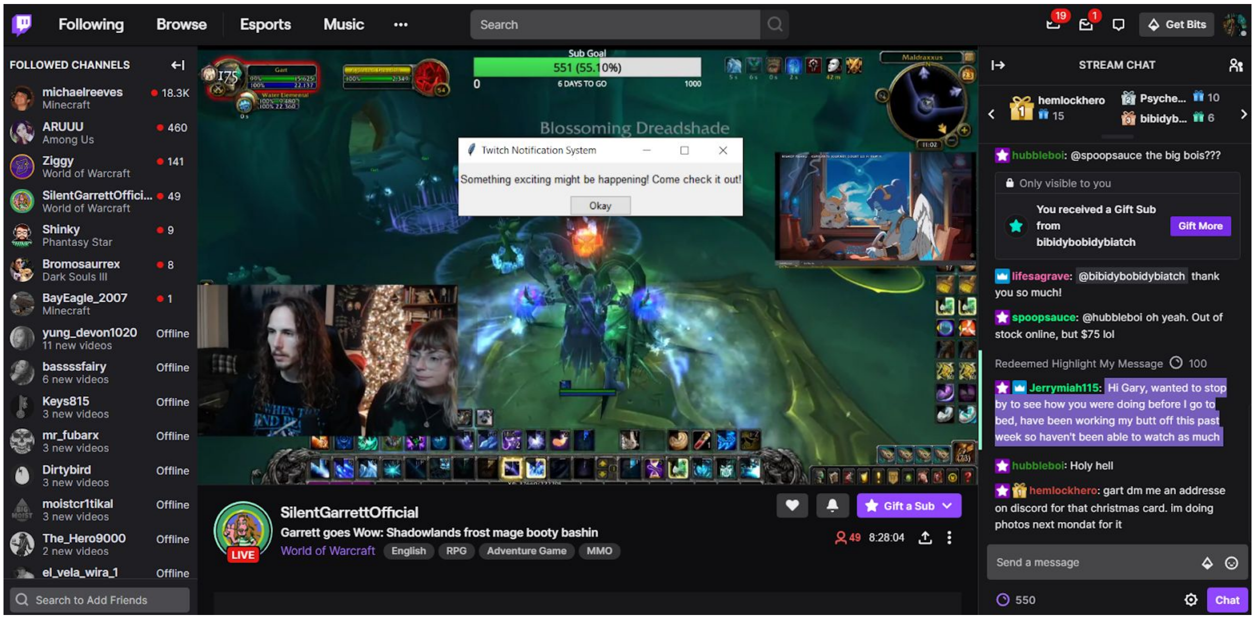

Following
Browse
Esports
Music
Search
19
1
Get Bits
FOLLOWED CHANNELS
michaelreeves Minecraft 18.3K
ARUUU Among Us 460
Ziggy World of Warcraft 141
SilentGarrettOffici... World of Warcraft 49
Shinky Phantasy Star 9
Bromosaurrex Dark Souls III 8
BayEagle_2007 Minecraft 1
yung_devon1020 11 new videos Offline
bassssfairy 6 new videos Offline
Keys815 3 new videos Offline
mr_fubarx 3 new videos Offline
Dirtybird 3 new videos Offline
moistcr1tikal 3 new videos Offline
The_Hero9000 2 new videos Offline
el_vela_wira_1 Offline
Search to Add Friends
Sub Goal
551 (55.10%)
0
6 DAYS TO GO
1000
Blossoming Dreadshade
Maldraxxus
SilentGarrettOfficial
Garrett goes Wow: Shadowlands frost mage booty bashin
World of Warcraft
English
RPG
Adventure Game
MMO
LIVE
Gift a Sub
49
8:28:04
STREAM CHAT
hemlockhero 15
Psyche... 10
bibidyb... 6
Send a message
550
Chat

Twitch Notification System

Something exciting might be happening! Come check it out!

Okay

hubbleboi: @spoopsauce the big bois???

Only visible to you

You received a Gift Sub from bibidybobidybiatch

Gift More

lifesagrave: @bibidybobidybiatch thank you so much!

spoopsauce: @hubbleboi oh yeah. Out of stock online, but $75 lol

Redeemed Highlight My Message 100

Jerrymiah115: Hi Gary, wanted to stop by to see how you were doing before I go to bed, have been working my butt off this past week so haven't been able to watch as much

hubbleboi: Holy hell

hemlockhero: gart dm me an addresse on discord for that christmas card. im doing photos next mondat for it

Hype Train Notification System

11/25/2020

Andres Ramos

Jake Lasley

Speech Language Processing

Dr. Nigel Ward

Background

In today's world, entertainment has shifted drastically. In person events are no longer an option while the ongoing COVID-19 pandemic is happening, so many people are moving to online platforms to keep them busy such as Twitch. According to an article from The New York Times, Twitch has had a near 20% surge in traffic since stay at home orders have been into effect (Koeze & Popper, 2020). With that being said, many individuals use Twitch to still stay social and connected amid being home with little social interaction with others. However, that does not mean they don't have other commitments to attend to. Twitch by many will be watched and left on as background noise, while others use !lurk commands to indicate that they have the stream open to stay supportive but are not actively paying attention.

