Twitch Chat Visualization

Visualization: https://clrizy.github.io/infoviz-twitch-chat/

Info247 - Information Visualization and Presentation - Spring 2021

Team: Keming(Alex) Gao, Claire Zhu

Project Goals / Background

Our project focused on analyzing sentiment on Twitch livestream chats. Twitch is a livestreaming platform on which streamers play video games while viewers watch and post messages in a “chat” on the side.

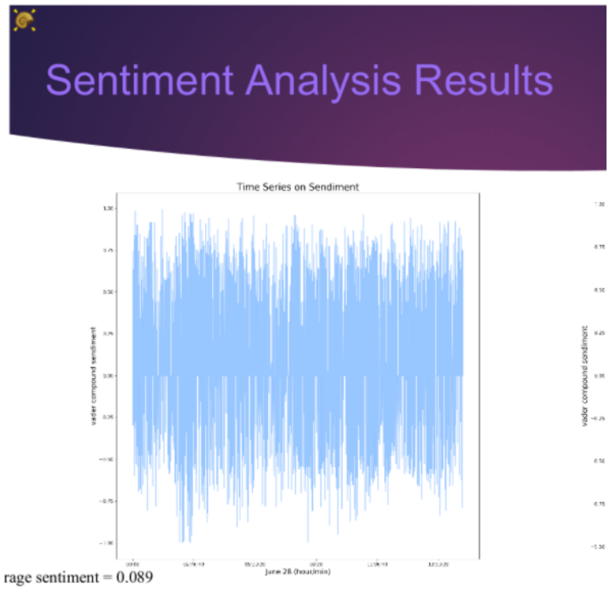
Of particular interest was (1) visualizing and explaining how sentiment analysis may be applied to Twitch chat (2) isolating events in a stream that could lead to shifts in sentiment, and (3) comparing different streamers based on the general attitudes of their stream chats,

This space poses unique visualization and data collection challenges. Twitch streams (especially for individual streamers) use highly specialized language, and an individual stream can contain hundreds to thousands of messages every minute. This means the individual data points are essentially impossible to resolve, and it needs to be visualized in aggregate / at the macro level. However, sentiment analysis is best understood at the micro-level. Users need to understand what a “sentiment value” means at the level of an individual message, not in aggregate. Therefore, we have an overarching goal (4) to explore techniques for visualizing data at different scales to give users a holistic understanding of the information.

Discussion of related work

Analyzing Sentiment in the Twitch Jungle, by Michael Lin

This was a basic visualization of sentiment over time in a time series chart. However, this visualization faced some serious legibility issues, as we can see below:



Due to the high density of data, the time series was essentially impossible to process. A popular Twitch streamer is typically receiving dozens of messages every second. In a 2 hour time period this corresponds to tens of thousands of data points, which looks incomprehensible on a time series chart. Second, each individual message is potentially very different from the messages before and after it. In a single Twitch chat, dozens of users individually decide to send whatever message they want at a given second; 99% of the time, they are not responding to prior messages, but just independently reacting to whatever is happening on the stream. Thus, though the data is ordered by time, each message can fluctuate wildly; on a time series chart, this looks like thousands of jagged zig-zag edges rather than a curve. The paper above demonstrates the weakness of a naive time series visualization of the chat.

Mining User Feedback from Livestream Chats, by Jaime Tese Entrega

This paper analyzed Twitch sentiments expressed throughout a particular event (Blizzcon 2018) and plotted them as a stacked-area chart. An example is displayed below:

