Visualization of Chat Interactivity in Twitch and YouTube VODs

Ahmed Awan
University of Maryland, Baltimore County
Baltimore, Maryland
qe66653@umbc.edu

Jack Bitzel
University of Maryland, Baltimore County
Baltimore, Maryland
jbitzel1@umbc.edu

Jace Hull
University of Maryland, Baltimore County
Baltimore, Maryland
jach1@umbc.edu

Siven Odenwald
University of Maryland, Baltimore County
Baltimore, Maryland
sivenl317@umbc.edu

***Abstract*—The goal of this paper is to demonstrate the information that can be gained from observing chat interactions over the course of a past live stream, in particular, the use of emojis. The problems addressed in this document include discovering timestamps for moments of interest within past live streams by visualizing chat interactions during the live stream.**

***Keywords*—** Twitch, YouTube, Live Stream, VOD, Video on Demand, Chat Interactivity, Visualization

I. Introduction

Content creators have begun to popularize the idea of turning live content, particularly from live streaming services such as Twitch and YouTube, into YouTube videos as a way to double the revenue generated from the same content. In order for content creators to turn a past live stream into a video, it requires re-watching a live stream to figure out which segments may make for interesting content. This can be time consuming, especially for longer broadcasts which may have several moments of interest. Reviewing broadcasts to find interesting moments can also slow down the time to turn a past live stream into a video. Generally, interesting moments from live streams generate higher chat interaction, similar to a crowd reacting to a comedian's performance. The funnier the joke, the more the crowd may laugh. Chat interactions follow this same idea. In the next section, we will discuss how we can simplify the process of extracting potential highlights from past live streams in order to facilitate an easier method of transitioning live stream content into YouTube videos.

II. Background

A. Inspiration

Our inspiration in creating this visualization is to analyze an individual YouTube or Twitch stream and to use raw metrics to quickly find highlights in a stream, active participants in a stream, and common chat usage amongst chat members. Doing so will allow streamers and content creators to easily find how their chats behave in individual streams and find highlights to turn live streams into edited reels to upload at a later date. Another reason for creating this is that while many methods of Twitch and YouTube live stream visualization exist, almost none are for analyzing individual stream moments. Most visualizations, such as *TwitchMetrics* and *TwitchTracker*, focus on long-term metrics like channel growth, total hours watched by viewers, and viewership milestones [3][4]. Instead of focusing on long-term data to support sustained channel growth, we want to analyze individual videos for key moments.

B. Related Work

In preparation for this project we found several similar works that aimed to visualize emote usage. However, several of them, such as *EmojiViz*, are more for eye candy than for finding trends and drawing conclusions from the data. EmojiViz shows real-time emote (emoji) usage from Twitter and displays the most popular emotes in the form of a ranked spiral. EmojiViz only shows total emote usage and the time reel only shows the last 30 seconds or so [1]. With our project we would like our visualizations to lead to more meaningful conclusions, such as typical peak times during a VOD. The inspiration for this Twitter emoji visualization came from Matthew Rothenberg's *emojitracker*. Created in 2013, the emojitracker was designed as both an online site and a physical installation at Eyebeam Art and Technology Center in January 2014 [7]. Again, this visualization is more an art exhibit than a statistical tool, showing only total emoji usage. Another website, called *TwitchMetrics*, takes a higher level view of content creators channels by analyzing followers gained, subscribers gained, and average number of viewers across multiple streams. It also records average viewership when playing certain games or activities [3]. Overall, this website provides an excellent, high-level perspective on the status of a content creator's channel. While this website provides valuable information about whether or not a particular content creator is growing, it provides little insight into individual stream metric information other than viewer count. Similarly to TwitchMetrics, *TwitchTracker* also takes a high-level approach to visualizing various channel metrics for Twitch streamers. This high-level information includes followers gained, average viewers, hours watched, subscriber count, and several other channel metrics [4]. This information is useful to keep track of the channel's status over time, such as if a channel is growing in popularity. However, TwitchTracker lacks the ability for users to view community interaction throughout past live streams.

