All of which were expected terms, with the exception of the term “claim”, which seemed out of place.

SAS Text Miner provides a technique called “**Concept Linking**”, which outputs a hyperbolic tree of the terms that are most associated with the term you’ve selected. By using this capability, we are able to find out the neighbour terms associated with certain terms. We applied this technique to the term “claim”, in hopes of figuring out why it was placed so high in our term matrix. By analysing the concept tree depicted in Figure 4 and expanding its branches, we are able to quickly determine the nature of this term: whether it is by misinformation or a regulated giveaway, we determined the cause is due to multiple users trying to claim some sort of gift or prize by typing a certain message in chat that contains the term “claim”. This sort of message falls under the scope of irrelevant information.

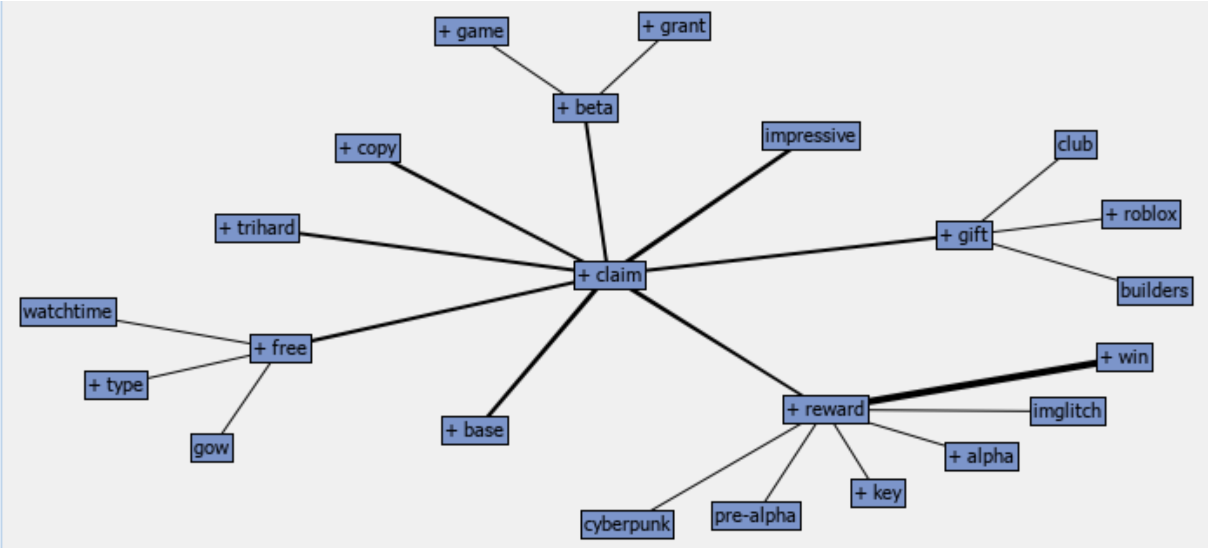


Figure 4 – Concept Link Tree of the term "Claim"

Using this same technique, we study other prominent terms in an attempt to understand more about the communication and context in which these terms are being used with the goal of improving our sentiment analysis model.

3.5. Model Design

Using the insights extrapolated from our data analysis and research, we developed a model adjusted to the particularities of the environment of this project. The model chosen was the following:

- Technique
  - Statistical Approach
  - Rule-Based Approach
  - **A combination of both**
- Granularity
  - **Document level**
  - Sentence level
  - Word level
- Rating Level
  - Aspect Rating
  - **Global Rating**

The choice of using a **hybrid approach**, combining both a rule-based and a statistical approach was in part due to the nature of the aforementioned "Share Language" (section 3.4). Through a rule-based model we can introduce various rules and conditions to address the particularities of this type of communication and with the use of a statistical model, which allows for feature extraction through the use of manual reviewed chat messages, we are able to find features and topics that will facilitate the building of our rule-based model. The specifics and drawbacks of these language processing techniques will be discussed later in sections 3.5.2 and 3.5.3.

In terms of **granularity**, due to the previously mentioned type of communication found on Twitch, which generally consists of one sentence or one opinion, the decision was made to use document-level, aiming to find the sentiment polarity of the whole message, which is consistent with most sentiment analysis models used in the reviewed literature.

Finally, in terms of the **rating level**, a few elements weighed in on the decision. The first element relates to one of the purposes of the project, which strives to develop a model that functions for different types of streams, whether it is a company revealing new shoes or a videogame company releasing their new game trailer. This desire for external validity makes global rating a better fit for our model, as brands would not have the need to manually add the different aspects that could be associated with each stream. Another important element that lead to the decision of using global rating is the lack of context in a chat message. Since most chat messages will forgo the subject of their opinion as it is deduced by what they are seeing, most messages do not have an aspect and thereby decreasing the benefits of using aspect ratings over global ratings.

3.5.1. Manual Classification of Data

As previously mentioned in 3.2, the dataset was segmented using randomized sampling to manually classify our data. After extracting 1500 messages we categorized their sentiment polarity to either positive, negative or neutral messages. The Table 6 below displays the results.

Table 6 – Manually Trained Data Results

Sentiment Polarity	**Number of Documents**
Negative	532
Positive	323
Neutral	645

The documents were then divided into their respective folders, “Positive”, “Negative” and “Neutral”, ready to be fed into our statistical model for training.

3.5.2. Statistical Model

The statistical model is an automatic method that relies on statistical techniques such as the Bayes Theorem, to classify the sentiment polarity of each document. In our case, we are just interested in using the features this model outputs so that we can implement them into our rule-based model. In order to achieve this, the model relies first, on a technique designated “Bag-of-Words”.

3.5.2.1. Bag-of-Words

The bag-of-words technique, in short, is a way to extract prominent features from text that we can later use for modelling. It is called “bag-of-words” because it discards the structure, order and context of the sentences and just keeps the words. In order to do this, the machine must first convert the text into numbers, thereby allowing the use of machine learning algorithms. Thus, it first converts every document into a numerical vector. Consider the following example of documents:

Doc1: What an awesome game
Doc2: That was horrible
Doc3: this was an awful performance

The first step involves creating a list of unique terms of words used in all documents. In our case, the list would be the following:

1. what
2. an
3. awesome
4. game
5. that
6. was
7. horrible
8. this
9. awful
10. performance

Finally, it creates numerical document-vectors, essentially converting each document to a numerical binary vector, removing any text from the equation:

Doc1: [1,1,1,1,0,0,0,0,0,0]
Doc2: [0,0,0,0,0,1,1,0,0,0]
Doc3: [0,1,0,0,0,1,0,1,1,1]

This way, the order and structure of our documents is not taken into account, and as such, proper feature extraction can begin. This example, however, would prove very inefficient if the size of the dataset is large, as it would create a very extensive vocabulary list and consequently incredibly large document vectors. A common way to reduce the dimensions of these vectors and one SAS SM (SAS Sentiment Analysis Studio) does, is to do pre-processing of the dataset. This pre-processing involves a lot of the techniques that have been discussed previously in this thesis (section 3.4):

- Removing punctuation
- Identifying and merging miss-spelled words
- Term-stemming (using only the original version of the word)
- Removing stop-words
- Being case-insensitive

Once our document vectors are built, the final step is to extract prominent features. Using our manually classified data, we divided our previously-classified documents into positive, negative and neutral folders and through this method, the words that score higher in each category will be considered a prominent feature.

The output is a list of positive, negative and neutral features with an associated weight based on the frequencies of those word across the other terms in their category. These weights were not directly used in the model, instead, they were used as a valuable frequency indicator. The features themselves were used to improve and complete our word lexicons, thereby improving our rule-based model.

3.5.3. Rule-based Model

Rule-based models work well in data centered around domain-specific taxonomies, which is the case of our thesis, centered around Twitch and its type of communication. This type of sentiment analysis model offers the tools needed to address these domain-specific taxonomies by building rules customized to our domain which can be improved cyclically by looking at the results and adding new rules to address the issues encountered.

3.5.3.1. Lexicons

The first step of our model-building was creating our word, phrase and emote lexicons where our rules could be built upon. These lexicons englobe words, phrases or emotes that serve a purpose in determining the sentiment polarity of a sentence. Our main domain-independent lexicon that we used in our thesis is an English opinion lexicon which contains roughly 6800 words, separated between positive opinion words and negative opinion words. This lexicon has been developed over many years of analyzing different reviews and extracting opinion words from prominent features that were found (Hu & Liu, 2004). The presence of these opinion words, however, do not necessarily indicate whether a sentence expresses a positive or a negative opinion, consider the following example:

*"These shoes don't look **good** at all"*

The sentence contains one of the opinion words found in the positive list but the sentence expresses the opposite sentiment. Word lexicons alone are not good enough and it is important we build rules upon which we can train the model to identify the different cases which do in fact express a sentiment.

Although we based our domain-independent lexicon on Hu & Liu's (2004), some modifications were necessary in order to address problems discovered in the multiple testing iterations. For example, the word "wow" was included in the positive word list but in gaming it is mostly used as an acronym for the game "World of Warcraft". To address this particular example, we removed the word "wow" and replaced it with a REGEX rule that dictates it will only consider the word as a positive word if it has more than one "o", in which case, the word would not be used as the acronym unless a spelling mistake was made. Most modifications made were due to conflicts of the type of domain-specific language and domain-independent language, a few examples of removed words include "loot", "leak", "leaking", "leaks" and "reward". However, there were also modifications that were made for different reasons, examples include the word "dope" which was included in the negative opinion word list as opposed to the positive list and the word "hype" which was also incorrectly located in the negative word list.

Next, we built our own lexicon for Twitch Emotes. Based upon the previous work in 3.3, a lexicon was compiled containing commonly used emotes that expressed, with a strong degree of certainty, a sentiment expression, enhancement or modification. Table 7 shows the list of emote codes (which refer to an actual emote) used.

Table 7 – Emote Lexicon

Positive Emotes	Negative Emotes	Sentiment Modification
PogChamp	DansGame	Kappa
Pog	ResidentSleeper	Keepo
Poggers	FailFish	Kapp
PogU	BrokeBack	Krappa
AngelThump	VoteNay	KappaHD
VoteYea	CoolStoryBob	KappaRoss
❤	NotLikeThis	KappaClaus
SeemsGood	WutFace	
Clap	☹	
FeelsGoodMan	haHAA	
FeelsAmazingMan		
HYPERS		
CurseLit		
GivePlz		
bleedPurple		

Finally, a slang-type lexicon was compiled with the aim of capturing domain-specific language used on Twitch. This language ranged anywhere from common internet slang words or phrases to specific Twitch language, designated previously as “Shared Language”. A few examples include: “lit”,”take my money”,”yikes”,”trash”,”F”,”ruined”.

These lexicons were compiled based on the product of the previous text-mining analysis done in section 3.4, the emote qualification of section 3.5, the statistical model features uncovered in section 3.5.1, the analysis of results of the multiple model iterations as well as previous tacit knowledge of the domain.

3.5.3.2. Intermediate Entities

The next step involved using these lexicons and building different lists that could then be referenced and used by our rules. SAS SM provides a place in which we can extract them to, designated “Intermediate Entities”. In addition, SAS SM also provides a Part-of-Speech (POS) tagging tool, identical to the one found in their text mining software. This POS tagging tool is extremely useful as it helps avoiding ambiguities between words. Consider this next example:

*“I’m **pretty** sure this I’ve seen this before”*

The word pretty is included in one of our positive opinion word lexicons because if used as an adjective, it carries a positive connotation attached to it. However, as the above example showcases, “pretty” may also be used as an adverb, carrying no sentiment connotation. These multi-purpose words carry an ambiguity that can be solved by using a POS tagger, isolating the type of grammar category that, in conjunction with the opinion word, give it a sentiment connotation.

The final list of intermediate entities used in this thesis can be found on Table 8:

Table 8 – List of Intermediate Entities

Intermediate Entities	Description
posWords	List of words extracted from the positive opinion-word lexicon
negWords	List of words extracted from the negative opinion-word lexicon
posTwitch	List of words or phrases extracted from the positive domain-specific lexicon.
negTwitch	List of words or phrases extracted from the negative domain-specific lexicon.
posEmotes	List of positive emotes extracted from our Emote Lexicon
negEmotes	List of negative emotes extracted from our Emote Lexicon
posAdj	References positive adjectives sourced from the "posWords" list.
negAdj	References negative adjectives sourced from the "negWords" list.
sarcasmIndicators	List of emotes extracted from our emote lexicon that modify the sentiment of the message.
negationWords	List of words or phrases that indicate negation.

3.5.3.3. Rules

The next step involved designing the rules. These rules are used to find patterns in our text documents by matching specific words, phrases or word-sequences that can be considered an indicator of whether the document is positive, negative or neutral. These matches work on a left-to-right basis and matches that are longer will take precedence over shorter ones.

There are 6 possible rule types:

- **Classifiers:** The most basic type of rule and which consists of a simple string representing one word. For example, the list "posWords" consists of 1,992 classifier type rules, one rule for each word.
- **Concept:** Essentially an upgraded version of classifier-type rules allowing the reference to intermediate entities or other concepts through the "_def" tag as well as adding modifiers, for example, specifying that it will only match words if the first letter is capitalized with the "_cap" tag.
- **C_concept:** Same type of rule as the concept rule, but it will only match concepts that are within a certain context. In this type of rule, you use the "_c" tag to identify the concepts you want to match and then provide other rules to state the context in which this concept will be accepted as a match.
- **Concept_rule:** Building upon the "c_concept" rule, in this type of rule, the context you specify must have, as a requirement, at least one Boolean rule. For example, consider the problem described in section 3.5.3.2, where we would like to reference the words

located in our "posWords" list, but only if they are adjectives. Then we'd create a "concept_rule" type rule:
(AND,"_c{_def{posWords}}",":A")
The "AND" operator is a Boolean operator that will return true in case both arguments are present in the document, or false if they are not. The first argument, prefixed by the "_c" tag, references our "posWords" intermediate entity, using the "_def" tag, which is a list of classifier type rules containing every positive word or term that we extracted from the positive opinion word lexicon. The second argument ":A" is a POS tag that will match an adjective. Therefore, this rule will only find a match if it has found a word located in our posWord list (the concept) but only if this word is considered an adjective (the context).