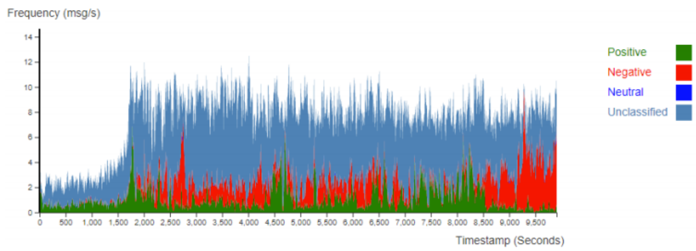


Figure 7 – Complete Stacked-area Chart of the BlizzCon 2018 Dataset

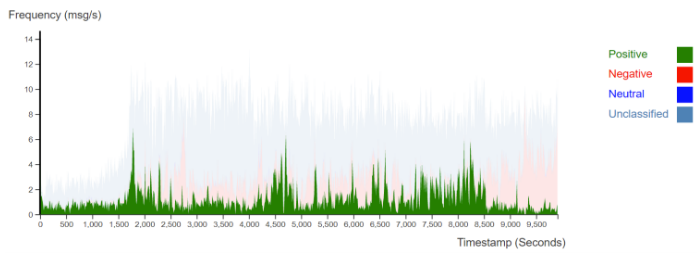


Figure 8 – Positive-selected Stacked-area Chart of the BlizzCon 2018 Dataset

We isolated numerous issues with these representations that we attempted to address with our visualizations. First, this graph faced the same issue as the previous work (too high density). Second of all, it was clear that we needed better ways to compare positive against negative messages. Though the "area" represents the total quantity of positive and negative messages, a user is ostensibly less interested in the individual areas and more interested in *comparing* the areas of positive/negative/neutral messages. However, when the stacked area chart is so dense, it is very hard to compare ratios. Additionally, the color scheme used is potentially confusing; red and blue are typically considered at opposite "ends" of the color spectrum, but red and green have metaphorical meanings (e.g. stop and go at a traffic light). We explore a more intuitive color scheme in our visualizations.

Emote-based sentiment analysis on comments of popular Twitch.tv channels, by Kobs, Zehe, and Bernestetter et. al

This paper was primarily a methods paper focused on developing new sentiment analysis methods. While we could not replicate their exact methods, we took inspiration from the following visualization in the paper (the only one presented):

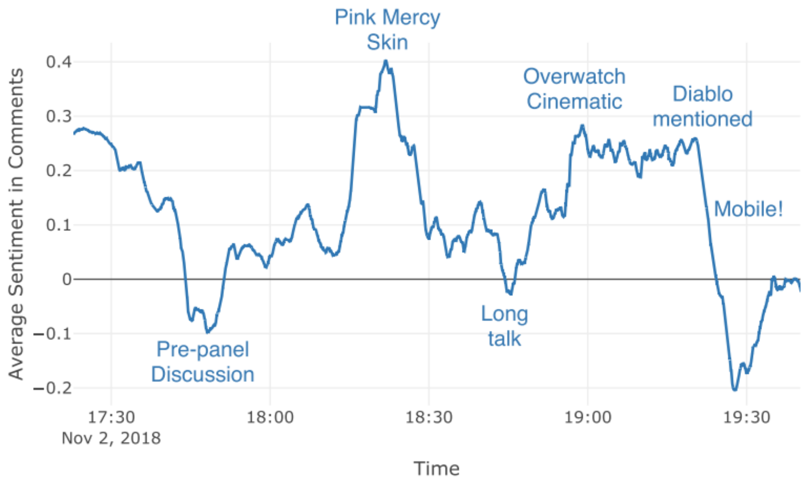


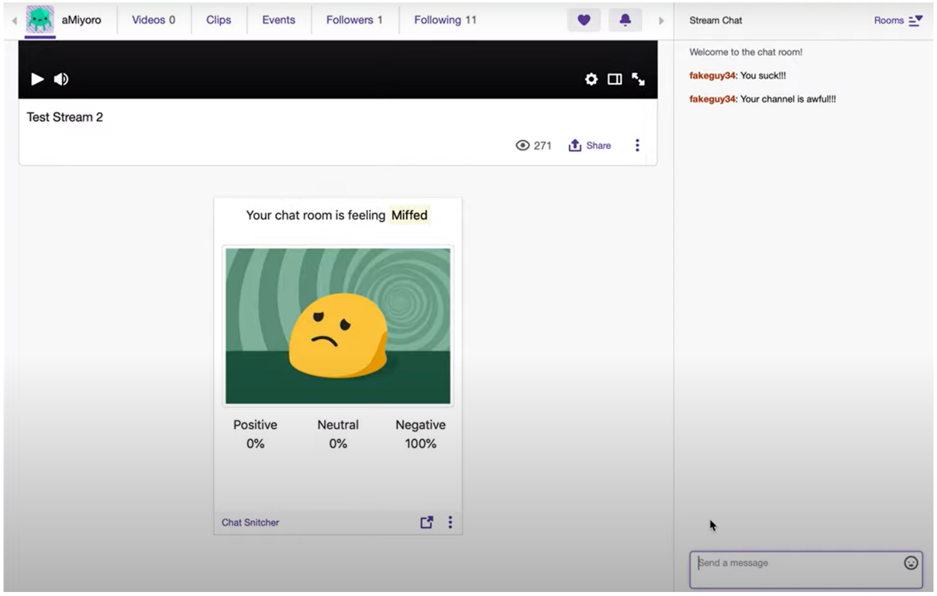
Fig. 10. Sentiment trajectory of the keynote presentation of BlizzCon 2018.

This was an analysis of BlizzCon 2018, the same event discussed in the paper prior. However, it makes a lot of smart choices - it limits the window of time to only two hours so it's less overwhelming, it utilizes a moving average so the time series appears less "spiky", and it annotates key events in the timeline so the user can associate abstract notions of sentiment with tangible events.