There are some tools to analyze short-term VOD data in existence, however they are relatively lightweight or not in use. GitHub user ofietze's twitchVodAnalyzer is a tool that grabs scrubs through a chat and returns a text file displaying metrics such as top 20 used emotes, average emote usage per 10 minutes, and a "hypescore" calculated from adding positive points for exciting emotes (specifically the emotes PogChamp and Pog and the acronym "DPS") and subtracting points for emotes indicated boredom (specifically the emotes ResidentSleeper and GoatS, GoatS being a third-party emote used by the FrankerFaceZ browser extension) [10]. This tool operates on a similar purpose to what we wish to accomplish with our tool, in analyzing a specific VOD's chat and emote usage to gather data about it, however the twitchVodAnalyzer is limited in scope and lacks any visual data that could better chart nuanced changes in emote usage, rather focusing on data over the entire stream. Another tool, is GitHub user LasTechLabs's Twitch Vod Chat Analyzer, which graphically represents the percent change in total messages over an adjustable interval [11]. The Twitch Vod Chat Analyzer focuses on nuanced change in chat activity, which can be used to gauge notable events in a VOD, and allows for adjustable intervals to make the chart more precise or more legible if needed. However, this tool only focuses on total chat messages whereas we would also want to analyze more data from the chat, such as emote usage and subscribed user metrics. Additionally, the Twitch API this is built upon will be decommissioned on February 20th, 2022, preventing the tool for working past this date.

C. Purpose

With our project we hope to be able to display VOD chat data that could be informative to content creators in a way that is specific to their community and that could be constructive in increasing community interaction. We also want to provide a simple way for content creators to analyze how their individual streams may have been perceived by their audience. In using the visualizations, one should be able to analyze how chat interactions operate over the course of a given stream, including how chat interactions may vary between streams (seeing how different games/streamers/lengths affect chat activity); what commonalities there are in individual chat interactions (such as emotes used at the same times as other emotes); what specific limitations are set on chat (for example, subscription-only mode or emote-only mode); and how chat interaction correlates to events occurring on stream, such as finding stream highlights, raids of viewers, or large donations through chat activity. By creating these visualizations, we hope that content creators are able to quickly identify moments of interest, dictated by their community, that may suffice as well received content in YouTube video form. Also, we hope content creators are able to effectively identify what content drives more community interaction in their live streams.

III. Implementation

A. Design

The design of our website, available at , was for the homepage to be a blank page with a single text-input area and a submit button. The homepage will ingest the user input and submit the query to our RESTful API. The API will use Chat-Downloader to fetch the chat logs from the queried VOD, and return the flattened chat metadata as a JSON to the requester. Once the website receives the results from the query, it will feed the data into several functions that will draw the visualizations on the homepage beneath the search box. Each query the user makes should be cached to prevent overloading our API. The user should be able to query both Twitch and YouTube VODs.

B. Visualizations

We chose to present a number of simple visualizations rather than trying to combine multiple layers of data into a single visualization. Given that our data can be generally described as some change in a variable over time, the most logical choice of visualization type was the line graph. The following visualizations were generated from several publicly available Twitch VODs, specifically the streams used in Fig. 9 [9].

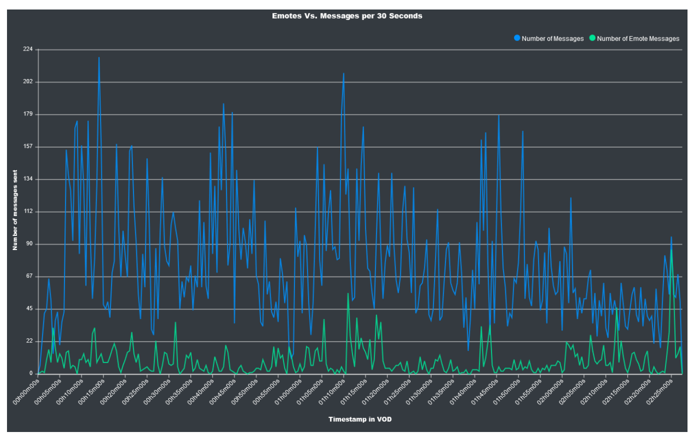

Fig. 1. Shows the number of total messages being sent over time separated into bins, defaulting to 30 second bins. The graph measures chat messages in two separate charts - one which measures messages that contain emotes and one which measures messages that do not contain emotes.