- **Predicate_rule:** Expanding on both "concept_rule" and "c_concept" type rules, this type allows the matching of patterns to be product or aspect-specific. Since we do not use products in our model, we will not be using this type of rule.
- **REGEX:** While other types of rules are aimed at the sequence of words, REGEX type rules are aimed at the sequence of characters that make up a word or term. This type of rule allows for a much more thorough match of specific terms, especially helpful when dealing with emotes.

These rules are then assigned as neutral, positive or negative and given a weight. This weight is a numerical value which attempts to score a document on its sentiment. Consider a document that returns 5 rule matches, 3 of them positive, 2 of them negative. Assuming each rule weighs the same, then the document will be classified as a positive document with a positive sentiment score of 1.

The rules used in this thesis have been built upon from past research in the area (Becker et al., 2013; Nasukawa & Yi, 2003; Wiegand, Balahur, Roth, Klakow, & Montoyo, 2010; Wilson et al., 2005; Wu & Jin, 2013), adapted to the particularities of our domain, scope and data. The collection of rules was built organically from multiple iterations by analyzing the results and making the necessary changes needed to address them. They are designed so that they will match patterns that suggest the sentiment polarity of the whole document, whether they match specific terms, phrases or sequences of words.

The most basic, yet fundamental rules were added first. These are the concept-type rules that identify positive and negative words or phrases present in a document. Consider the following examples on Table 9:

Table 9 – Examples of Starting-point Rules

Positive	Weight	**Negative**	Weight
_def{posAdj}	1	*_def{negAdj}*	1
_def{posTwitch}	1	*_def{negTwitch}*	1
_def{posEmotes}	3	*_def{negEmotes}*	3

The rule presented in the first row references the "posAdj" or "negAdj" intermediate entities, which contain one rule that matches words from the "posWords" or "negWords" intermediate entities if they are also considered an adjective. It is important to mention that single-word documents, which is a common occurrence in this type of communication, will also be counted as adjectives by the POS tagger and therefore count as a match if this word matches one of the words in either the "posWords" or "negWords" list. This rule is given a weight of 1, so that each match found will give the document a sentiment score of 1.

The second row rules references our "posTwitch" and "negTwitch" intermediate entities, included in those entities are the list of words and phrases present in our slang lexicon. Each occurrence of these words or phrases in a document will also give the document a sentiment score of 1.

Finally, the rules presented in the third row of Table 9 reference our "posEmotes" and "negEmotes" intermediate entities which include the list of positive and negative words (emote codes) extracted from the emote lexicons. As stated in 3.3, a single emote is enough to shift a sentiment of a sentence from a positive to a negative and vice versa, making it a considerably stronger sentiment indicator than a single adjective, as it uncovers the true feelings of the person who sent the message. Thus, the weight given to this rule is higher than the previous two.

These fundamental rules are a good starting point but they are not sufficient. While words, common phrases and emotes can help deciphering the sentiment of a document, contexts too, are important. That is why we should also build rules that consider the sequence of words rather than just the words themselves. Take the following examples of documents found in our dataset:

Doc1: "mmm not bad"
Doc2: "Self lacing shoes isn't revolutionary sodaTHINKING"
Doc3: "d3 was never good OMEGALUL"

If our rule-based model only matched specific positive, negative or neutral terms and phrases, then our model output would be the following:

Doc1: "mmm not bad"
Doc2: "Self lacing shoes isn't revolutionary sodaTHINKING"
Doc3: "d3 was never good OMEGALUL"

Thus, classifying Doc1 as negative and Doc2 and Doc3 as positive. However, any human can determine these documents carry, in fact, the opposite sentiment. One of the ways we can avoid these errors is by using a negation rule which detects the negation effect caused by certain sequences of words. Wiegand et al. (2010) and many others have also introduced rules that account for the negation effect. By introducing a list of words that indicate a negation e.g. "not","isn't","never","nobody","doesn't","can't" we are able to create a rule that finds these types of phrases and assigns their correct sentiment. For example, the following concept rule:

_def{negationWords} _def{posAdj}

This rule will match positive word adjectives that are preceded by a negation word, giving the document a negative sentiment score. The counterpart rule, with negative adjectives, would also be added. Together, these two simple rules would cover the majority of negation cases found in our dataset, but not all. Consider the following example present in our dataset: "this isn't so good". As the negation word "isn't" was not directly preceding the word "good" it wasn't returning a match from our negation rules and therefore would consider the document positive. In order to broaden our rule whilst being careful not to allow false positives, we can modify the rule into a "concept_rule" type:

(ORDDIST_3, "_def{negationWords}",(OR,"_c{_def{posTwitch}}","_c{_def{posAdj}}))

ORDDIST_N is an operator that will only return a match if the arguments are in a left-to-right sequence and within a defined *N* distance from each other. *N* being the number of words in between. In other words, this rule will only find a match if it encounters a negation followed by a positive adjective (or a positive slang) within a 3-word limit.

Another form of speech that is present throughout our dataset that must be accounted for relates to the use of sarcasm. Sarcasm has been studied in multiple fields, including sentiment analysis and is considerably difficult to detect, even for human beings (Farias & Rosso, 2017). Luckily, there are certain indicators that are unique to the type of communication present on Twitch which clearly state that the message carries a sarcastic tone. These indicators come mostly in the form of emotes that have the sole task of telling the reader that the recipient meant the message to be interpreted as sarcasm. Much like the negation effect, these emotes discussed in 3.3, which were also pulled from the emote lexicon, are used for "Sentiment Modification" and modify the sentiment of the whole document towards their polar opposite. Consider the following examples, present in our dataset:

Doc1: "very cool Keepo"
Doc2: "wow so excited Kappa"

These two documents contain two emotes normally used to indicate sarcasm, "Keepo" and "Kappa". Without the two, both documents would appear to have very positive sentiment connotation attached, but when one of the sarcasm indicators is present, they are meant to be perceived as negative sentences.

To address this effect, we first created an intermediate entity denominated "sarcasmIndicators" where we deployed all the words, phrases and emotes that indicate sarcasm. These include the emotes found in the emote lexicon that were classified as sarcasm indicators, as well as other words and phrases that are used to indicate sarcasm e.g. "said no-one ever". Then, we created the necessary "concept_rule" type rules that would address all the possible scenarios, for example the following rule:

(SENT,(OR,"_c{_def{posTwitch}}","_c{_def{posAdj}}","_c{_def{posEmotes}}),"_def{sarcasmIndicators}")

"SENT" is an operator that will only return a match if both arguments are found in the same sentence. In other words, if the model finds positive slang, positive adjectives or positive emotes as well as a sarcasm indicator in the same sentence, then it will return a match. In this example, it would then give the document a negative sentiment score.
This example only servers to address one possible scenario, but there are many others like a sentence that contains a negation, a positive adjective and a sarcasm indicator. Thus, multiple rules were created to address all the different possible scenarios.

A few neutral rules were also created; these types of rules serve to override the positive or negative rules and grant the document a neutral status. Most notably a rule was created to avoid comparative sentences, such as these found in our dataset:

Doc1: "guys shut up diablo is better than heatstone and owerwatch"
Doc2: "Adidas is more advanced than NIKE lol"
Doc3: "more amazing than any other athletes Kappa"

Comparative sentences have been used in sentiment analysis before, notably Ganapathibhotla & Liu's (2008) paper whose work in opinion mining of comparative sentences served as the groundwork for other research. However, for the purpose of our model, we noted that most comparative sentences present in our dataset were actually irrelevant, as they contained opinions targeted at different topics.

Previously, we discussed how the type of communication observed on Twitch had a "video-based context", rather than "stated context", implying that most sentiments/opinions are targeted at what is being shown on stream and not explicitly stated in the message. Comparative messages, however, often come with context and the target of their opinions is clearly stated, which works against us on two fronts: First, more often than not, the opinion these messages transmit is targeted at something other than what is being shown on-screen, which is an opinion we are not interested in capturing. Secondly, because our methodology does not involve products or aspects, it would be very difficult to capture whether the target is in fact whatever is being shown on stream. Consider the following example extracted from our dataset:

"destiny is better than hearthstone"

This message was played during an announcement related to the game "destiny". Therefore, this particular opinion was targeted at something happening on-stream which would in fact fall under our scope as something we are interested in capturing. However, this information is only available to us because we can go to the corresponding timestamp of the video and find out what product was being revealed at the time. Since we can't capture that in our model, these types of comparative sentences are very hard to evaluate. Nevertheless, there are a few exceptions. Consider the following examples:

Doc1: "this is much better than what I expected"
Doc2: "that looks even worse than no man's sky"

Documents 1 and 2 are also comparative sentences but this time the target of opinion in both examples are what is currently being shown on stream. These types of sentences therefore fall under the scope of our project as opinions we want to capture. The following are few examples of the rules created as to address comparative sentences:

Positive: (ORDDIST_6,(OR,"this","that","it"),"_c{_def{posAdj} than}")
Negative: (ORDDIST_6,(OR,"this","that","it"),"_c{_def{negAdj} than}")
Neutral: _def{posAd} than
Neutral: _def{negAdj} than

Since longer matches take precedence over smaller ones, with these rules, if a document contains the words "this","that" or "it" before a positive or negative adjective, followed by the word "than", then it will consider them as positives or negatives accordingly. If the document doesn't contain the words "this", "that" or "it" before a positive or negative adjective, followed by the word "than", then it will consider the document as neutral.

In addition to these comparative rules stated above, a few rules were added due to the fact the software's POS tagger has a slight problem identifying comparative adjectives and so manual rules had to be implemented to circumvent these difficulties.

Finally, it is also important to mention that every step of the way, the rules created were built upon from either past research in the area, or built organically from the results displayed over countless iterations, where we tried to broaden our rules as to cover more and more cases, without creating false-matches, consequently improving our model. The bulk of the rules in our model address little deviations in words, emotes or phrases that viewers use which can have a significant impact on the results. For example, the word “yeesssss” should be considered as a positive slang-type word as it transmits a positive sentiment connotation. However, if the word appeared in its original form “yes”, then it would have no sentiment connotation. Therefore, in our “posTwitch” intermediate entity, one of the rules added is a REGEX-type rule:

[Yy][Ee][Ee][Ss][Ss]**

This rule will return a match, independently of letter capitalization, if there are any additional letters “e” or “s”. Another example of these types of rules is for example the REGEX-type rule located in our “posEmotes” intermediate entity:

*[P][Oo][Gg]\w**

This rule will return a match when a word starts with the capitalized letter “P”, followed by case-insensitive letters “o” and “g” including words that contain more letters after the initial three. This type of rule will address the fact that there are many positive emote-codes that start with “Pog”, a few of them are present in the emote lexicon but by adding this rule, we address all of them and future emotes that might be added.

3.6. Website Creation

The type of communication present on Twitch, lacks context as this context is answered by what is visibly or audibly being transmitted on the livestream. This lack of context is a product of the nature of livestreams and it is reinforced the more people there are chatting. As such, there are two main reasons that lead to the development of a website:

1. Further testing and validating our model.
2. Finding a more complete visualization of our sentiment analysis results.

The first reason relates to the design phase of our model. SAS Sentiment Analysis Studio (SAS SM) provides a good way to go over the results and understanding the textual reasoning that lead to certain sentiment polarity classification decisions which was regularly used to improve and validate our process when designing and testing rules. However, this method is still bound by that lack of visual and audible context that is only given by the accompanied video stream. This context may influence our decisions by providing additional information that might change our interpretation of certain messages, regarding the intended sentiment. By building a website containing these contextual elements, we’re able to further improve the results of our model. Moreover, due to the size of the dataset, analyzing results one-by-one as SAS SM offers, is both difficult and inefficient. Through a website, a more holistic overview of the results could be achieved, finding additional insight that we would otherwise miss.

The second reason that lead to the website creation involves the need for a better overview and visualization of the results. The intent of this project involves providing an efficient tool, that businesses might use, that can be used to collect insightful feedback from users when revealing products. The output given by our sentiment analysis model will only serve to dictate when the users were displaying positive or negative sentiments but would not give any additional information as to the target of their content or discontent. As such, creating a tool that addresses those issues and allows for a more complete visualization, so that meaningful information can be extracted, was imperative to our project.

With these objectives in mind, a set of goals were extrapolated:

- Easy-to-use
- Allows for context to be explored and cross-referenced with results
- Allows for a more holistic overview of results
- Is adaptable to any data size

With an objective set and with these goals in mind, a website that would satisfy these requirements was designed using HTML (HyperText Markup Language), CSS (Cascading Style Sheets) and JavaScript. The plan involved creating a one-page website that would accept the input from the model and visually represent it alongside the stream VOD (Video-On-Demand) and the chat. These three components (The chat, the video and the visual representation) would all be interactive and synced.

The output given by SAS SM can be extracted via CSV file that contains all the documents fed into it and the adjacent positive, negative and neutral sentiment scores. This file serves as the input for our website. In addition, the video that was previously extracted, mentioned in 3.1.1, would also serve as input.

The next step involved finding the best way to visually represent the data. A stacked area chart enables the display of the evolution of several groups on the same graph by stacking them on top of each other. In our case, we have four different variables we want to study: negative, positive, neutral and unclassified. This type of graph gives us the opportunity to visualize all four at once and study the relative proportions of each throughout the course of the livestream. Additionally, this graph would have to be interactive due to the possible huge datasets that could be inserted.