Background Terms

- Twitch Video on Demand (VOD)
    - Recently recorded live streams as well as their chat history is available for about 3 months after a stream has ended. We will be using some of these to collect our data and label them accordingly.
- Twitch Clips
    - Clips are made to be highlights of streams that viewers make themselves for the streamer. This will enable us to easily collect the chat samples from these clips and mark them as 'exciting' depending on the context of what is happening in the chat.

What are we trying to achieve using this information?

"Lurking" is a common action users of twitch take when not actively viewing the stream but still supporting a streamer by leaving their page open. These users put themselves in positions where they might miss something they wish they would have seen live and actively been a part of the moment. Often members of stream communities can clip these moments however, it does not offer the same feeling as being there watching live. This is where this notification tool can benefit users by alerting them when it detects something interesting is happening.

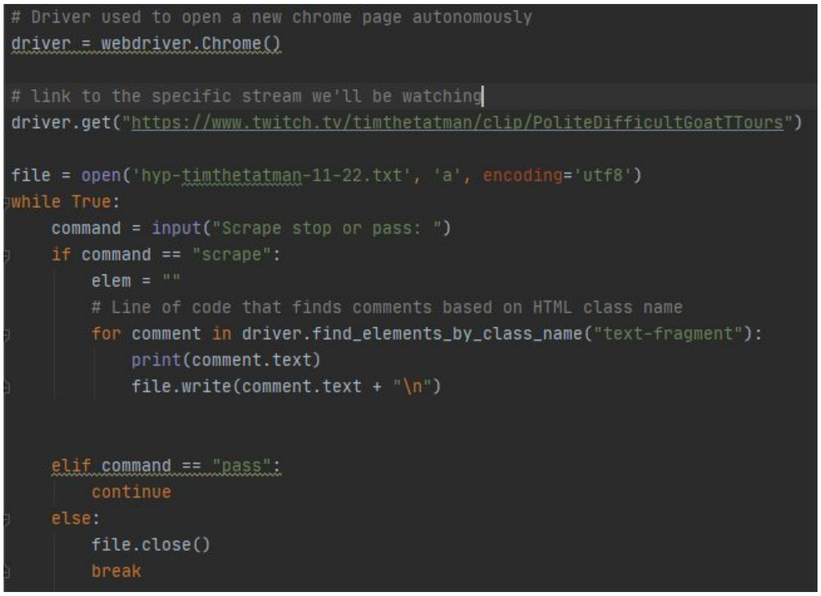
- The most interesting part of the system is being able to execute our state machine to work with our classification model. Being able to get it to successfully reach the correct state upon analyzation of a block of text will be interesting to see in implementation.
- If we are able to display a notification either at or right after something exciting happens, that will be a huge success for our project deliverable. If we are able to determine when something exciting happens during the stream in the command line, that will also be a success however we will try and implement some kind of pop up notification system. We would rather have something that interrupts the users and notifies them however we will rely on the command line if not.

Data Collection

- Hype Data
    - While watching a number of streams, gathering and labeling data proved to be a tedious task in some cases. Again, this was due to some instability of the web driver browser.
    - For our “Hype” classified data, we were able to web scrape around 1025 comments exclusively from clips that were exciting. Some of which were featured on the top clips of the week page of twitch, others just from known streams where we knew the data would be appropriately labeled.
    - The number of streams we needed to scrape for this data set was large because not every stream has long lasting “hype” moments so we needed to constantly find more. We scraped about 15 stream chats in total.
- Normal Streaming
    - This dataset was much easier to gather because of the larger stream of comments and larger time we had to gather. This data was also gathered last and we were able to keep the web driver functioning with less problems at this point.
    - For our “Normal” data, we were able to webscrape about 1175 comments from various streams. The number of streams needed to scrape was much less than the other set, ending up at around 5 streams in total.
- Emotes
    - One important thing to note about Twitch chats is that they are full of emotes. Small images uploaded by the channel owner themselves to offer incentives for people to subscribe to their channel. Many of these can be found in chat data and each have a unique code identifier however for the sake of this project we have excluded those as we cannot accurately extract their codes.
- Full Clips
    - Following the same process for collecting “hype” and “not hype” comments for training our classifier, we gathered sequential comments from streams and stream clips to validate the state model. These comments are separated by stream, and are not used for training the classifier.

Web Scraping

- Our data was collected using a web driver that autonomously opens and loads a twitch page that looks for specific elements within the web page’s HTML. Ideally there is an API that allows for explicit extraction of comment data created by Twitch themselves, however for the sake of this project this was the method used to extract comments.