Fig. 1 displays the total number of messages sent during a stream divided into bins of a user-input length, defaulting to 30 second intervals and allowing input from 30 seconds to 3600 seconds (1 hour). The y-axis provides the total count of messages in a given interval and the x-axis is divided into equally-sized bins from the beginning to the end of the VOD. While it is a simple visualization, it provides an immediate glimpse as to how a chat engages with a given stream throughout through emote usage and overall chat activity.

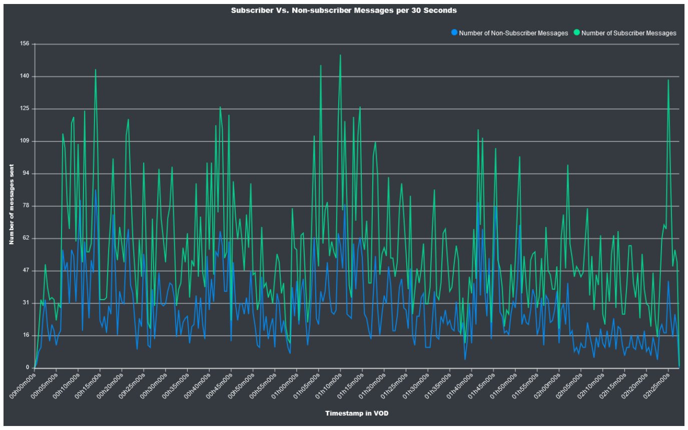

Fig. 2. Shows the number of total messages being sent over time separated into bins, defaulting to 30 second bins. The graph measures chat messages in two separate charts - one which measures messages sent from subscribed users and one which measures messages sent from unsubscribed users.

Fig. 2 is similar to the first in design, but displays chat messages sent by subscribed chatters and non-subscribed chatters. This visualization provides the user with a view of the amount of chat interactions being generated by viewers who are willing to pay a monthly subscription to watch the streamer, versus chatters who do not pay. With this visualization, the user is also able to toggle off either the subscribed chatters variable or non-subscribed chatters variable from the visualization. This allows the user to get a clearer picture of the particular variable they are interested in.

Both Fig. 1 and Fig. 2 measure number of messages across time, and in doing so lumps all chat messages under a given interval into a group. This interval length is provided by a text box input on the website. The default size is 30 second intervals, however the program supports intervals from 30 seconds to 3600 seconds (or 1 hour) in length. Fig. 3 shows a comparison between 30 second intervals and 120 second intervals using data from

Fig. 4 focuses on showing the 10 most active chatters throughout a live stream. Each user is listed to the left of the graph, ranked from most messages sent to least messages sent. This allows the end user to gauge who their most engaged audience members are, and just how much those users are contributing to the overall chat interactions in a given live stream.

Fig. 5, similarly to Fig. 4, displays the 10 most used emotes from a given VOD. The emote name is provided on the left-hand side and are ranked in order of total usage, while bars are placed next to each emote that display how many times an emote was used throughout the stream. This helps show in what instances or circumstances the chat was most engaged in a stream. For example, if emotes such as ”LUL”, primarily used to show laughter, and ”Kappa”, often used to imply sarcasm, both rank among the top 10 used emotes across the stream, one can infer that the chat was most engaged in comedic or jovial interactions.