In order to design such a graph, a JavaScript library called d3.js was used. The X-axis would represent the timestamp in seconds, with an interval adjusted to the length of the video and the Y-axis would represent the frequency of messages per second. Consider the following example:

The length of the video is 60 seconds. The sentiment analysis model outputs a table containing the timestamp of the message, and its classified sentiment. The website would then divide this output into four categories (Unclassified, Neutral, Positive, Negative) and count the number of messages of each category every 10 seconds. I.e.

[0,10] seconds = 20 Unclassified, 7 Neutral, 5 Positive, 15 Negative
[10,20] seconds = 14 Unclassified, 3 Neutral, 13 Positive, 12 Negative
[20,30] seconds = 0 Unclassified, 0 Neutral, 32 Positive, 0 Negative
(…)

Then, each individual frequency is calculated by dividing each count by 10:

[0,10] seconds = 2 Unclassified, 0.7 Neutral, 0.5 Positive, 1.5 Negative, 4.7 Total
[10,20] seconds = 1.4 Unclassified, 0.3 Neutral, 1.3 Positive, 1.2 Negative, 4.2 Total
[20,30] seconds = 0 Unclassified, 0 Neutral, 3.2 Positive, 0 Negative, 3.2 Total

(…)

Using this method, we essentially condense the amount of data points being represented by a factor of 10, improving readability and processing time, especially in large datasets. Then, we finalized the stacked area chart by giving it a window-zoom functionality (allowing the graph to be zoomed in on specific timestamp intervals), a legend, and a hover mechanism (enabling the user to quickly extract values by hovering over specific areas).

Next, aiming to enable ease of data exploration alongside context, we first added a line marker on our graph that identifies the exact timestamp our video player is currently on. Then, we synchronized the chat, the graph, and the video player together such that any change in timestamps would also change the other elements. This in turn enables the controller to view specific times on the video and their corresponding results and vice-versa.

Finally, we added green, red and blue background to certain chat messages, based on our sentiment analysis results, signaling that the specific message was classified as positive, negative or neutral respectively.

The following Figure 5 shows a screen cap of the website with data from the Blizzcon 2018 dataset and its respective results:

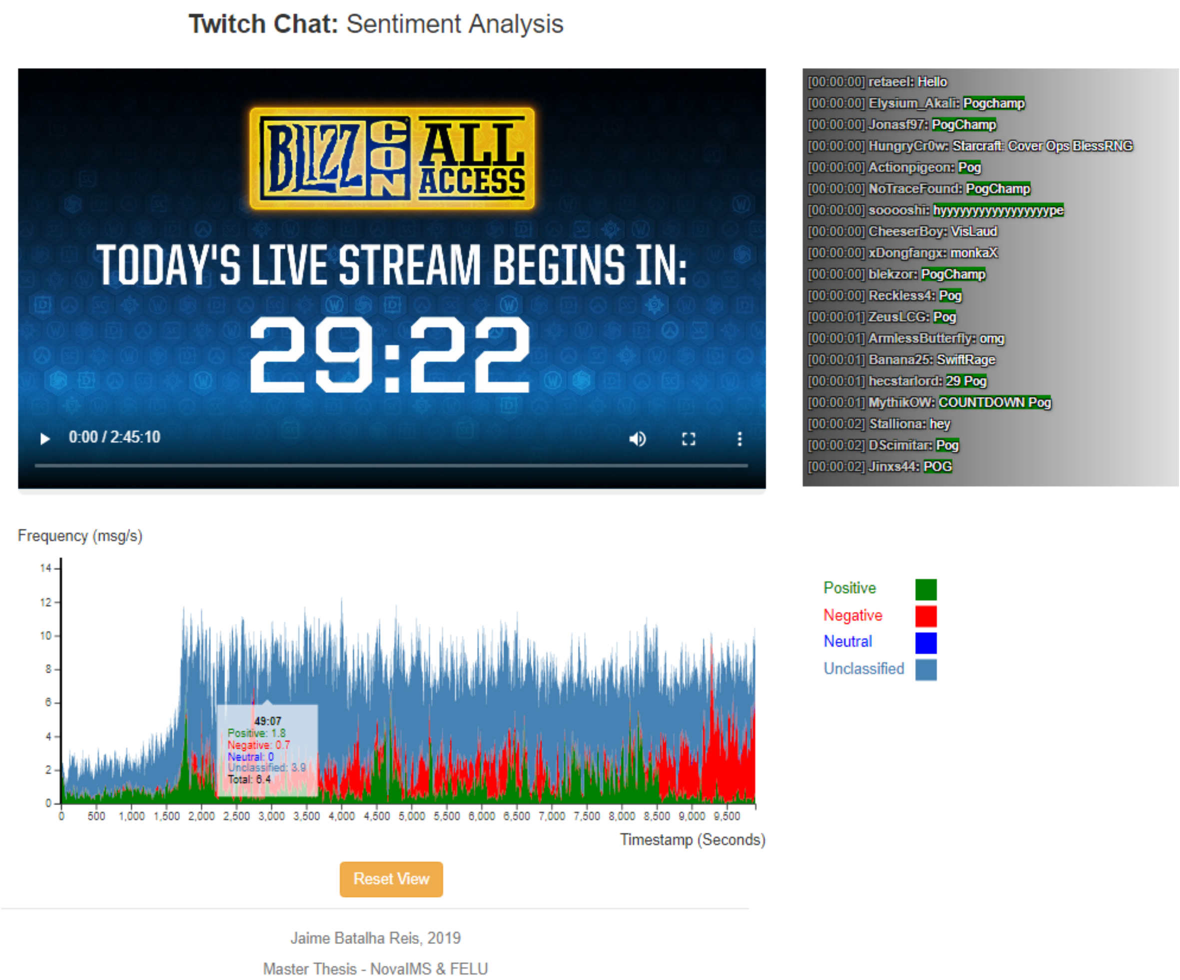


Figure 5 – Screen cap of the Website

4. RESULTS AND DISCUSSION

4.1. SENTIMENT DISTRIBUTION OVERVIEW

Upon multiple iterations of our sentiment analysis rule-based model, improving the results with every new iteration, the model achieved results that were satisfactory for the purpose of this project. Below, Figure 6 represents an overview of the results for each of the three datasets studied.

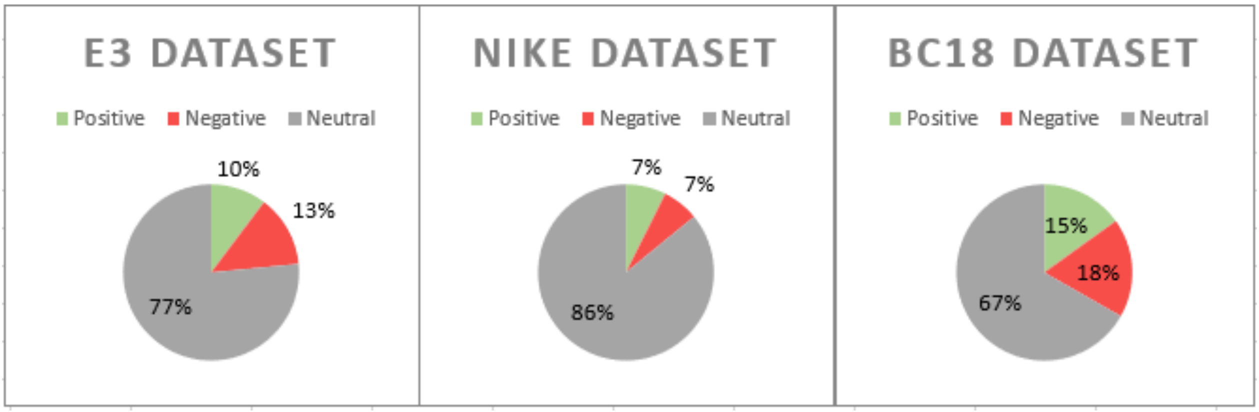


Figure 6 – E3, Nike & Blizzcon 2018 Dataset Pie-Chart Results

Table 10 – Average Frequencies

Dataset	Average Pos Freq. (msg/s)	Average Neg Freq. (msg/s)	Total Freq. (msg/s)
Nike	0.06	0.05	0.813
BlizzCon 2018	1.11	1.331	7.197
E3 2018	0.741	0.947	7.360

For the purpose of displaying these graphs, the unclassified messages - messages the model did not deem neutral, positive or negative based on the set of rules given, were merged with the neutral messages.

Regarding the **BlizzCon 2018** dataset, the model categorized 11,026 positive messages and 13,267 negative messages out of a total of 74,409 messages. This dataset was our second largest and yet contained the biggest number of messages classified with a sentiment polarity, at 33 percentage points. Additionally, the model considers the overall sentiment was more negative than positive, by 3%.

On the **E3 2018 dataset**, the model categorized 24,995 positive messages and 31,949 negative messages out of a total of 248,173. An important distinction is the fact that this dataset, in addition to having the greatest number of messages in a dataset, originated from a stream that revealed the most products. And yet, analysing the distribution, the model only classified 23% of the total messages as negative or positive.

As for the **Nike** dataset, the smallest dataset of the three, the model categorized 358 positive messages and 325 negative messages out of a total of 4972 messages. The model only recognized 14% of the total number of messages as positive, or negative. A possible reason that justifies these values is the change in the type of communication observed in smaller streams as previously discussed. By analysing individual messages, we can identify an increase in conversations between users in chat as opposed to messages targeted at the product being revealed. It is also observable there are considerably less messages carrying a sentiment connotation as opposed to the other datasets.

This type of result analysis serves only as a starting point as it is missing important contextual information that may only be obtained by analysing different segments of each dataset. First, we analyse this sentiment distribution by means of our website-generated graph.