Though we can look at the graph and see "generally positive" sentiment, comparisons of *how much* positive and negative sentiment are hard to see. Though we can see drops in sentiment below the highlighted "zero" line, it is hard to get a feel of the ratio to positive/negative at a glance. We think using an extra channel, such as color, may help people grasp this faster.

Twitch Chat Snitcher, by Anthony Miyoro

This work uses a real-time updating visualization to great effect. There are a few takeaways that we took from the project (a screencap from the demonstration video is below).

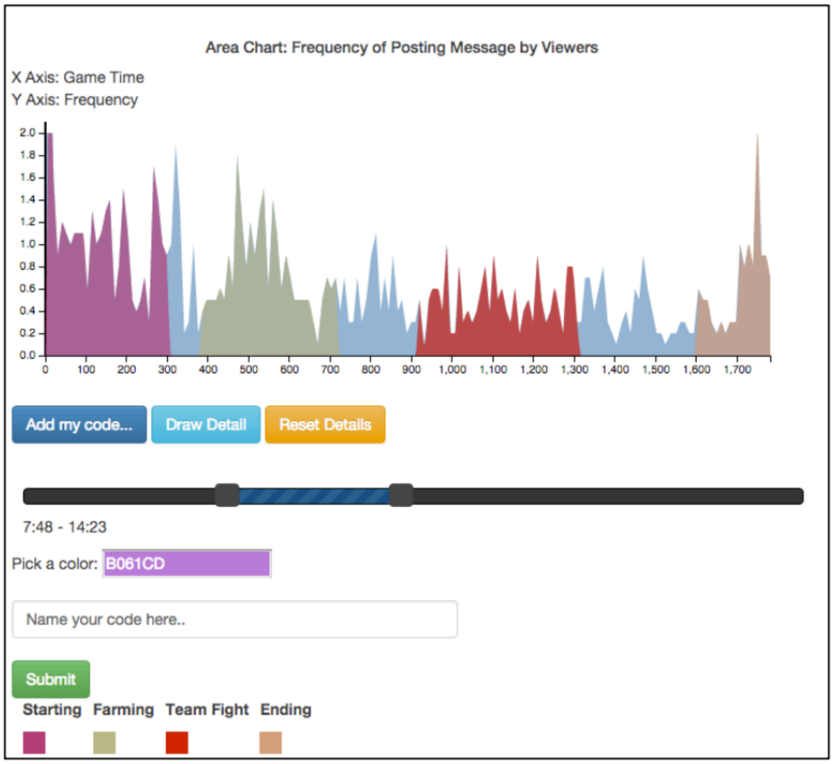


First of all, we note how the positive, neutral, and negative values are placed horizontally against each other. This is easier to process than, say, “neutral | positive | negative”, because people implicitly think about a spectrum of values (think a number line) from one extreme, passing through neutral, to another. We use this insight in our introductory visualization.

Second of all, we note the educational effect that real time updates have. Each message typed in the chat causes the visualization to update; it allows the user to see exactly how an individual message’s content is categorized with a sentiment, and allows the user to contextualize the message against the overall chat. In contrast with the first three visualizations, where each data point is abstract and not truly “legible” to the user, this is easy for a regular person to understand. This was a major inspiration for our first visualization.

TwitchViz: A Visualization Tool for Twitch Chatrooms, by Pan et. al

The following visualization gave us more insights on how to potentially highlight “events” over the time of a Twitch stream. While the earlier annotated visualization allowed pointing out specific individual “points” in the stream, this visualization is concerned with “intervals”, as seen below:



This is potentially useful because the viewers posting chats aren't responding only to individual events, but to events that take place over long time periods. The four colors on the bottom (starting, farming, team fight, ending) correspond to in-game "periods", throughout which users may have a heightened sense of positivity/negativity (for instance, the team fight stage tends to be more exciting). We took inspiration from this visualization to try highlighting or potentially brushing periods of time in a time series rather than annotating individual points (ultimately, however, it didn't look particularly good).

TwitchChat - A Dataset For Exploring Livestream Chat

This work gave us insight on how to clean and process the chat data itself. Though we could not access the actual dataset mentioned in the paper, we used some of the methods described in the paper - for instance, using twitchemotes.com to access information about the different emotes/slang messages, and also how to clean and process tokens in preparation for deeper analysis.

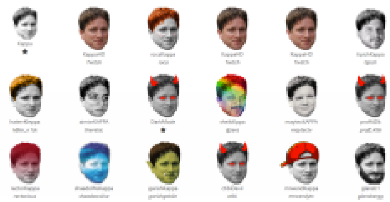


Figure 2: Variations of 'Kappa' (top left) collected using the 'View Similar Emotes' feature on https://twitchemotes.com (accessed 08-10-2019).