Fig. 6 shows the total percentages of users who are sub-

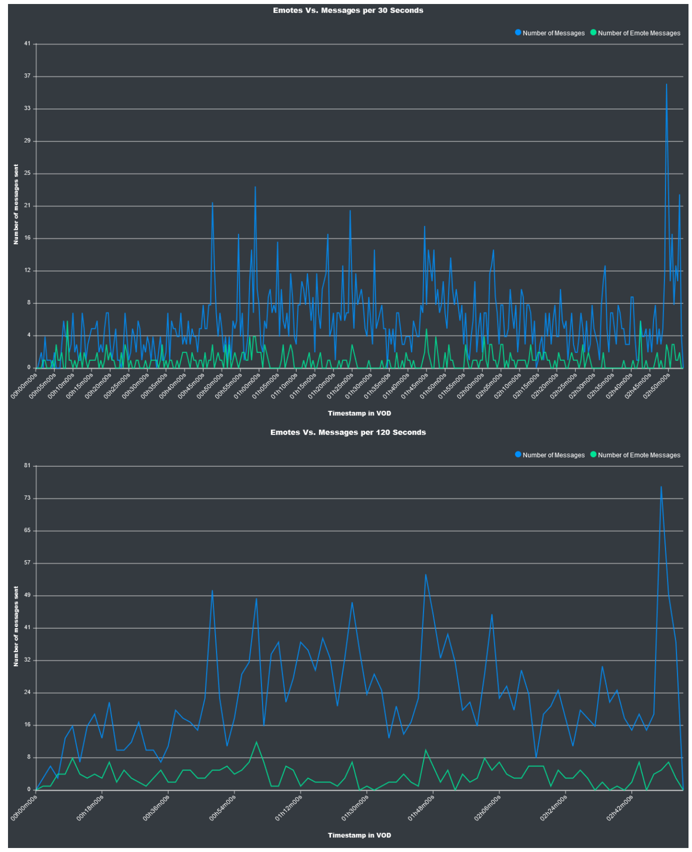

Fig. 3. A comparison of emote message and non-emote message visualization using bins of 30 seconds (top graph) and bins of 120 seconds (bottom graph).

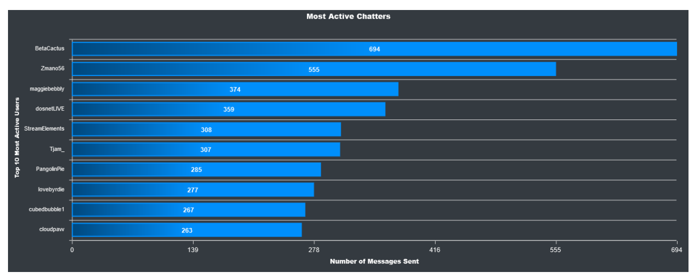

Fig. 4. Displays the 10 users who sent the most chat messages over a given video, ranked in order of most messages sent to least, and displays how many messages that user sent.

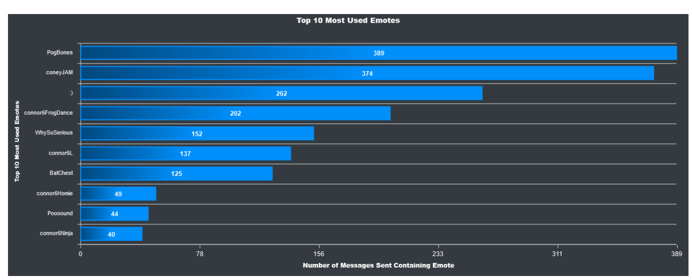

Fig. 5. Displays the 10 most used emotes during a given video, ranked in order of most messages used in to least, and displays the number of messages with a given emote in it.

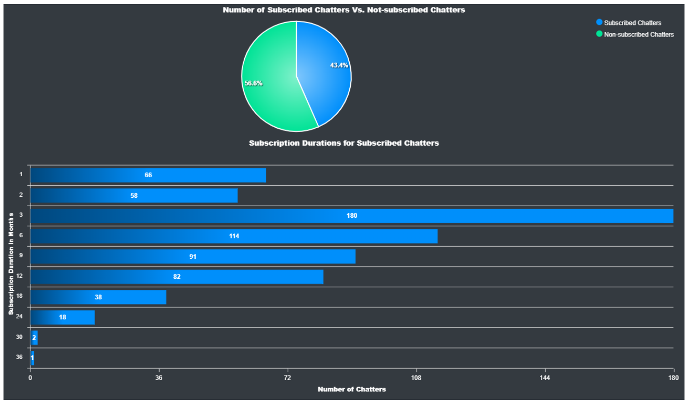

Fig. 6. Shows a pie chart showing the total percentage of messages sent subscribed and unsubscribed users. In addition, it also shows the duration of chat users' subscriptions.