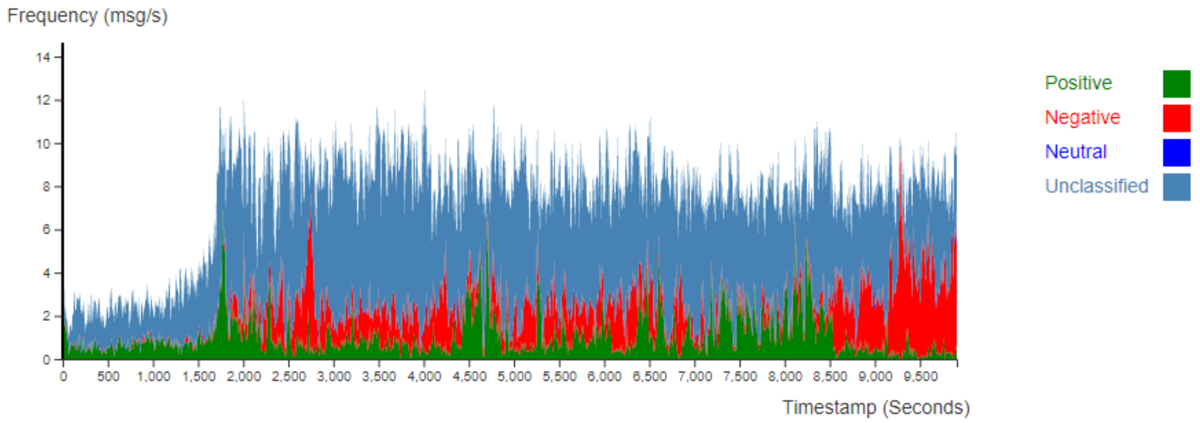


Figure 7 – Complete Stacked-area Chart of the BlizzCon 2018 Dataset

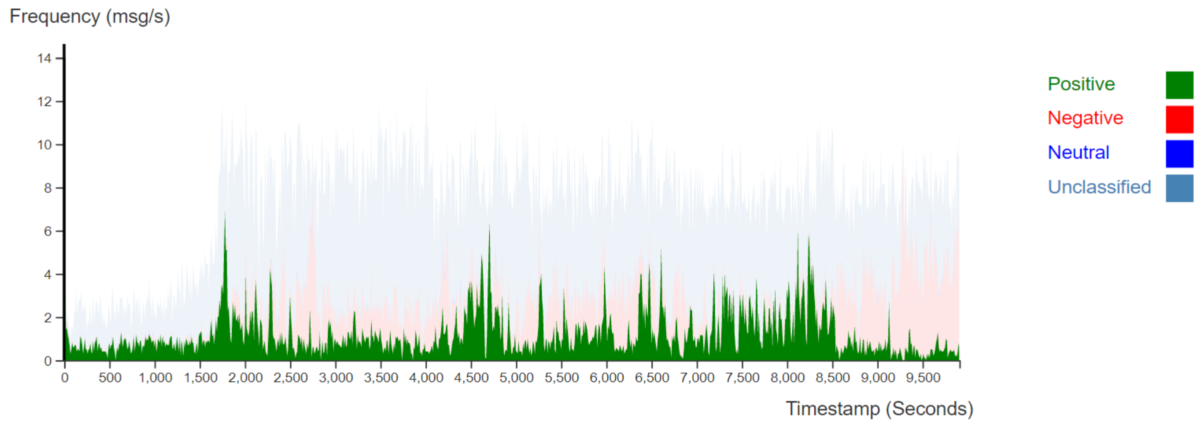


Figure 8 – Positive-selected Stacked-area Chart of the BlizzCon 2018 Dataset

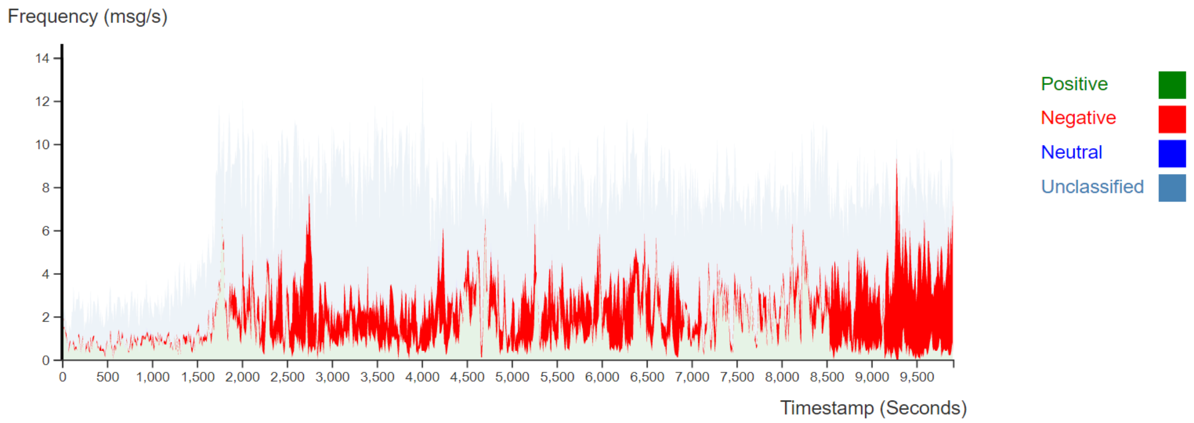


Figure 9 – Negative-selected Stacked-area Chart of the BlizzCon 2018 Dataset

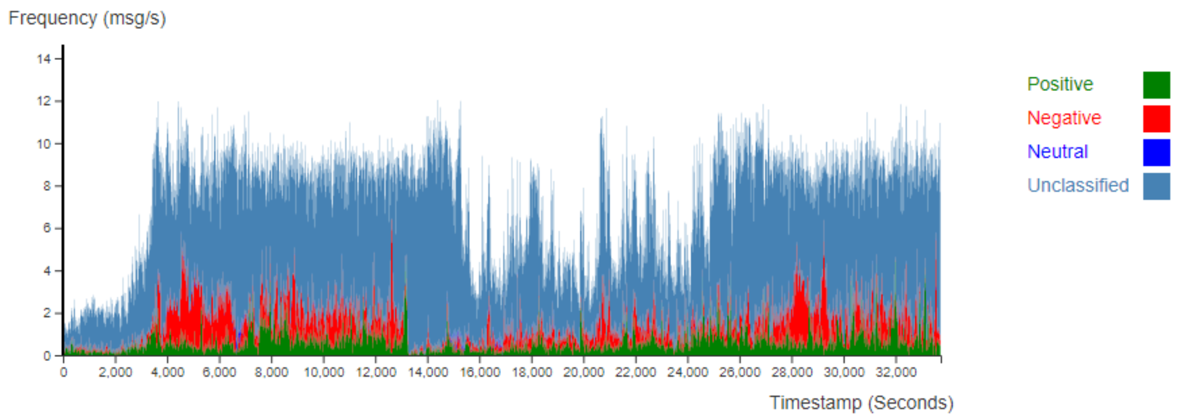


Figure 10 – Complete Stacked-area Chart of the E3 2018 Dataset

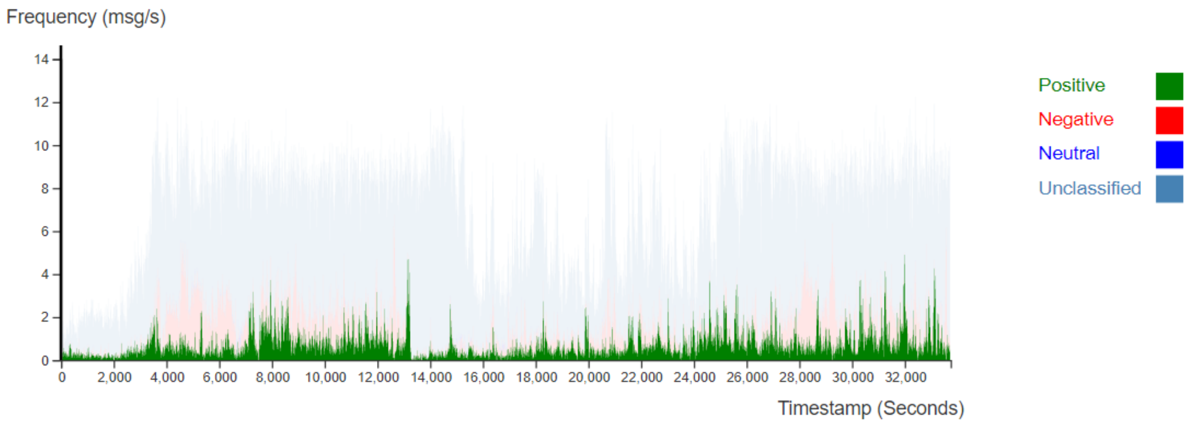


Figure 11 – Positive-selected Stacked-area Chart of the E3 2018 Dataset

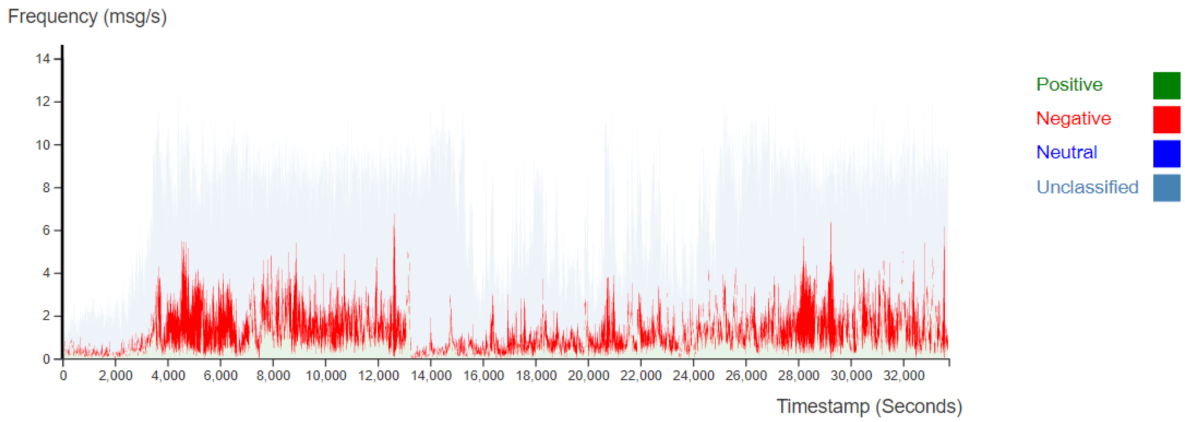


Figure 12 – Negative-selected Stacked-area Chart of the E3 2018 Dataset

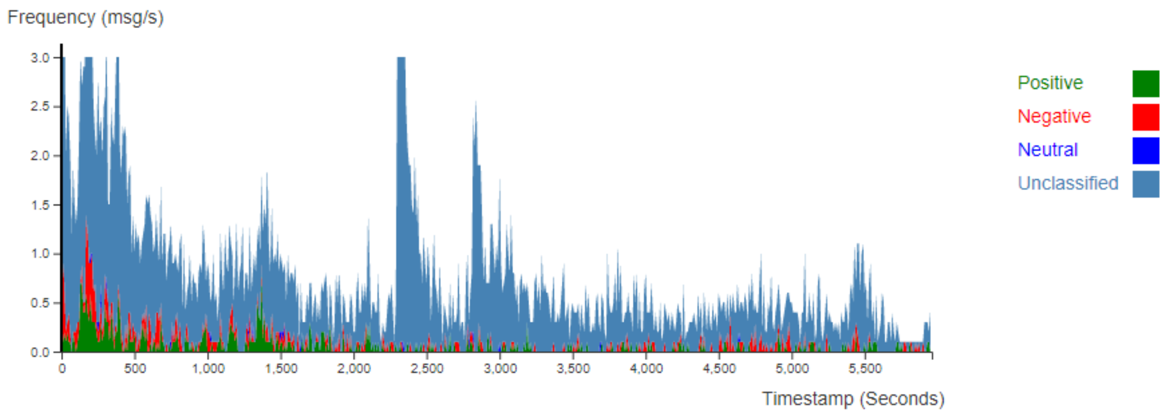


Figure 13 – Complete Stacked-area Chart of the Nike Dataset

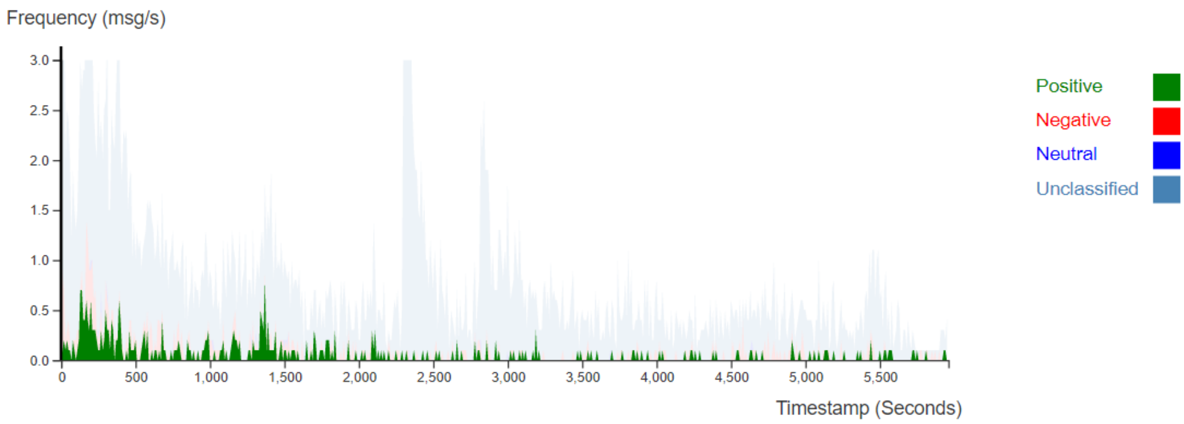


Figure 14 – Positive-selected Stacked-area Chart of the Nike Dataset

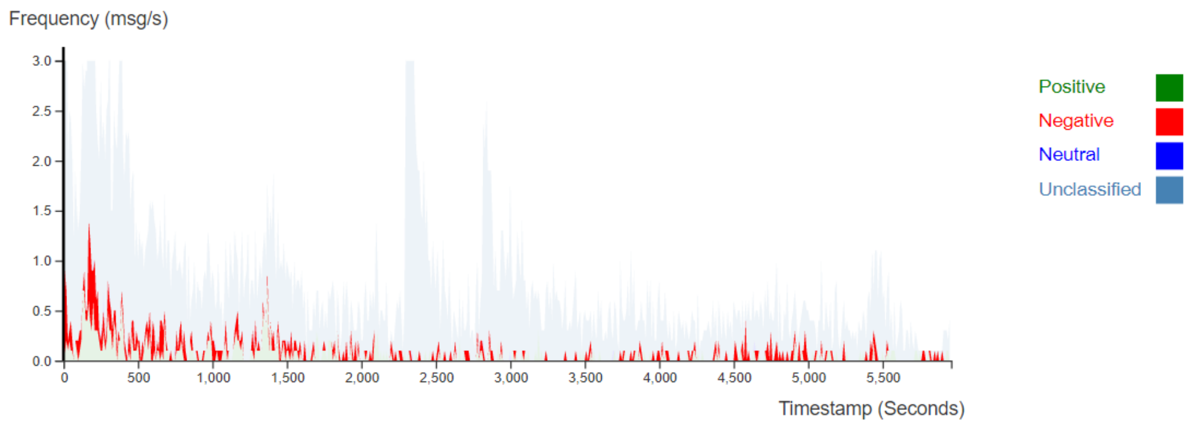


Figure 15 – Negative-selected Stacked-area Chart of the Nike Dataset

By looking at the stacked area chart generated by the website, we can get a better understanding of the results. Figures 9 through 17 represent the resulting graphs of the three datasets. Each dataset has three figures: the main graph containing all four variables, and two others, that isolate the negative and positive variables. By looking at the amplitude of each variable we are capable of drawing conclusions on the frequency of negative, positive or neutral messages. One of the first observations is the fact that the distribution of sentiments is not uniform across all three streams. Most 10 second intervals tend towards a positive or a negative distribution, rather than being evenly distributed. The places where sentiment distribution is in fact evenly distributed is during parts of the streams where there is downtime and no products are being revealed.

By looking at the our stacked-area chart, represented by Figures 9 through 17, multiple peaks of both positive and negative message frequencies can be identified. By making use of the website, we are able to get precise values of these peaks, analysed in Table 11, Table 12 and Table 13.

Table 11 – Positive & Negative Peaks of the BlizzCon 2018 Dataset

Positive Peaks	Timestamp Intervals	Pos. Freq. **(msg/s)**	Neg. Freq. **(msg/s)**	Total Freq. **(msg/s)**	Positive %
#1	[0:29:40, 0:29:50]	**6.9**	0	10.4	**66%**
#2	[1:18:30, 1:18:40]	**6.4**	0.2	9.3	**69%**
#3	[2:15:20, 2:15:30]	**6.1**	0.4	9.9	**62%**
#4	[2:17:20, 2:17:30]	**5.9**	0.2	9.2	**64%**
Negative Peaks	Timestamp Intervals	Pos. Freq. **(msg/s)**	Neg. Freq. **(msg/s)**	Total Freq. **(msg/s)**	Negative %
#5	[0:45:50, 0:46:00]	0.6	**7.2**	10.2	**71%**
#6	[2:34:40, 2:34:50]	0	**9.3**	10.2	**91%**

Pertaining to this dataset, Table 11 shows that the highest number of positive messages per second happened on an interval in the beginning of the stream. The model identified 6.9 positive messages per second on this peak (**Peak #1**) which amounted to 66% of the total number of messages recorded during that period. Another particularity of this peak is that the model did not identify a single negative message during this interval. Further analysis as to the context of this peak reveals that this interval is the exact moment that the countdown ended, and the opening ceremony began. The large quantities of positive messages can be justified by the excitement viewers felt as the product reveals were finally about to begin.