Table 2: Dataset size (in terms of number of tokens and number of unique tokens) before and after data cleaning stages.

Metric	Raw Dataset	After Stage 1	After Stages 2 & 3
Tokens	61,040,692	47,783,915	38,751,630
Unique Tokens	1,658,055	1,405,084	10,011

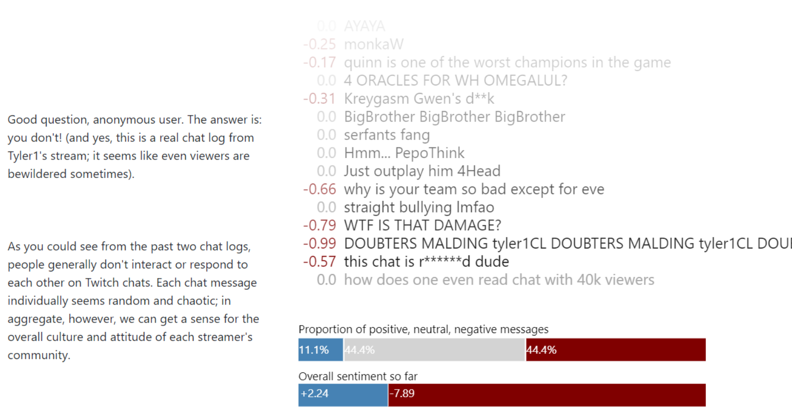
Description of visualization

To accomplish the goals above, we used three distinct sections. To form a more cohesive experience, they are ordered as follows: (1) an interactive introduction to Twitch chat and sentiment analysis, (2) a look into the events driving different sentiment expression over the course of a 2 hour stream, and (3) a look at the different aspects of sentiments expressed in each stream.

Section 1 - An Interactive Introduction

This section is intended to give users a low-level introduction to the overall topics of the visualization - Twitch chat and sentiment analysis. This is presented in a static screenshot below, but it is in reality an *interactive scrolling visualization*. As the user scrolls down the page, more messages pop up - each tagged with a specific sentiment value - and explainers emerge on the side pointing out elements that contributed to the assigned value. The messages are color-coded based on their sentiment, and their scores remain color-coded as the user scrolls so they can pick out positive/neutral/negative messages at a glance.

More importantly, as the user scrolls and sees how each individual message is tagged, there are two stacked columns which dynamically update with the overall proportion of positive/negative/neutral messages and the proportion of pos/neg sentiment. The bars are strategically placed on top of each other to show how quantities of messages don't necessarily align with impact. For instance, when the user scrolls to a single extremely negative message, they can see how it swings the overall sentiment against many slightly positive ones.



This allows users to *connect the micro with the macro*, to show how each individual message affects the overall qualities of the chat in aggregate. By the end of this section, the user will have an understanding of what types of messages are sent in Twitch chat, how an individual message gets tagged with a sentiment, and how each message relates to the overall sentiment.

Section 2 - Sentiment Over Time

Armed with a low level understanding, users can now begin to think about Twitch Chat in aggregate and ask what in-stream events affect sentiment at larger scales. This is presented through an annotated time series chart where certain peaks and valleys are annotated with specific in-stream or in-game events.