scribed to a streamer and unsubscribed across a stream in terms of a pie chart with two sections comprised of both populations. This allows for an easy view of which groups engage more with a stream, and if members who engage with content are likely to help pay to support a streamer. This graph also provides a chart of the duration of subscriptions in number of months, allowing an easy view to how long users have been subscribed for and if subscribers are likely to continue their subscription into the future.

IV. Results

Since our API is designed as a tool to ingest information from users, there is no static data set from which results can be drawn from. Instead, results are brought in and visualized on an individual basis based on what VODs the user wishes to input. As such, our results take various VODs from Twitch, analyze select pieces of data presented for each, and discuss what the data could indicate to an individual using the web app in such a scenario.

A. Total Messages and Correlation to Highlights

Fig. 7 depicts a spike in messages from a stream of a Super Mario 64 120 Star speedrun from speedrunner "cheese". On the graph "Number of messages per 30 seconds", a large spike in total messages occurs between 2:49:00 and 2:49:30, jumping from 5 messages over 30 seconds to 37 messages over the same span [7]. This corresponds with a moment where the runner manages to save himself from a death with a series of close jumps only to ultimately die shortly after. The chat responds immediately with shock at the accomplishment and as the events unfold progresses into a mixture of shock and dismay at the failed run with a small amount of joking that pokes fun at the sudden failure of the run and the nature of the events that unfold. In the event that this VOD would be turned into a YouTube highlights reel, this event would likely serve as a good highlight given the notable chat response and the collective emotion that the chat conveys over such a short span of time.

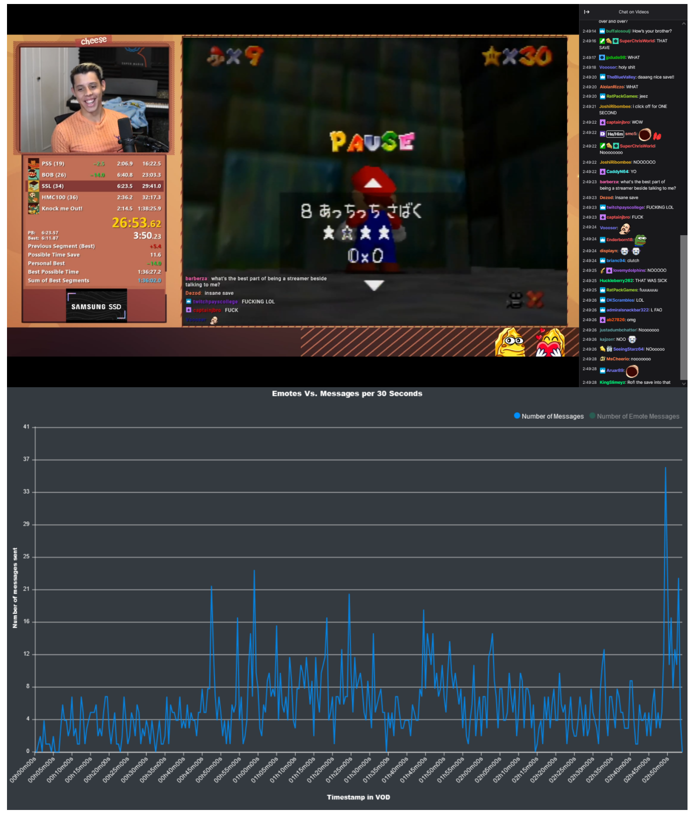

Fig. 7. A screenshot of Twitch user cheese's "120 star speedruns for WR" stream and chat on November 6th, 2021. This screenshot is at timestamp 02:49:29, which occurs during the spike in total messages, as marked in red on the graph above.