Peak #2 recorded the second highest positive message frequency and the highest number of positive messages relatively to the total messages in that time period. In addition, this interval only had around 2% negative messages. The context of this peak, after cross-referencing the same timestamp with the stream, relates to a moment during the opening ceremony where the CEO of the company hosting the event, made a verbal shout-out to all the people watching the stream at home. This resulted in an influx of positive messages, showing love and appreciation for the shout-out.

Peak #3 had the lowest positive message count, relative to the total number of messages during that interval out of all the positive peaks at 62 percentage points. In addition, it had the highest number of negative message frequency of the positive peaks at 0.4 negative msg/s. These values do not represent a significant change relative to the other peaks however the context of this peak is what makes it interesting. Analysing the video, this particular interval represents the first interval we analysed that followed one of the products revealed, “Warcraft 3 Remastered”. This product was an unannounced game and this spike in positive messages happened right after the trailer finished, which, as expected, are moments greeted with a large quantity of sentiment-loaded messages.

The time interval of **Peak #4** is only 1 minute and 50 seconds after Peak #3, this peak occurred during the same product reveal, “Warcraft 3 Remastered”. During this time period, the stream revealed a second trailer revealing more details about the game. Once more, the large quantities of positive messages can be attributed to the excitement and joy of the viewers in response to this product, and its trailer.

In terms of negative peaks, there were only two distinguishable peaks, the first (**Peak #5**) recorded 7.2 negative msgs/s out of a total of 10.2 msgs/s. This represented a total of 71% negative messages during that time period, recording only 0.6 positive msgs/s. This time interval corresponded to a moment before the opening ceremony began, when there was an interview with a special guest, Alex "Goldenboy" Mendez. Unfortunately, one of our emotes located in our emote lexicon, that was classified as a “**Sentiment Expression**” and “**Sentiment Enhancer**” type emote, is an emote made from a picture of this guest. As a result, once the interview started, viewers excessively sent messages containing this emote, not because they were trying to express a sentiment, but as “**Mirroring**” – one of the possible emote use-cases identified in section 3.3. This coincidence serves as a way to shed light on a flaw of our model, one which was considered irremediable.

Peak #6 was a particularly unusual peak across all the datasets. This interval was an outlier containing 9.3 negative messages per second. Totalling 91% negative messages during that time period alongside 0 positive messages. Further analysis reveals that this outlier was in fact not another technical problem related to our model, but rather an authentic response by the viewers. In this interval, Blizzard, the company hosting this stream, revealed another one of their products, “Diablo Immortal” which came as a negative surprise.

Table 12 – Positive & Negative Peaks of the Nike Dataset

Positive Peaks	Timestamp Intervals	Pos. Freq. **(msg/s)**	Neg. Freq. **(msg/s)**	Total Freq. **(msg/s)**	Positive %
#1	[0:02:20, 0:02:30]	**0.7**	0.2	3	**23%**
#2	[0:22:50, 0:23:00]	**0.8**	0.1	1.8	**44%**
Negative Peaks	Timestamp Intervals	Pos. Freq. **(msg/s)**	Neg. Freq. **(msg/s)**	Total Freq. **(msg/s)**	Negative %
#3	[0:00:10, 0:00:20]	0.1	**0.8**	6.3	**13%**
#4	[0:03:00, 0:03:10]	0.4	**0.8**	3.7	**22%**

Peak #1 pertains to the time interval corresponding to the beginning of the video reveal of Nike's new smart shoes. Comparatively, the number of negative messages was considerably lower than positive messages during this time period. It is important to point out that, as previously mentioned, due to the low number of messages per second, the type of communication shifts from mostly feedback messages to messages that cover a wide range of topics. Additionally, as this platform is mostly frequented by gamers or people tied to the area of technology, the product being launched possibly didn't resonate too well among the viewers. An assumption that originates from data analysis from this time interval, which reveals a lot of messages of confusion, messages that are out of the scope of our model. As a result, the number of positive messages during this interval only accounted for 23% of the total number of messages.

The second peak (**Peak #2)** contains the greatest number of positive messages per second out of every interval. Furthermore, the amount of positive messages amount to 44% of the total messages captured during that time period, a substantial increase from the 23% of the other peak. By analysing the time interval on the video and the accompanied chat messages, we can attribute this peak to an announcement made by the speaker, stating "All of you will be the first to try it out (the shoes)". This statement sparked enthusiasm in the chat, thinking the statement was directed at the viewers of the stream, when in fact it was directed at the audience members present at the venue.

The interval included in **Peak #3** contains one of the highest total message frequencies of its dataset and the highest negative message frequency, tied with #4. Analysing the causes for this peak revealed this was caused by a technical failure, causing the stream to crash. This resulted in large quantities of messages stating the issue, as well as messages portraying discontent that such event transpired right before the product reveal.

Finally, **Peak #4** also had the highest negative message frequency totalling 22% of this intervals' total number of messages. Notably, this interval also contained 11% positive messages. Suggesting a somewhat mixed reaction. Contextually, the corresponding moment in which this peak happened relates to the exact moment the shoe design was first revealed. This justifies the difference of opinions as some users reacted negatively, a smaller part reacted positively, and the majority of chat messages related to different reactions like confusion, disbelief, laughter, all of which are not captured by the model as they do not unambiguously carry a positive or negative sentiment.

Table 13 – Positive & Negative Peaks of the E3 2018 Dataset

Positive Peaks	Timestamp Intervals	Pos. Freq. **(msg/s)**	Neg. Freq. **(msg/s)**	Total Freq. **(msg/s)**	Positive %
#1	[2:12:30, 2:12:40]	**3.8**	1.7	8.8	**43%**
#2	[3:39:10, 3:39:20]	**4.8**	0.4	10.1	**47%**
#3	[6:49:50, 6:50:00]	**4.2**	0.4	9.4	**45%**
#4	[8:25:00, 8:25:10]	**3.9**	0.2	8.4	**46%**
#5	[8:41:00, 8:41:10]	**4**	0.4	11.5	**34%**
#6	[8:53:10, 8:53:20]	**4.9**	0.1	10.1	**48%**
#7	[9:12:10, 9:12:20]	**4.3**	0.4	9.6	**45%**
Negative Peaks	Timestamp Intervals	Pos. Freq. **(msg/s)**	Neg. Freq. **(msg/s)**	Total Freq. **(msg/s)**	Negative %
#8	[1:16:30, 1:16:40]	0.7	**4.8**	11.2	**43%**
#9	[3:30:30, 3:30:40]	0.1	**6.7**	8.6	**78%**
#10	[7:50:00, 7:50:10]	0.5	**5.2**	10.4	**50%**
#11	[8:07:30, 8:07:40]	0	**6.5**	8.1	**80%**
#12	[9:19:30, 9:19:40]	0	**6.4**	9.7	**66%**

The peaks with the biggest positive message frequency were **Peaks #2** and **#6** recording 4.8 and 4.9 positive messages per second, respectively. Both of these peaks had a similar total number of messages sent during their respective time period amounting to around 46% number of positive messages relatively to their totals. Contextually, **Peak #6** happened during the announcement trailer for a long-awaited game called "Cyberpunk", sparking a large amount of messages carrying a positive sentiment connotation, anywhere from happiness to excitement. Similarly, **Peak #2** also happened during another game announcement, this time, the trailer had just revealed "The Elder Scrolls VI", resulting in a swarm of positive messages.

As a matter of fact, with the exception of **Peak #3**, all of the positive message frequency peaks can be traced back to a product reveal or a release day reveal. These numbers therefore can be used as a good indicator for which products the audience is most excited about.

Peak #3 contains a 4.2 positive messages per second in that particular interval, classified by our model. However, further scrutiny of the data reveals the cause for this spike in positive messages is due to a somewhat ambiguous emote ("Kreygasm") which usually carries a positive sentiment connotation but is also rarely used as a response to something of the sexual nature. In this case, the emote was used for the latter, causing a significant inflation in the number of positive messages accounted for. The decision was made to keep this ambiguous emote as it is an important positive sentiment expresser as well as sentiment enhancer and the cases in which it is used in other forms are very few in comparison. Additionally, we found no feasible way to distinguish between the two cases, only by analysing the context can we understand its meaning.

As for negative peaks, **Peaks #9, #11 and #12** can be distinguished from the other two as they contain considerably higher negative message frequency with a lower total message frequency. Notably, 80% of the total messages sent during the interval of **Peak #11** were considered negative by our model and it didn't recognize any positive messages either. The data reveals that this timestamp corresponds to a product reveal that was ill received by the audience, feelings of discontent and awkwardness towards both the reveal, and the way it was revealed were the cause of these results. Similarly, analysing the context that could explain **Peak #9,** revealed another product reveal that was also met with negative sentiments. Conversely, however, the negative sentiments portrayed in the chat were not all directed at the product, but instead, words and expressions were also used as a reaction to the story behind the product trailer, which involved sad events.

4.2. Segmented Data by Product Release

By segmenting our data by the different products revealed, we can analyze and understand the different reactions targeted at these individual products, rather than the overall stream. We are able to rank the performance of each product reveal by the response of the audience. Additionally, by comparing our results with secondary data sources that rank the products revealed, we are able to get a sense of how our model performed.