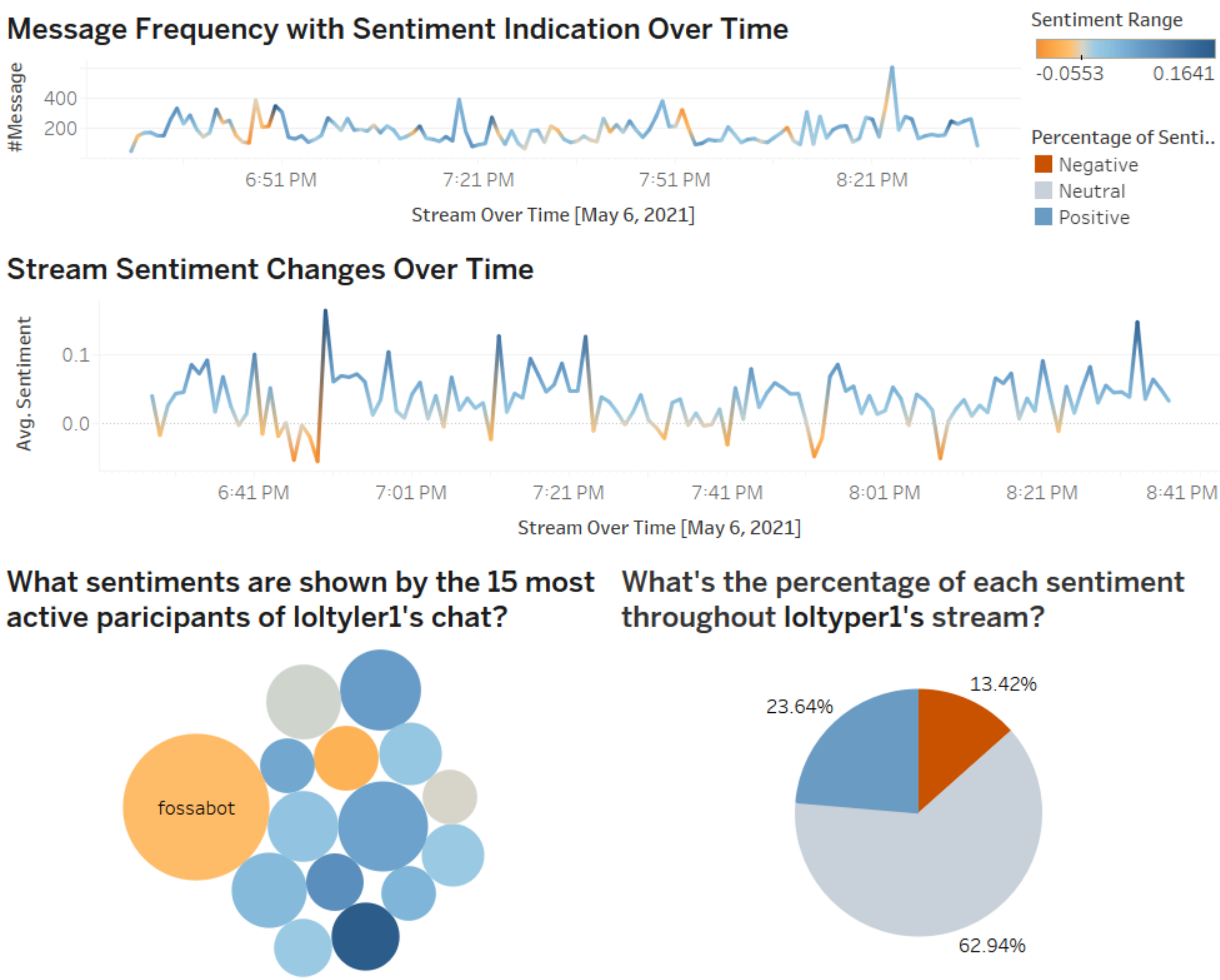
However, visualization problems emerge because of the scale/context of the data (this is explored in depth in the critiques of the "related work"). Thus, a data transformation is required to visualize the trends that occur in the aggregate. After experimenting with a few methods, we decide to take a rolling average; we average 100 messages at a time, and slide this "window" of size 100 across the length of the stream, to obtain a smoother signal that tracks the general trend.

To keep things consistent, we reuse the same color channels used in the Interactive Introduction. Rather than only use the y-axis to communicate the value, we also use color. This is done because time series charts are great to communicate trends and up and down movements, and the color adds the extra ability to see quantity at a glance ("how much red is there? = how much negative sentiment?).

Section 3 - Aspects of Sentiments in a Stream

After logging the data and key events, additional EDA was performed for visualizing different aspects of sentiments for individual streamers, as well as the streamer group as a whole using Tableau. We had a few questions in mind as the general direction of our EDA. For example, since our streamers of choice have such different perceived personas, content, and styles, how are those differences reflected in sentiments? What are some of the key events and elements that drive changes and collective actions in Twitch chat? How about some interesting breakdowns that we're able to acquire based on sentiment analysis?

After data cleaning and reorganizing, we performed EDA and tried various visualizations in Tableau. Those visualizations were then filtered and selected to meaningful visualizations that could best accomplish our dataviz goal, which is to inform the viewer and allow them to explore what they've previously unaware of chat sentiments. Screenshot of streamer loltyler1's compiled visualizations as an example here:



We were also able to identify the straightforward shifts in message frequency and mood shift of the chats based on those visualizations, and by isolating those events and cross-reference with our previous event log, we were then able to connect the dots of stream events with chat mood, and finalized on which elements or element categories actually had influences on chat moods.