B. Emote Usage, Notable Moments, and Contextual Emote Usage

Fig. 8 depicts a spike in emote usage from a stream of a Batman: Arkham City Any% (Easy) speedrun done by speedrunner "ShikenNugggets" and hosted by Games Done Quick (GDQ), a group that live streams speedruns for charity. On the graph Number of emote messages per 30 seconds, a dramatic increase of messages with emotes happens between 1:08:00 and 1:08:30, going from only 4 total emote messages over the 30 second period to 107 total emote messages [8]. In this time frame, the speedrunner calls out "Time", indicating that the speedrun is completed and successful. The chat responds with a series of emotes congratulating the runner, primarily using two emotes from the GDQ channel: gdqTime, being a black-and-white checkerboard flag with "TIME!" written underneath it; and gdqClap, being a pair of white gloves clapping with pink lines to indicate noise. From this data, we can infer that gdqTime is likely used as a call-and-response to the declaration of "Time" at the end of a speedrun and that gdqClap is used in a congratulatory manner. That is not to say that these are the only circumstances under which these emotes are used, however that in this specific instance these emotes are used in such manners to respond to the event.

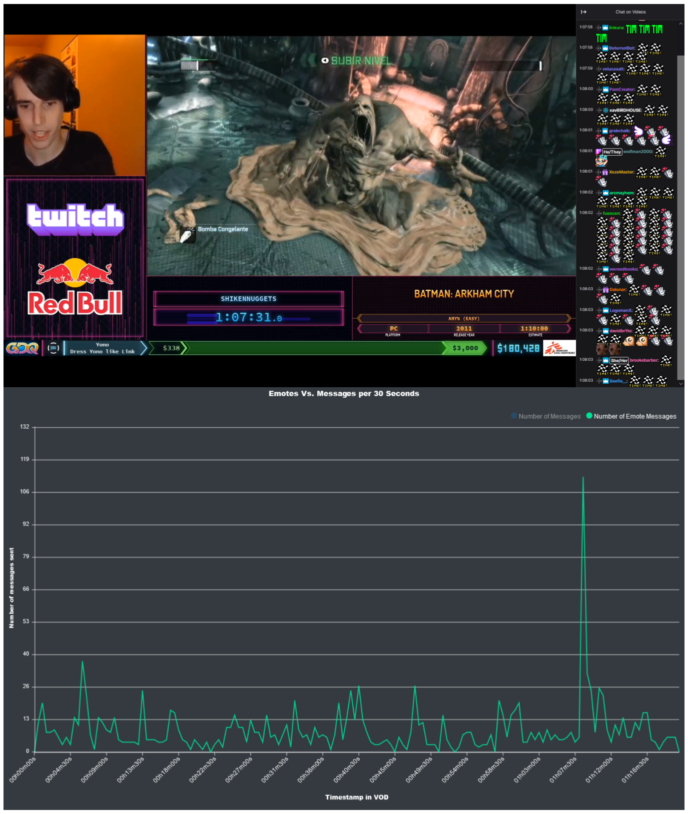

Fig. 8. A screenshot of Twitch user GamesDoneQuick's "Batman: Arkham City by ShikenNuggets" stream and chat on August 18th, 2020. This screenshot is at timestamp 01:08:04, which occurs during the spike in messages including emotes, as marked in red on the graph above.