4.2.1. Blizzard Dataset

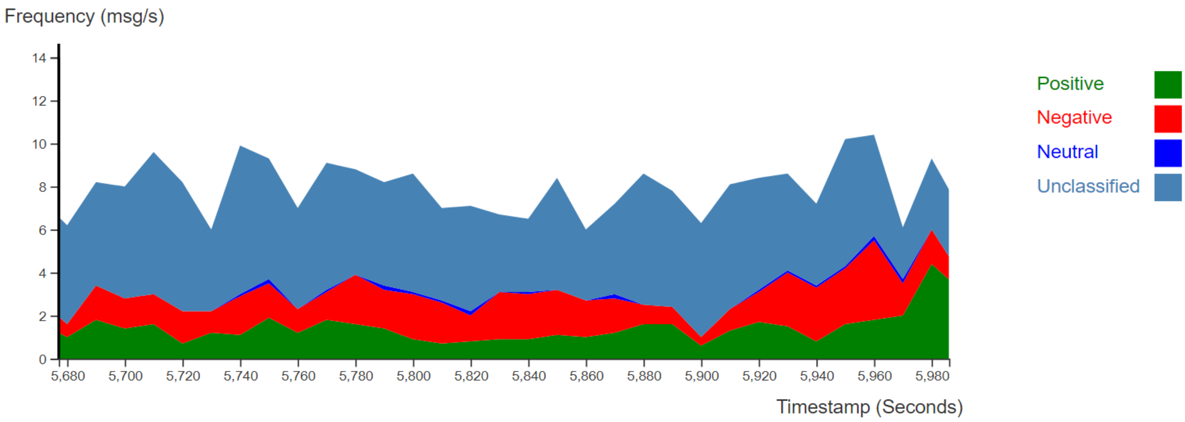


Figure 16 – Stacked-area Chart of the “Heroes of the Storm” Segment

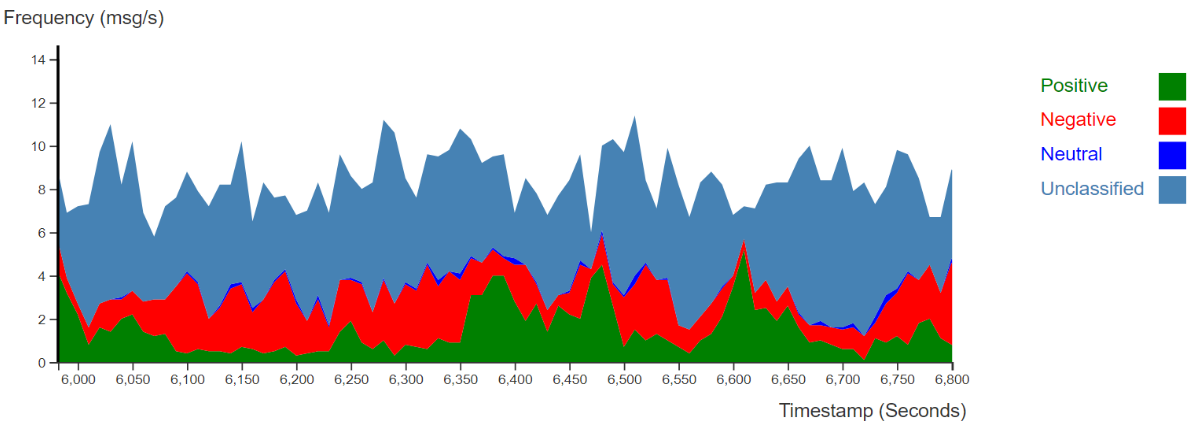


Figure 17 – Stacked-area Chart of the “World of Warcraft” Segment

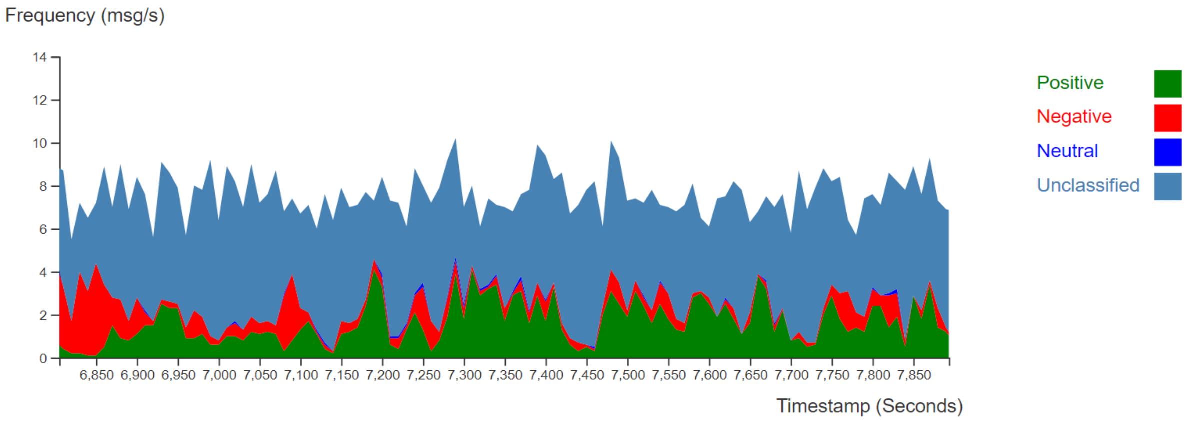


Figure 18 – Stacked-area Chart of the “Overwatch” Segment

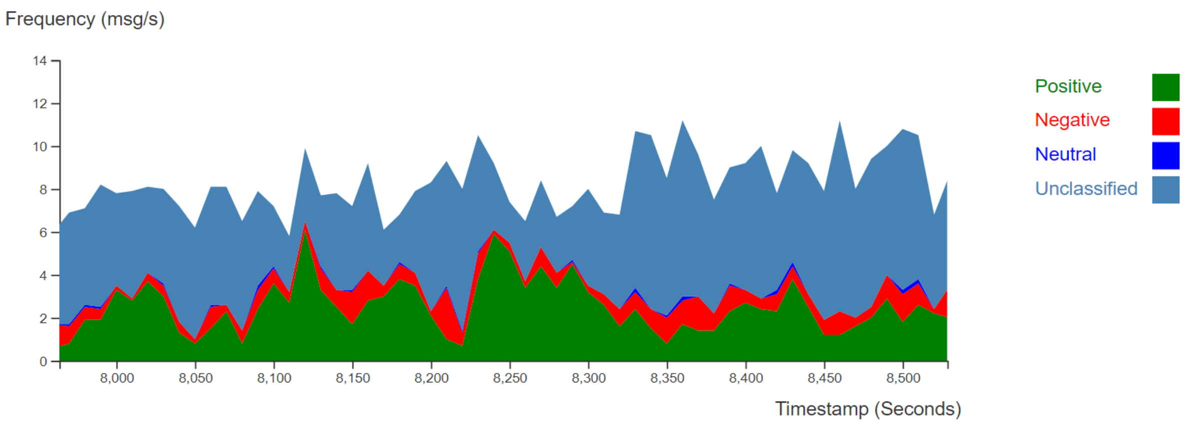


Figure 19 – Stacked-area Chart of the “Warcraft 3 Remastered” Segment

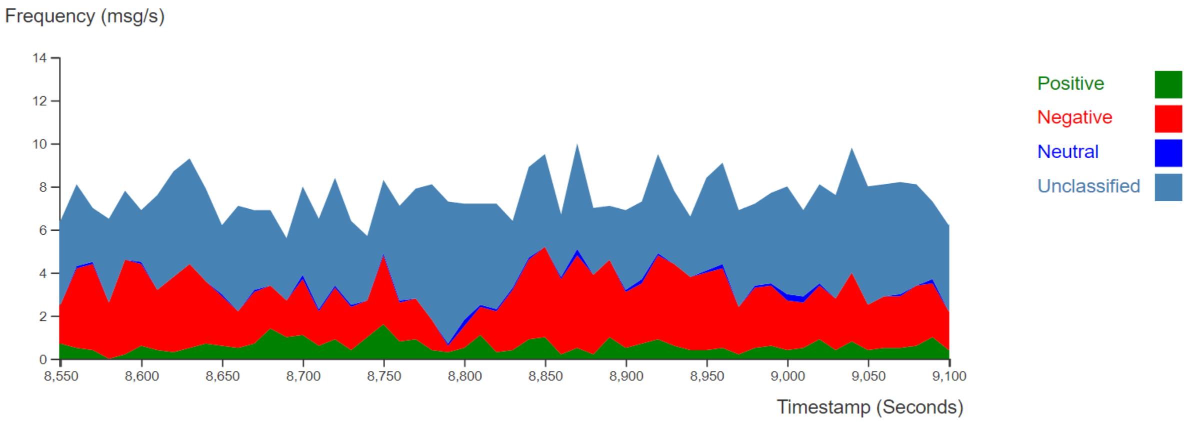


Figure 20 – Stacked-area Chart of the “Hearthstone” Segment

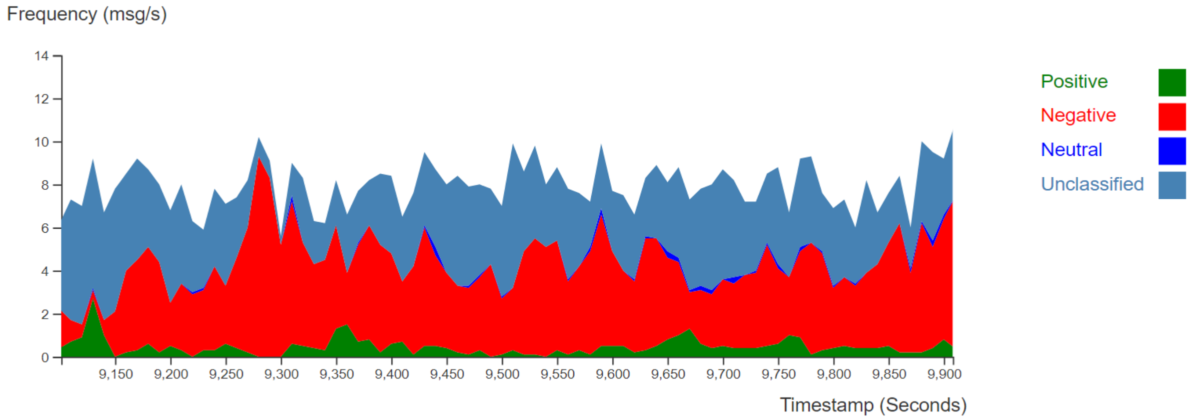


Figure 21 – Stacked-area Chart of the “Diablo Immortal” Segment

Table 14 – Blizzcon Sentiment Frequencies by Segment

Segment	Timestamp Interval	Average Pos Freq. (msg/s)	Average Neg Freq. (msg/s)
“Heroes of the Storm”	[1:34:21, 1:39:37]	1.196	1.403
“World of Warcraft”	[1:39:50, 1:53:10]	1.307	1.513
“Overwatch”	[1:53:34, 2:12:05]	1.540	0.653
“Warcraft 3 Remastered”	[2:13:13, 2:22:09]	2.397	0.607
“Hearthstone”	[2:22:25, 2:31:35]	0.488	2.574
“Diablo Immortal”	[2:31:35, 2:42:45]	0.353	3.907

The “**Heroes of the Storm**” segment was not a product announcement but instead, announced additions to an already existing game. As illustrated in Figure 16, the overall sentiment did not change much and stayed somewhat uniform throughout. This segment had a bigger number of negative messages than positive, and by looking at the data, this might be explained due to regular messages transmitting boredom or disinterest during the whole duration of the announcement.

The interval corresponding to the **“World of Warcraft”** segment, found similar positive and negative message frequencies, tending towards a more negative sentiment. This announcement was also not a product reveal but rather revealed the future and direction of an already existing game. Figure 17 reveals that the majority of negative messages happened between the interval [6070, 6360]. Further analysis of the content of this interval reveals that most of these messages were in fact false negatives: This game contains two major factions, and players can choose to play either one. These two factions are at war and because of this, every time one of the factions gets mentioned, a lot of negative messages aimed at the opposite faction show up. In this particular timeframe, the announcer is purposely inciting both factions which spawns an uproar of messages aimed at insulting the other faction. The model, lacking context, is not able to isolate which messages are targeted at what, as the messages themselves lack context.

The segment “**Overwatch**” had the second highest positive message frequency and the second lowest negative frequency of all the others. The two announcements made for this already existing game were well received across the segment, as depicted in Figure 18. The initial high number of messages can be attributed to a technical fault, causing the stream to become without sound, resulting in a stream of negative messages aimed at the production mishap.

"Warcraft 3 Remastered" was a surprise announcement, as they had not previously revealed this game was in the workings. The segment received the highest score of all the other segments, containing both the highest positive message frequency and the lowest negative message frequency. Throughout the entire segment (Figure 19), viewers posted messages that conveyed excitement, love and interest.

"Hearthstone", is an already existing card game that was revealing an expansion. This segment scored the second highest negative message frequency of all the other segments of 2.574 negative messages a second (Table 14 & Figure 20). By looking at the raw data as well as the video, it is clear this announcement was ill-received and the messages contained negative-loaded sentiments like boredom and disinterest.

The final segment, **"Diablo Immortal"** scored the lowest by a fair margin. Averaging 3.907 negative messages a second, and only 0.353 positive messages a second. The peak of positive messages, shown in Figure 21, can be explained by the excitement of the viewers, who expected a different kind of announcement, a continuation of the old "Diablo" series. Instead, as the announcement continued, they realized that the game was instead a mobile game. This led to an outrage effect, causing large amounts of negative messages throughout the segment. This outrage was so widespread, multiple media organizations posted stories about the fan backlash.

4.2.2. NIKE Dataset

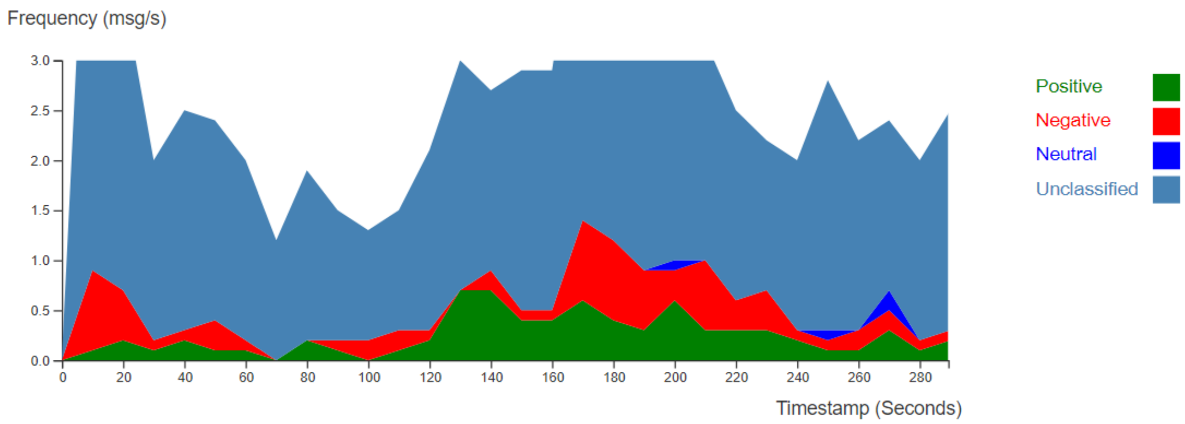


Figure 22 – Stacked-area Chart of the Nike Product Announcement Segment

Albeit not uniform, this segment pertaining to the part of the stream where the product (new smart shoes) were first introduced, had an almost perfect even sentiment frequency. Averaging 0.2633 positive and negative messages per second. The initial increase in negative messages, depicted in Figure 22, is due to technical difficulties causing the stream to crash, as previously mentioned. Later, the increase in positive messages from 120 seconds onwards, is a result of the build-up towards the product reveal. Following that, we see a rapid increase in negative messages right when the shoe design was first revealed. Eventually, the frequency of these negative messages was less common as the characteristics of this shoe were described.

4.2.3. E3 Dataset

The E3 Dataset contains chat logs of 9 hours and 22 minutes, corresponding to one day of an event that lasted 4 days and either announced or revealed 124 games. It would be a massive endeavour to segment and analyse each and every product. As such, in hopes of validating our model in the same way we have for our other datasets, the decision was made to pick 3 announced products that media platforms deemed were great announcements or bad announcements, and then segment those results to see how the model performed.

After carefully reviewing multiple online magazines (Cnet, 2018; Games Radar, 2018; GQ, 2018; The Verge, 2018), grading the best or "most promising" games revealed at E3, we chose two game reveals that had overwhelmingly positive reviews: Cyberpunk 2077 and Fallout 76. As for a product that received negative reviews, we found media sources naming the game "Gears Five" as a disappointment.