As mentioned above, color-coding is one of the most important elements in our visualization design. By keeping the brick red - negative / grey - neutral / blue - positive channels consistent, viewers were able to draw clear comparison of chat sentiments and breakdowns across three streamers,

Data used (including data transformation methods)

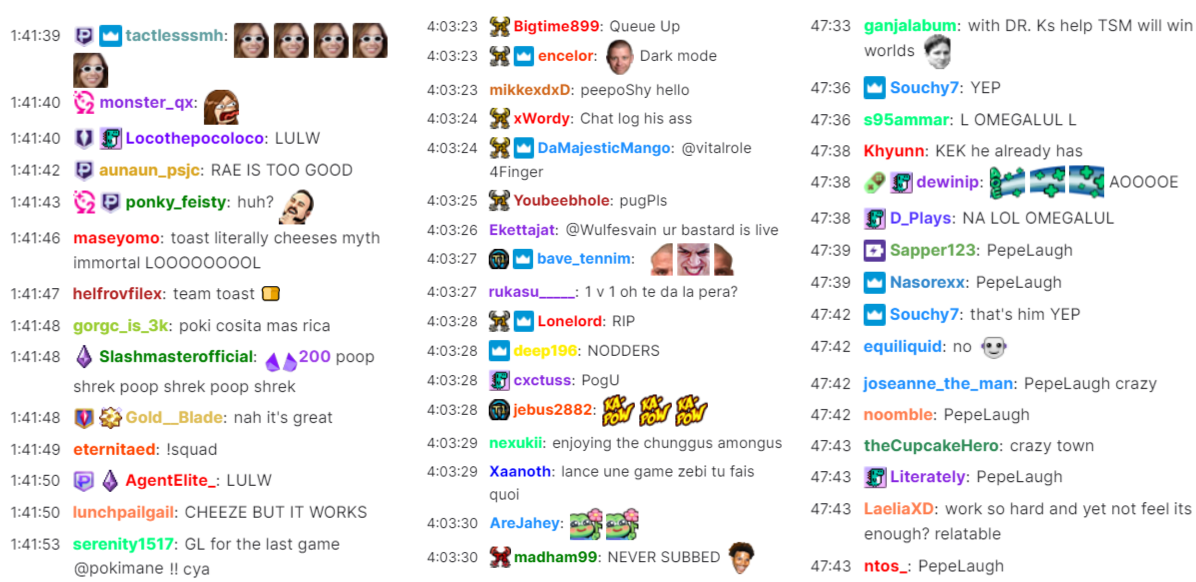
Retrieval

We scrape the data from three different Twitch chats. We choose three streamers that appear to have diverse types of communities and focuses (Tyler1, who is a 'hardcore gamer' type; Pokimane, a woman who is known to have a more 'positive' community; and HealthyGamerGG, a Harvard Medical School psychiatrist who focuses on mental health and personal deveopment) and scrape their logs for two hours.

Privacy and consent matters were considered; we applied for a Twitch API authorization token which allows us to scrape and use chat. All chat is publicly (and permanently, if the streamer saves a video-on-demand version of the stream) viewable in the first place.

Sentiment Analysis Data

We had to construct some of this data manually. Twitch chat uses unique slang that is not found anywhere else on the internet. There is generalized slang such as "SourPls, monkaS, poggers", etc. and individualized streamer slang such as "pokiW, pokiShy, pokiBan" that are known as 'emotes'. This is important because, unlike regular conversational text, such emotes and slang are actually one of the primary ways that people communicate on Twitch chat and express their sentiments.



We collect information about each streamer's commonly used emotes/slang from TwitchEmotes.com as well as internet searches about common Twitch slang. Then, we assign a human sentiment score to each slang term based on our prior knowledge of how that term is used, and hack our sentiment analysis tool a bit to include this data (more details in the tools section). We then run the tool on each of our messages to get a sentiment score for each message.

In [33]:
```python
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
```

In [34]:
```python
posEmotesSlang = ['pokihype', 'pokiw', 'pokipride', 'pokiwater', 'pokint', 'pokiwow', 'pokiya', 'pokidrool',
                  'pokio', 'pokiheart', 'pokifree', 'pokisub', 'pokigift', 'pokikek', 'pokigood', 'pokiaww',
                   'pokihey', 'pokishy', 'pokih', 'pokil', 'pokihypers', 'pokimoney', 'pokiflex', 'pokiuwu',
                   'pogu', 'pog', 'poggers', 'pogchamp', 'tyler1ssj', 'tyler1bb', 'tyler1ayy',
                   'tyler1good', 'tyler1champ', 'tyler1pride', 'drhgpog', 'drhghypers', 'drhgsmile', 'drhgok', 'drhgpepega',
                   'kappapride', 'kreygasm', 'clap', 'hyperclap', 'feelsgoodman', 'hypers', 'giveplz'
                  ]
```

In [35]:
```python
negEmotesSlang = ['sadge', 'monkas', 'pokiree', 'pokibruh', 'pokipuke', 'pokiknife', 'pokiban', 'pokigun', 'pokiyikes',
                  'pokicry', 'pokiweird', 'weirdchamp', 'monkaw', 'monka', 'monkas', 'inting', 'malding',
                  'tyler1ban', 'tyler1sleeper', 'tyler1bruh', 'tyler1monk', 'drhgweird', 'drhgehh', 'drhgwtf', 'notlikethis',
                  'wutface', 'dansgame', 'residentsleeper', 'failfish', 'brokeback', 'swiftrage', 'babyrage', 'feelsweirdman'
                  'feelsbadman', 'pepehands', 'pjsalt', 'hahaa']
```

Annotation and Further Processing

For our time series visualization, we needed to collect information about in-stream events to relate to the sentiment shifts in the chat. We had to watch the 2 hour streams and manually annotate timestamps of interest and note what might have driven sentiment up or down. These timestamps were then used to annotate the time series charts.