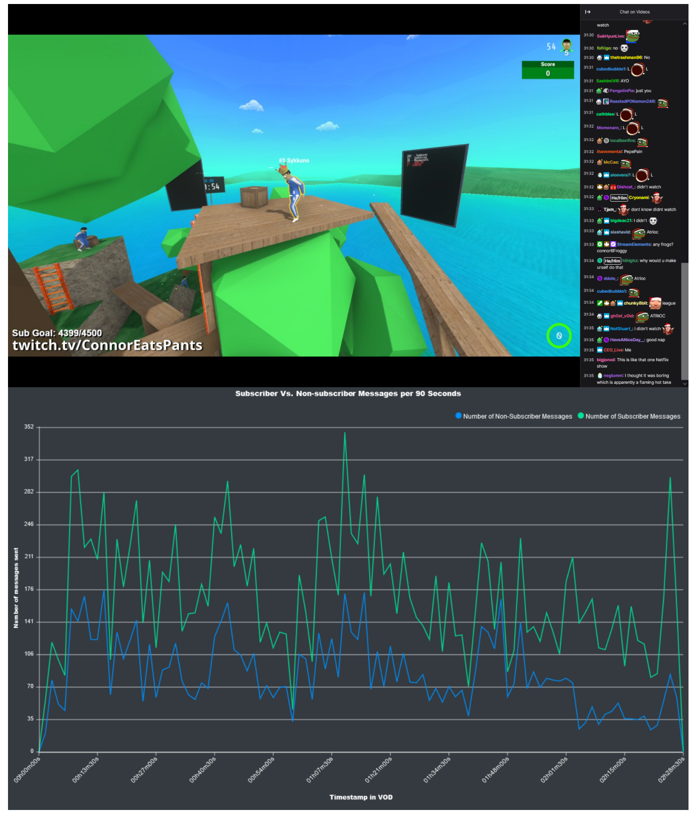

Fig. 9. A screenshot of Twitch user ConnorEatsPants's "THE FACE OF AMONG US IS BACK" stream and chat on November 10th, 2021. This screenshot is at timestamp 00:31:35, which occurs during the period marked in red on the graph. Subscribers are indicated with a badge next to their name which is depicted as the character *Chao* from the game franchise *Sonic the Hedgehog*.

C. Messages Sent by Subscribed and Unsubscribed Users and Viewer Engagement

Fig. 9 depicts the number of messages sent by subscribed users and unsubscribed users during a stream of Crab Game by ConnorEatsPants. The peaks and troughs of both messages from subscribed users and messages from unsubscribed users coincide with one another throughout the entire data set so individual events at specific moments are difficult to discern. However, there is a noticeable trend throughout the stream where there are twice as many messages sent by subscribed users than by unsubscribed users [9]. With this, we can conclude that subscribed users are most likely to engage with the streamer's content of a similar format to this stream. This trend may not necessarily apply to all streams of a user, but content similar to this may have a similar ratio of subscribed users' engagement to unsubscribed users' engagement. If a similar engagement for a future stream or event is desired, one can plan accordingly to have content similar to a stream such as this.

V. Conclusion

At this stage, our visualizations on their own can only draw general conclusions— such as in using large spikes in activity to find notable moments in streams, how many subscribed members versus unsubscribed members use the chat over a given stream. This indicates what moments in a VOD may be ideal candidates for highlights or what kind of audience engages with certain content. However our visualizations can be used in conjunction with manual analysis of stream data for more complex analysis to find what occurrences in a given stream yield more engagement from a chat, in what contexts given emotes are used to respond to given events, or what chatting cultures are created around particular streamers or in particular environments. It is also worth discussing that how chats respond to events, and thus the data collected from and analysis of such events, is entirely context and culture dependent. Each stream's chat will be a completely different environment with completely different attitudes given certain events. Larger group events where individuals are not as central to the stream, such as charity streams, may yield interaction from large donations and big moments while an individual streamer may be able to get more interactivity from inside jokes or comments directed to stream viewers. This will ultimately impact how the visualization will need to be implemented and why it is geared more towards streamers and content creators rather than the general public, as they will best be able to judge how their chat will engage with content and how this may impact the data.