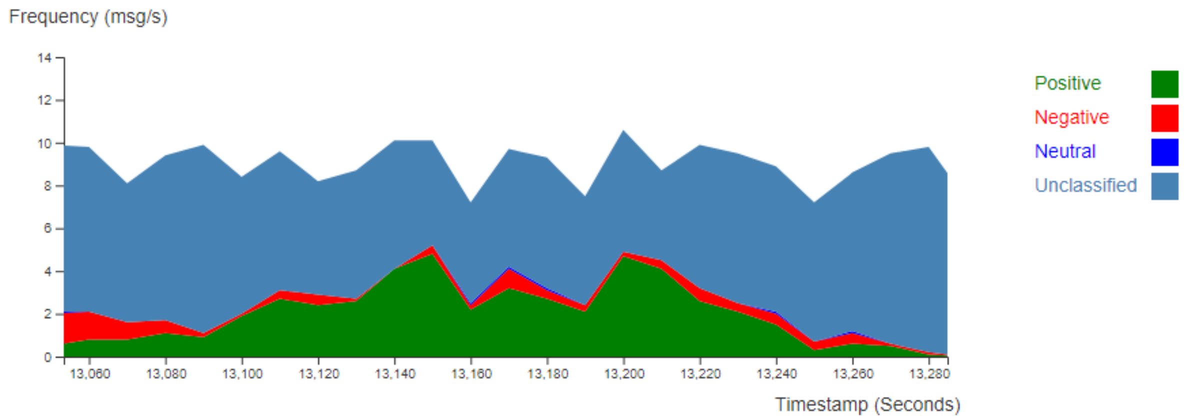


Figure 23 – Stacked-area Chart of the "Cyberpunk 2077" Segment

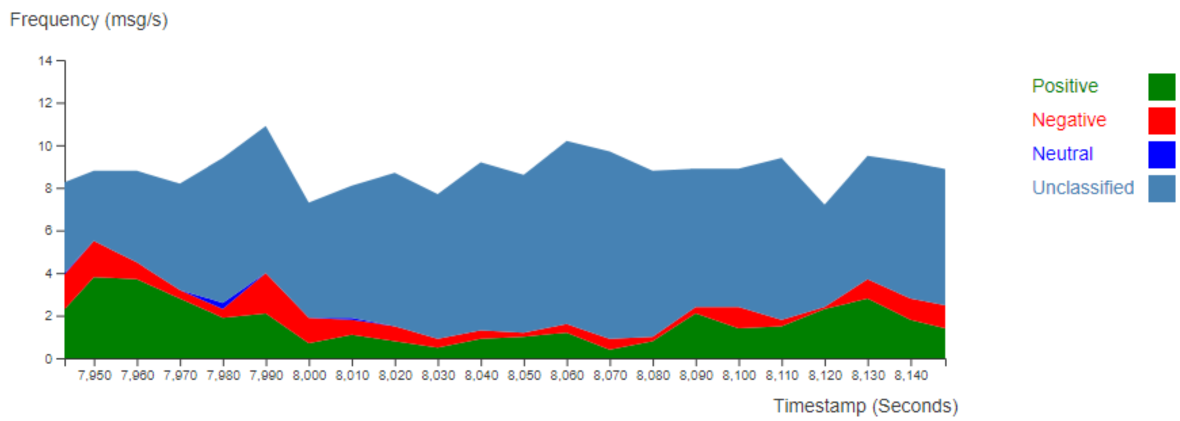


Figure 24 – Stacked-area Chart of the "Fallout 76" Segment

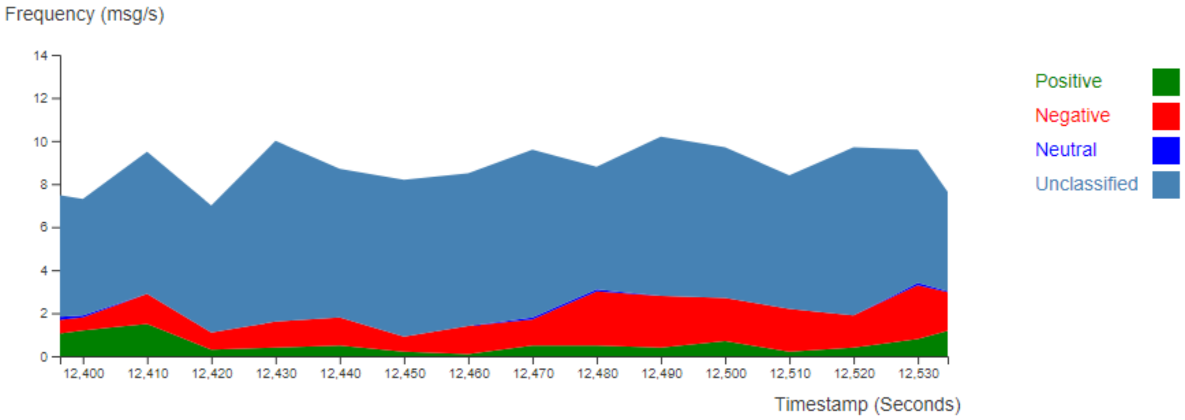


Figure 25 – Stacked-area Chart of the "Gears Pop" Segment

Table 15 – E3 Sentiment Frequencies by Segment

Segment	Timestamp Interval	Average Pos. Freq. (msg/s)	Average Neg. Freq. (msg/s)
"Cyberpunk 2077"	[3:37:33, 3:40:10]	**1.935**	0.465
"Fallout 76"	[2:12:23, 2:15:25]	**1.676**	0.719
"Gears Pop"	[3:26:36, 3:28:45]	0.566	**1.473**

With the aim of validating our model, the results were satisfactory. As Table 15 depicts, The two products that were expected to have positive feedback did in fact perform well above the average 0.741 positive messages per second of the entire E3 dataset. Upon further analysis, the time periods where negative messages increase, as shown in Figure 23 and Figure 24, can be explained by various extraneous factors related to the way the products were launched. I.e. the slow start of a trailer, causing viewers to express boredom.

The "Gears Pop" segment, portrayed in Figure 25, containing the product reveal that was expected to underperform did in fact, contain above average negative messages per second. Scoring 1.473 negative msg/s when the total average for the E3 set was 0.947 msg/s. Further analysis of the raw data and video reveals that the small number of positive messages displayed in Figure 25 can be attributed to the way the game was revealed. The fans of this company were expecting the release of long awaited "Gears 5" game. Instead, the company teased that they would be announcing "Gears 5" but announced "Gears Pop", a mobile game, instead. As such, the initial uptick in positive messages can be explained by the excitement fans displayed, thinking "Gears 5" would be released, only to be disappointed once they realized this announcement was in fact a mobile game.

5. CONCLUSIONS

The objective of this project was to develop a tool that allowed companies to tap into this new channel of communication and extract consumer feedback. This project, in an attempt to address this objective, built a comprehensive rule-based sentiment analysis model adapted to this specific channel of communication (Twitch), and built a platform that allowed the quick exploration and analysis of the output.

The analysis of results found in the previous section, show the model successfully managed to identify the correct sentiment conveyed by the users in chat, with only few exceptions. The results reveal this model is better adjusted for larger streams, rather than small ones, as the communication is more often feedback of what is being displayed or said on stream, rather than conversation between different users. Additionally, the results reveal that, while the model is aimed at specific situations like product reveals, the model performs satisfactory throughout the entirety of the streams.

By analyzing individual sections, such as product reveal segments, we were able to validate the model by cross-referencing the actual raw data, which is the extensive and exhausting process a company would have to go through without this tool, with the model results. As the results show, the final model iteration was successful in identifying the overall sentiment of the users in each segment, across all three different types of streams.

Furthermore, one of the goals of this thesis was to integrate the proposed model with a platform that could be used to explore the results. The website we developed, that can be adapted to any stream, was the main resource used for the analysis of results in the previous section, and was crucial for a complete visualization of results, reinforcing its usefulness.

We hope this project serves as a starting point for further research in opinion mining of livestream chats and, albeit far from perfect, we hope the tool developed and its results validate the usefulness and showcase the possible capabilities of a similar tool.

6. LIMITATIONS AND RECOMMENDATIONS FOR FUTURE WORKS

Over the course of the execution of this thesis there were clear barriers and limitations that both modified the initial intent of the project and caused multiple unexpected delays along the way. The first big barrier encountered early on had to do with the unavailability of software. Originally, the intent of this thesis was to be able to use the model we built to perform sentiment analysis on chat messages in-real-time, which would give instant feedback to the person or company that was streaming, as opposed to waiting for the stream to be over to perform this type of analysis. The software and hardware that allows the execution of this type of model in-real-time is already available but unfortunately, no licenses were granted. Such pushback lead to the decision of doing this type of analysis on old datasets, using software that was already licensed. However, the licensing on this software had expired and more delays followed until a proper license was issued.

Another clear hamstring relates to the unavailability of data. There has been little to no research done in the area of Sentiment Analysis, directed at livestream chats. This thesis therefore had to rely on very limited secondary data and come up with ways to generate primary data. In addition, we had to suggest our own methodology, adapted from past research in other areas such as tweets and reviews, as there were clear differences between the different areas of study.

Further on, the unavailability of an extractable complete database of emotes would also hinder the development of the project. Emotes play a big role in sentiment mining on Twitch, as such, the more sentiment-enhancement, sentiment-modification and sentiment-expression type emotes, the better the model. Unfortunately, a large quantity of these emotes had no feasible means of extraction as there isn't a proper resource allowing emote extraction in bulk.

The chat log data itself was also the cause of major hindrances. One of the biggest problems encountered related with the sheer size of our datasets. As the hardware was restricted to a personal computer - the processing power, memory and storage of the data caused challenges that had to be overcome. From pre-processing the large quantity of data to creating the necessary interfaces so that they could be fed as input to the different software used. Every step of the way, new challenges would arise which would cause multiple delays while trying to find a solution. Fortunately, despite the challenges, both the integrity and the totality of the data originally extracted were present throughout the entire project.

Finally, the utmost important limitation relates to the utility available on SAS Sentiment Analysis Studio. This software is outdated and a new software with more features is on the market. Unfortunately, this software was unavailable and as such, the software used had very little documentation and had a clear lack of functionalities that could have been used to improve the model significantly. As an example, in our rule-based model we had hoped to use rules that specifically target messages that consist of one word, as it is a common characteristic of a lot of messages in our dataset, since words, if isolated, may have a different meaning than if accompanied by other words. Unfortunately, there is no functionality that would allow this type of rule-setting and as such we had to forego some of these planned rules in favor of other rules that aren't as specific.

The following are recommendations for future works:

- Research in the linguistic aspects of chat rooms. How does the "shared-language" evolved overtime; How does the type of communication shift in large chats vs small chats; Building a more accurate lexicon of internet-slang as well as domain-specific slang.

- Using real-time technology. Such that these types of models may provide on-the-go feedback to streamers.
- Analyzing the external validity of these types of models. Due to the limitations referenced above, the number of channels analyzed were very limited. With more resources, these models can be tested across more channels with different characteristics as to understand how these types of models perform across different types of channels and at different times.
- Building upon the proposed framework and improving rule-based models, adapting the rules to this particular type of communication that is on the rise, using new technology that allows for more freedom in rule-building.

7. BIBLIOGRAPHY

Aggarwal, C. C., & Zhai, C. (Eds.). (2012). *Mining Text Data*. https://doi.org/10.1007/978-1-4614-3223-4

Becker, L., Erhart, G., Skiba, D., & Matula, V. (2013). *AVAYA: Sentiment Analysis on Twitter with Self-Training and Polarity Lexicon Expansion* (Vol. 2). Retrieved from http://leebecker.com/resources/semeval-2013

Briggs, D. (2018, December). "Ninja," the Fortnite streamer who's one of video gaming's biggest stars. *CNN*.

Cha, M., Kwak, H., Rodriguez, P., Ahn, Y.-Y., & Moon, S. (2007). *I Tube, You Tube, Everybody Tubes: Analyzing the World's Largest User Generated Content Video System*. Retrieved from http://an.kaist.ac.kr/traces/IMC2007.html.

Chevalier, J. A., & Mayzlin, D. (2006). The Effect of Word of Mouth on Sales: Online Book Reviews. *Journal of Marketing Research*, *XLIII*, 345. Retrieved from www.GlobalBooksinPrint.com

Cnet. (2018). The best and worst of E3 2018 - CNET. Retrieved October 8, 2019, from https://www.cnet.com/news/the-best-and-worst-of-e3-2018/

Dick Pritchett Real Estate. (n.d.). Eagle Cam - Live Feed. Retrieved May 10, 2019, from https://www.dickpritchettrealestate.com/eagle-feed.html

Facebook. (2016). Retrieved May 10, 2019, from https://www.facebook.com/marthastewart/videos/10154748445051289/

Farias, D. I. H., & Rosso, P. (2017). Irony, Sarcasm, and Sentiment Analysis. *Sentiment Analysis in Social Networks*, 113–128. https://doi.org/10.1016/B978-0-12-804412-4.00007-3

Feldman, R., & Sanger, J. (2006). *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data - Ronen Feldman, James Sanger - Google Books*. Cambridge University Press.

Ford, C., Tsaasan, A. M., Gardner, D., Nardi, B., Horgan, L. E., Rickman, J., & Liu, C. (2017). *Chat Speed OP : Practices of Coherence in Massive Twitch Chat*. https://doi.org/10.1145/3027063.3052765

Games Radar. (2018). The best, most exciting games at E3 2018. Retrieved October 8, 2019, from https://www.gamesradar.com/gamesradar-e3-2018-awards-the-best-most-exciting-games-at-e3-2018/

Ganapathibhotla, M., & Liu, B. (2008). Mining opinions in comparative sentences. *Proceedings of the 22nd International Conference on Computational Linguistics - COLING '08*, *1*, 241–248. https://doi.org/10.3115/1599081.1599112

Gandolfi, E. (2016). To watch or to play, it is in the game: The game culture on Twitch.tv among performers, plays and audiences. *Journal of Gaming & Virtual Worlds*, *8*(1), 63–82. https://doi.org/10.1386/jgvw.8.1.63_1

GQ. (2018). E3 2018: The most exciting games and announcements | British GQ. Retrieved October 8, 2019, from https://www.gq-magazine.co.uk/article/e3-2018-game-announcements

Guardian. (2016). Democrats stream gun control sit-in on Periscope after Republicans turn TV cameras off | US news | The Guardian. Retrieved May 10, 2019, from https://www.theguardian.com/technology/2016/jun/22/democrats-sit-in-periscope-facebook-live-gun-control

Guibon, G., Ochs, M., & Bellot, P. (2016). *From Emojis to Sentiment Analysis*. Retrieved from https://hal-amu.archives-ouvertes.fr/hal-01529708

Hackl, C. (2016). Now Streaming: Live Video as the Latest Tool for PR Professionals.