From there, we first made some edits and changes of our data set after importing the files into Tableau:

- Some data points were removed at the beginning and the end of the streams if they had a drastic impact on the viz;
- Numerical field [Compound] was coded into categorical field [comp] with negative, positive and neutral tags based on the numerical value, to better present the breakdown of different sentiments.

For the introductory visualization, we needed to manually filter out messages that were too profane or offensive and cut down on spam as needed.

Tools used to accomplish goals

Data scraping and processing

We used Python to scrape the data from Twitch, primarily based off this tutorial (https://www.learndatasci.com/tutorials/how-stream-text-data-twitch-sockets-python/). We then parse it into a CSV file (see the github repository link for the code, specifically the Chat Logger and Chat Parser notebooks)

For sentiment analysis, we found a sentiment analyzer explicitly trained for online/social media text - the Vader Sentiment Analyzer (https://github.com/cjhutto/vaderSentiment). We then hacked it a little bit to adapt the analyzer to Twitch by altering the lexicon that it draws from to include the slang data we collected earlier (see the github repository link for more details, specifically the Sentiment Analysis notebook). Then, we run the Vader tool on every message scraped.

Section 1 / Interactive Introduction

This was done with d3.js, jQuery, and standard HTML/CSS/JS, based slightly off a tutorial created by Jim Vallandingham (https://github.com/vlandham/scroll_demo). The tutorial is outdated (uses d3 v3, which is extremely different from d3 v6); most of the code and methods used are far different, and the tutorial serves only as a general paradigmatic guide. See the Github Repository for more details.

Section 2 and 3 / Time & Event Series + Sentiment ?

Other than manually annotating the events, EDA visualizations for those two sections were both done with Tableau. Chat logs with sentiment analysis were imported into the software, and changes and edits were made with the calculated fields feature in Tableau. Data brushing was made possible with the Brush Filter extension that was available on the Extension Gallery.

Website

The website was built with HTML and Bootstrap5 with elements from a template by TemplateMo (https://templatemo.com/tm-561-purple-buzz).

Results Obtained [Usability Testing]

We ran a usability test with three participants. We particularly wanted to see how these visualizations can increase user understandings of:

1. What might be considered toxicity in livestream chats
2. What kinds of in-game events cause livestream chats to increase or decrease in toxicity
3. General problems and complexities surrounding live streaming

Methods

Participants: we selected 3 individuals who have pre-existing knowledge of livestream/game culture. One is a 23 year old male who watches Twitch streams almost every day and initially “strongly understands” Twitch culture. One is a 27 years old female who only watches “just chatting” content once or twice a week and initially “somewhat understands” Twitch culture. Our last participant is a 28 year old male who watches Twitch streams almost every day, and sometimes also streams himself, and initially “strongly understands” Twitch culture.

Scenarios/tasks: we used a combination of survey, observation, and interview. The survey tasks are described further below in “test measures”. As for observations, we asked users to go through each section of our visualization, verbalize their thought processes as they interpreted the visualization, and ask questions if there was something they did not quite understand. After observing them interact with the visualization, we re-asked the same survey questions as before and questions about the overall experience with the visualization in a semi-structured interview.

We present 4 questions prior to using the visualization and afterwards. The questions are as follows.

1. List some factors that can contribute to a toxic chat culture on Twitch.
2. List some events in a stream that may affect the overall mood of a chat on Twitch.
3. What are some difficulties you perceive with managing chat culture as a streamer or moderator?
4. Rank your self-perceived understanding of Twitch chat dynamics, from 1 (none at all) to 5 (intimately familiar).

In judging the responses to these questions, for the first two questions, we focused on whether their answers included the factors presented in our visualizations. For the third question, we looked to see if the information they retained was “useful” (are they just remembering facts, or do they have some sense of what those facts might *mean?*). For the final question, we looked for increases in the reported value.

We also present 3 questions post-visualization about the usability of the visualizations:
1. What visualization did you like the most and why?
2. Was there anything you found particularly difficult to understand? Why?
3. Do you have any other suggestions for improving the visualizations?