A. Future Directions

One of the most important steps for future implementation of the web app is to expand what types of VODs are used in the app. A majority of tests done came from Twitch, were from English-speaking streamers, and are gaming-related content. In order to demonstrate further functionality and analysis, using as diverse a selection of VODs as possible is critical. Specifically, we would like to use VODs of YouTube live streams, content that is not in English, and non-gaming streams such as: art streams, music streams, and chatting or discussion streams. Additionally, YouTube allows uploaded videos to premiere as a live stream with regular live stream interactivity. These uploads, as they are not recorded live, will require different analysis to that of live streams and would be worth investigating how data correlates with premiered videos and what conclusions can be drawn from the data collected. Now that we have the created a basic framework of the application that performs the basics of what we want to achieve, we now want to generate some more visually appealing visualizations, such as adding a dashboard to the top of the web page listing easily digestible information like total messages, VOD length, total users who chatted, total number of unique emotes used, most active chatter, and most used emote. We also need to incorporate user interactivity with the visualizations. Users should be able to zoom in and out of the visualizations that depict messages sent over time, and specify which emotes they want to see the usage of over time. Also, Twitch only recognizes its own built-in emotes, however a significant number of emotes come from third-party plugins called Better Twitch TV (BTTV) and FrankerFaceZ (FFZ). These third-party browser extensions render emotes that are hosted separately from Twitch and are displayed to the user by replacing text with an image on the user's end, which means Twitch does not tag the messages sent as containing emotes. We would like to implement the functionality of detecting and tagging messages that contain third-party emotes. Another interesting feature to integrate into this project would be to create a directed graph showing the interactions between individual viewers. This would be a similar visualization to Kiran Gershenfeld's *Visualizing Twitch Communities* tool, which displays the overlapping communities on Twitch [2]. This may prove to be an interesting visualization to show community interactions with each other, rather than just with the content creator. It may also visualize popular members of communities who actively engage with others in the community.

VI. References

References

[1] B. (n.d.). EmojiViz - Experiments with Visualization of Real-time Emoji Twitter Trends. Retrieved October 11, 2021, from https://www.nowherenearithaca.com/2015/11/emojiviz-experiments-with-visualization.html

[2] Visualizing Twitch Communities: Graphing communities on Twitch.tv in a visually intuitive way. Gershenfeld, K. (2021). GitHub. Retrieved October 11, 2021, from https://github.com/KiranGershenfeld/VisualizingTwitchCommunities

[3] Streamers and games. Twitchmetrics. (n.d.). Retrieved October 11, 2021, from https://www.twitchmetrics.net/

[4] Twitch channels, games and Global Statistics. TwitchTracker. (n.d.). Retrieved October 12, 2021, from https://twitchtracker.com/

[5] "EXCELLENT ! nice guy we winner this and wins next ok !! UNSTOPPABLE SWAGGGGG !!!!!!!" by loltyler1. (2021, October 13th). Twitch. Retrieved November 6th, 2021, from https://www.twitch.tv/videos/1175882232

[6] Rothenberg, M. (2013, July). Realtime emoji use on twitter. Retrieved November 9, 2021, from http://emojitracker.com/

[7] "120 star speedruns for WR — !ches !academy" by cheese. (2021, November 6th). Twitch. Retrieved November 10th, 2021, from https://www.twitch.tv/videos/1198226918

[8] "Batman: Arkham City by ShikenNuggets" by GamesDoneQuick. (2020, August 18th). Twitch. Retrieved November 10th, 2021, from https://www.twitch.tv/gamesdonequick/video/713902374

[9] "THE FACE OF AMONG US IS BACK // !emotecontest" by ConnorEatsPants. (2021, November 10th). Twitch. Retrieved November 10th, 2021, from https://www.twitch.tv/videos/1201839966

[10] ofietze/twitchChatAnalyzer: Analyzing the chat of twitch vods and assigning a score based on most used words/emotes. (2020, January 23rd). GitHub. Retrieved December 7th, 2021, from https://github.com/ofietze/twitchChatAnalyzer

[11] LasTechLabs/Twitch-VOD-Chat-Analyzer: Twitch VOD Chat Analyzer is a fast-and-dirty tool to visualize the chat activity in a twitch stream, over the course of that stream. (2020, November 29th). GitHub. Retrieved December 7th, 2021, from https://github.com/LasTechLabs/Twitch-VOD-Chat-Analyzer