Retrieved May 10, 2019, from http://apps.prsa.org/Intelligence/Tactics/Articles/view/11455/1124/Now_Streaming_Live_Video_as_the_Latest_Tool_for_PR

Haimson, O. L., & Tang, J. C. (2017). What Makes Live Events Engaging on Facebook Live, Periscope, and Snapchat. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17*, 48–60. https://doi.org/10.1145/3025453.3025642

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 - CHI '14*, 1315–1324. https://doi.org/10.1145/2556288.2557048

He, W., Zha, S., & Li, L. (2013). Social media competitive analysis and text mining: A case study in the pizza industry. *International Journal of Information Management*, *33*(3), 464–472. https://doi.org/10.1016/J.IJINFOMGT.2013.01.001

Hu, M., & Liu, B. (2004). *Mining and Summarizing Customer Reviews*. Retrieved from https://www.cs.uic.edu/~liub/publications/kdd04-revSummary.pdf

Kaytoue, M., Silva, A., Cerf, L., Meira, W., & Raïssi, C. (2012). Watch me playing, I am a professional: a first study on video game live streaming. *Proceedings of the 21st International Conference Companion on World Wide Web - WWW '12 Companion*, 1181. https://doi.org/10.1145/2187980.2188259

Kim, S.-M., & Hovy, E. (2004). *Determining the Sentiment of Opinions*. Retrieved from http://delivery.acm.org/10.1145/1230000/1220555/p1367-kim.pdf?ip=46.189.134.254&id=1220555&acc=OPEN&key=4D4702B0C3E38B35.4D4702B0C3E38B35.4D4702B0C3E38B35.6D218144511F3437&__acm__=1557440567_baf4b382465a59180e35e08b0ed3b452

Kralj Novak, P., Smailović, J., Sluban, B., & Mozetič, I. (2015). Sentiment of Emojis. *PLOS ONE*, *10*(12), e0144296. https://doi.org/10.1371/journal.pone.0144296

Liau, B. Y., Tan, P. P., & Pei, P. (2014). Uncovering customer service experiences with Twitter: the case of airline industry. *Industrial Management & Data Systems*, *114*, 705–723. https://doi.org/10.1108/IMDS-07-2014-0225

Litvin, S. W., Goldsmith, R. E., & Pan, B. (2008). Electronic word-of-mouth in hospitality and tourism management. *Tourism Management*, *29*(3), 458–468. https://doi.org/10.1016/j.tourman.2007.05.011

Liu, B. (2012). Sentiment Analysis and Opinion Mining. *Synthesis Lectures on Human Language Technologies*, *5*(1), 1–167. https://doi.org/10.2200/S00416ED1V01Y201204HLT016

M. Ewalt, D. (2013). The ESPN of Video Games. *Forbes*.

Markham, S. K., Kowolenko, M., Michaelis, T. L., Markham, S. K., Kowolenko, M., & Michaelis, T. L. (2015). *Unstructured Text Analytics to Support New Product Development Decisions*. *6308*(April), 30–39. https://doi.org/10.5437/08956308X5802291

McInnis, N. (2016). Update: Chat Replay is Now Live! - Twitch Blog. Retrieved July 18, 2019, from https://blog.twitch.tv/update-chat-replay-is-now-live-the-official-twitch-blog-aac0b82305b6

Mosher, D. (2016). The internet just lost its mind over an exploding watermelon. Retrieved May 10, 2019, from https://www.businessinsider.com/watermelon-rubberband-explosion-buzzfeed-video-2016-4

Nasukawa, T., & Yi, J. (2003). Sentiment analysis. *Proceedings of the International Conference on Knowledge Capture - K-CAP '03*, 70. https://doi.org/10.1145/945645.945658

Nguyen Thi Ngoc, T., Nguyen Thi Thu, H., & Nguyen, V. A. (2019). Mining aspects of customer's review on the social network. *Journal of Big Data*, *6*(1). https://doi.org/10.1186/s40537-019-0184-5

Olejniczak, J. (2015). A linguistic study of language variety used on twitch.tv: Descriptive and corpus-based approaches. *Proceedings of Redefining Community in Intercultural Context (RCIC15)*.

Pak, A., & Paroubek, P. (2010). Twitter as a Corpus for Sentiment Analysis and Opinion Mining. *LREC*. Retrieved from https://www.semanticscholar.org/paper/Twitter-as-a-Corpus-for-Sentiment-Analysis-and-Pak-Paroubek/ad8a7f620a57478ff70045f97abc7aec9687ccbd

Pang, B., & Lee, L. (2004). A sentimental education. *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics - ACL '04*, 271-es. https://doi.org/10.3115/1218955.1218990

Purcell, J. . (2019). ReChatTool. Retrieved from GitHub website: https://github.com/jdpurcell/RechatTool

Raman, A., Tyson, G., & Sastry, N. (2018). *Facebook (A)Live? Are live social broadcasts really broadcasts?* https://doi.org/10.1145/3178876.3186061

Rebitzer, D. (2019). Twitch Leecher - The Broadcast Downloader. Retrieved from GitHub website: https://github.com/Franiac/TwitchLeecher

Scheibe, K., Fietkiewicz, K. J., & Stock, W. G. (2016). Information Behavior on Social Live Streaming Services. *Journal of Information Science Theory and Practice*, *4*(2), 6–20. https://doi.org/10.1633/JISTaP.2016.4.2.1

Selmer, Ø., Brevik, M., Gambäck, B., & Bungum, L. (2013). *NTNU: Domain Semi-Independent Short Message Sentiment Classification* (Vol. 2). Retrieved from http://tinyurl.com/twitscraper

Shaw, L. (2018, August). *Amazon Has YouTube Envy*.

Statista. (2019). • Global social media ranking 2019 | Statistic. Retrieved May 10, 2019, from https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/

StreamElements. (n.d.). StreamElements Chat Stats. Retrieved July 18, 2019, from https://stats.streamelements.com/

The New York Times. (2016). Critics of Police Welcome Facebook Live and Other Tools to Stream Video - The New York Times. Retrieved May 10, 2019, from https://www.nytimes.com/2016/07/08/us/critics-of-police-welcome-facebook-live-and-other-tools-to-stream-video.html

The Verge. (2018). The 25 best game trailers from E3 2018 - The Verge. Retrieved October 8, 2019, from https://www.theverge.com/2018/6/12/17451588/e3-2018-games-trailers-xbox-playstation-ea-bethesda-nintendo

Turney, P. D. (2002). *Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews*. Retrieved from http://www.google.com

Twitch. (n.d.). Twitch.tv - Partners. Retrieved May 13, 2019, from https://www.twitch.tv/p/partners/

TwitchTracker. (n.d.). Twitch Statistics & Charts. Retrieved May 13, 2019, from https://twitchtracker.com/statistics

van der Weijden, J. (2017). The Emotes of Twitch. *Diggit*. Retrieved from https://www.academia.edu/34978315/The_Emotes_of_Twitch_diggit_magazine

Weinberger, M. (2016). Amazon's $970 million purchase of Twitch makes so much sense now: It's all about the cloud. *Business Insider*. Retrieved from https://www.businessinsider.com/amazons-970-million-purchase-of-twitch-makes-so-much-sense-now-its-all-about-the-cloud-2016-3

Wiegand, M., Balahur, A., Roth, B., Klakow, D., & Montoyo, A. (2010). *A Survey on the*

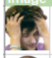
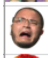
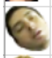
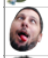
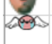
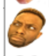
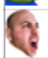
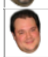
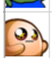
Role of Negation in Sentiment Analysis. Retrieved from http://www.cs.cornell.edu/people/

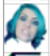
Wilson, T., Wiebe, J., & Hoffmann, P. (2005). *Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis*. Retrieved from http://www.cs.pitt.edu/

Wu, Y., & Jin, P. (2013). SemEval-2010 task 18: disambiguating sentiment ambiguous adjectives. *Language Resources and Evaluation*, *47*(3), 743–755. https://doi.org/10.1007/s10579-012-9206-z

Youtube. (2012). Official YouTube Blog: Mission complete: Red Bull Stratos lands safely back on Earth. Retrieved May 10, 2019, from https://youtube.googleblog.com/2012/10/mission-complete-red-bull-stratos-lands.html

8. APPENDIX

Image	Emote Code	Meaning - Meme - Feeling	SE	SM	SEN	M	C	Binary Feeling	Source
	NotLikeThis	Conveys frustration or disappointment.	X		X			Negative	Twitch
	WutFace	Used to express that the viewer is scared or disgusted.	X		X			Negative	Twitch
	KappaPride	Mostly used as mirroring for anything related to lgbt, but can sometimes also be used as a sarcasm indicator.		X		X			Twitch
	DansGame	Mostly used to convey disgust.	X					Negative	Twitch
	JeBaited	Most often used when the streamer or the viewers were lead to believe something that wasn't true.				X	X		Twitch
	LUL / OmegaLUL / GIGALUL / LULW	Mostly used to express laughter, whether it is a reaction to something funny or simply laughing at something/someone.	X						Twitch and the alternatives ones are from third-party entities
	ResidentSleeper	Used when the viewer feels like the stream is boring, carrying a negative connotation. In rare situations, it may also be used as mirroring when someone appears sleeping on the stream.	X		X	X		Negative	Twitch
	FailFish	Conveys the same meaning as the physical gesture of putting one's hand to their face. Used to convey frustration, embarassment or disappointment, often in face of some failure.	X		X			Negative	Twitch
	BrokeBack	Often used to describe something as irrational, carrying a negative connotation.	X		X	X		Negative	Twitch
	PogChamp / Pog / Poggers / PogU	Used to translate excitement.	X		X			Positive	Twitch and the alternatives ones are from third-party entities
	Kreygasm	Used to convey excitement. Sometimes it is also used when something of the sexual nature appears on the stream.	X		X	X		Positive	Twitch
	AngelThump	Sometimes used to convey tears of joy, other times to pay respects to someone or something that has died. Has a bigger positive connotation than "BibleThump"	X				X	Positive	Third-Party Entities
	BlessRNG	Translates to praying.					X		Twitch
	BibleThump	Conveys sadness without a negative connotation. Sometimes used as tears of joy mimicking "AngelThump"	X		X				Twitch
	EleGiggle	Is most often used as a sarcastic laugh.				X	X		Twitch
	cmonBruh / HYPERBRUH	Users use this emote when reacting to something that is deemed as racist.					X		Twitch
	monkaS / monkaX / monkaW	Signifies the viewer is scared	X						Third-Party Entities
	SwiftRage	Often used to show anger or to indicate that the message should be read as a shout.	X		X				Twitch
	KKona	This emote is used as a way to mrror any of the following things that may happen on stream: Country Music - Redneck - Incest				X			Third-Party Entities
	OpieOP	Used to express something related to fat.				X	X		Twitch
	ANELE	A mirror-type emote that reflects anything related to the arab ethnicity that shows up on stream.				X			Twitch
	Clap / HYPERCLAP	Translates to applause.	X		X	X		Positive	Third-Party Entities
	FeelsGoodMan / FeelsAmazingMan	Conveys feelings of happiness.	X		X			Positive	Third-Party Entities
	D:	Used to express a loud gasp.	X			X			Third-Party Entities
	HYPERS	Used to convey Excitement.	X		X			Positive	Third-Party Entities
	GivePLZ	Signifies that the viewer wants whatever is shown/spoken on stream.	X		X			Positive	Twitch
	:tf:	Used to indicate sarcasm.		X					Third-Party Entities
	:)	Can be used to signify happiness, but in Twitch chat its more often used in a sarcastic tone.	X			X			Twitch
	:D	N/A	X		X			Positive	Twitch
	:(	Mostly used to convey sadness or disappointment	X		X			Negative	Twitch
	:O	Used to express surprise.	X						Twitch
	FrankerZ / OhMyDog	A mirror-type emote that reflects when a dog is shown/heard on stream.				X			Twitch
	CurseLit	Used to express that something is great - amazing.	X		X			Positive	Twitch
	TriHard	A mirror-type emote that reflects anything related to the black ethnicity that shows up on stream.							Twitch

 VoHiYo	Mostly used as a greeting but may also be used to mirror anything related to anime				X	X		Twitch
 VoteNay	Often used as a sign of rejection, a way of rejecting something. Another common use is a way to answer negatively to a question posed on stream.	X		X		X	Negative	Twitch
 VoteYea	Often used as a sign of approval, a way of accepting something. Another common use is a way to answer positvely to a question posed on stream.	X		X		X	Positive	Twitch
 BabyRage	Mostly used to signify complaint or whining. But may also be used to mimic a streamer's whining.	X			X			Twitch
 4Head / YouDontSay	The usage of this emote varies a lot depending on context, it may be used when someone states the obvious or to acknowledge a weird laugh.				X	X		Twitch
 FeelsWeirdMan / WeirdChamp	Mostly used in reaction to something the viewer thought was weird or awkward.	X				X	Negative	Third-Party Entities
 FeelsBadMan / pepeHands	Used to express sadness or empathy towards something sad.	X		X			Negative	Third-Party Entities
 pepeLaugh	Laugh but more towards a person then LUL/OmegaLUL	X		X				Third-Party Entities
 Kappa / Keepo / Kapp	Used as an indication of sarcasm.		X		X			Twitch
 CoolStoryBob	Often used when the viewer thinks a story being told is a lie. May also be used to indicate the story is too long.					X	Negative	Twitch
 HotPokket	Used to either mirror a person who's "triggered" or as a reaction to something sexist that was said.				X	X		Twitch
 MrDestructoid	Most often used when anything related to robots is either said or shown on stream.				X	X		Twitch
 SMOrc	Is used when the stream references or shows an Orc. Sometimes used to express "head first".				X	X		Twitch
 HeyGuys	Used as a greeting.					X		Twitch
 <3 / bleedPurple	Mostly used as a way to express love - support.	X		X			Positive	Twitch
 PJSalt	Used to reflect the streamer's frustration after losing, or to express that the viewer is upset at something.	X			X		Negative	Twitch
 SeemsGood	Translates to the physical gesture of thumbs up	X		X			Positive	Twitch
 haHAA	Conveys the feeling of cringe. Carries a negative connotation.	X		X	X		Negative	Third-Party Entities
 MingLee	A mirror-type emote that reflects anything related to the asian ethnicity that shows up on stream.				X			Twitch

Appendix 1- Emote Lexicon