Results

pre-visualization:

	User 1	User 2	User 3
List factors contributing to a negative chat culture	Bullying the streamer, coded misogyny/racism/homophobia, spam	Bullying the streamer, misogyny, racism, homophobia, spam, bullying other chat members	Cursing, arguing with each other, just being mean in general
List events that affect chat mood	In-game events, streamer's own mood, streamer personas, donations/hype trains/chat events	In-game events, streamer's mood, streamer interactions with the chat	In-game events
List difficulties of managing chat	Banning people, needing lots of moderators, needing bots and automated filters, distinguishing whether something is "good" spam or undesirable spam	Moderator team often making subjective decisions, chat goes really really fast so things slip through the cracks, some streamers tolerate or even want "edginess"	Need a moderator team, probably a lot of trolling that happens
Self perceived understanding (1-5, 1 being none at all)	5	3	2

Post-visualization (we are only recording the *additional* things said that weren't already said pre-visualization)

	User 1	User 2	User 3
List factors contributing to toxic chat culture	Bad gameplay, slurs	In addition to everything said, talking about controversies can draw negative attention	Bad gameplay can be a key factor, especially for female streamers
List events that affect chat mood	"whatever the chat mood is now can affect the mood later?"; when a stream ends	Nothing new compared to pre-visualization	Some very vocal toxic commenters may affect other viewers' mood
List difficulties of managing chat	For some moderators maybe they want to wait till a couple of people report on the same toxic person to ban that guy. Don't just wanna ban everyone who says something negative	Hard to make an instant judgement	"Seems a bit like a runaway train"; Very difficult to pay attention to everything
Self perceived understanding (1-5, 1 being none at all)	5	4	2.5
Describe differences between the three streamers presented.	T1 is the worst "but I'm not surprised"; "wish I could see the chats and see how they're different".	Not really sure whether this is a consistent trend or just a result of the sample, but feels like Pokimane is clearly more positive and the rest are not	Pokimane seems nice and really approachable as a new viewer compared to the other two; "Honestly I could just guess these results from how their faces looked"

Discussion

In general, our survey results seemed positive; users were picking up on new factors that we discussed in our visualizations. However, there were some key usability incidents in which our users asked us questions about what something visualized, and we modified our visualization on that basis.

First, we presented another chat log for the users to compare, based on feedback that they would've liked to see more individual messages to get a better understanding of Twitch.

Second, we improved the narrative flow of our visualization project as a whole. We describe why we chose certain streamers to study, because it seemed that users wanted to understand the general visualization - sort of like an "intro paragraph" - before trying to process the visualizations individually. We also place the scrolling chat visualization *before* the aggregated statistics of sentiment across streamers, since the scrolling chat helped them connect individual messages to quantitative measurements and clarify what the high-level aggregated statistics are actually measuring.

Finally, we improved the narrative text used in the interactive visualization and standardized the color palettes used across the entire set of visualizations. These changes were made just for ease of interpretation and to help users through things they pointed out were confusing.

Links

Git Repository: https://github.com/Clrizy/infoviz-twitch-chat

Tableau Public for the project:
https://public.tableau.com/profile/yuan.zhu4969#!/vizhome/twitch_16199950231110/p-1

Viz - Hosted on Github: https://clrizy.github.io/infoviz-twitch-chat/

Related works (in order of appearance)

- Analyzing Sentiment in the Twitch Jungle https://drive.google.com/file/d/1RgTqCa1gdgiZmZUYHGNUDlaUOfVb7sP_/view
- Mining User Feedback from Livestream Chats https://run.unl.pt/bitstream/10362/95285/1/TGI0288.pdf
- Emote-based Sentiment Analysis https://dl.acm.org/doi/10.1145/3365523
- Twitch Chat Snitcher https://devpost.com/software/twitch-chat-snitcher#updates
- TwitchViz: A Visualization Tool for Twitch Chat http://clab.iat.sfu.ca/pubs/Pan-TwitchViz-CHI2016.pdf
- TwitchChat: A Dataset For Exploring Livestream Chat https://www.charlieringer.com/files/Ringer_AIIDE_2020.pdf

Tools used

- Streaming Twitch Chat Data https://www.learndatasci.com/tutorials/how-stream-text-data-twitch-sockets-python/
- VADER Sentiment Analysis https://github.com/cjhutto/vaderSentiment
- Scrollytelling Guide https://github.com/vlandham/scroll_demo
- Tableau Brushing Extension https://extensiongallery.tableau.com/extensions/112?version=2021.1&per-page=200
- Website template https://templatemo.com/tm-561-purple-buzz

Work Distribution

Tasks	Alex / Keming	Claire
Data Retrieval	10	90

Sentiment Analysis	100	0
Event Annotation	0	100
Website Construction	50	50
Interactive Visualization	100	0
Tableau Visualizations	0	100
Usability Testing	60	40
Final Report	70	30