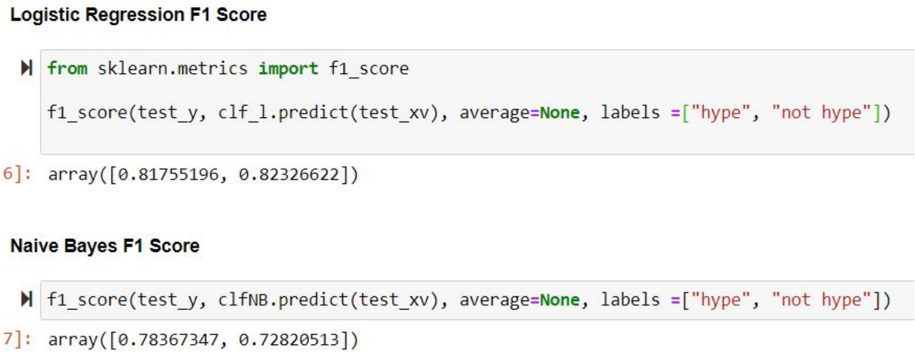


```
# Driver used to open a new chrome page autonomously
driver = webdriver.Chrome()

# link to the specific stream we'll be watching
driver.get("https://www.twitch.tv/timthetatman/clip/PoliteDifficultGoatTTours")

file = open('hyp-timthetatman-11-22.txt', 'a', encoding='utf8')
while True:
    command = input("Scrape stop or pass: ")
    if command == "scrape":
        elem = ""
        # Line of code that finds comments based on HTML class name
        for comment in driver.find_elements_by_class_name("text-fragment"):
            print(comment.text)
            file.write(comment.text + "\n")


    elif command == "pass":
        continue
    else:
        file.close()
        break
```

Figure 1: Code needed for web scraping.

Classification Using Logistic Regression

We decided to test using Logistic Regression and Naive Bayes classification. First creating a simple bag of words model, then vectorized all the words found in each comment. For testing, we split all our data by 80/20 %. We decided to test these two classification models and found that the logistic regression classifier performed better. This is likely due to the nature of the data and the classes they belong to. Most "Hype" comments we've noticed include the same thing, with heavy uses of the words "POG" and "KEKW" being some of the most frequent. Since there are only two types of class, it's easier for the logistic regression model to determine what is hype and what is not. The notebook with these models will be submitted with this report.



Logistic Regression F1 Score

```
from sklearn.metrics import f1_score

f1_score(test_y, clf_l.predict(test_xv), average=None, labels =["hype", "not hype"])
```

```
6]: array([0.81755196, 0.82326622])
```

Naive Bayes F1 Score

```
f1_score(test_y, clfNB.predict(test_xv), average=None, labels =["hype", "not hype"])
```

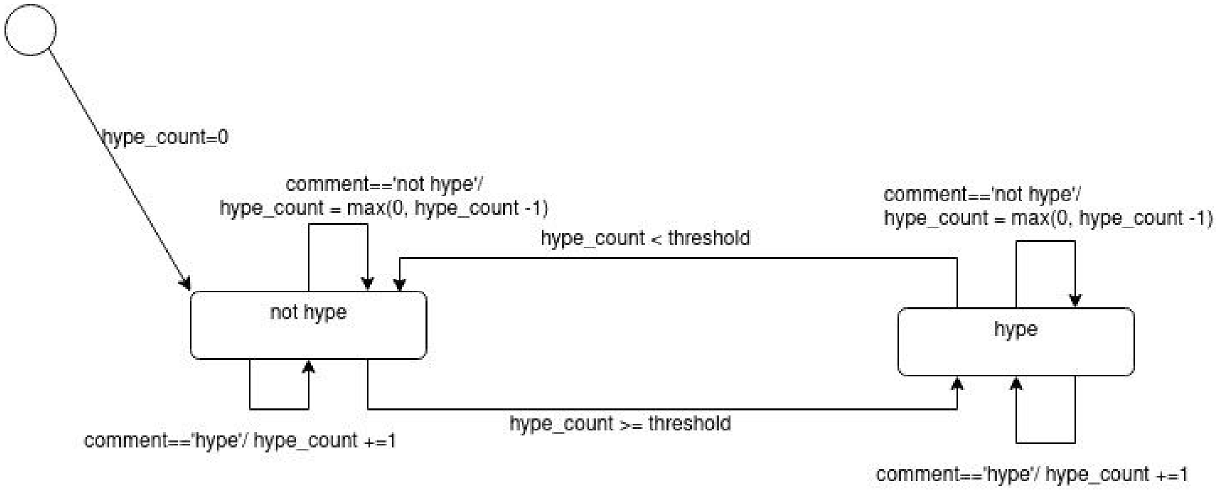
```
7]: array([0.78367347, 0.72820513])
```

Figure 2: F1 Scores using Logistic Regression and Naive Bayes

State Prediction

In order to determine if the stream itself is currency 'hype' or 'not hype' we built a simple state machine that uses the classification for it's individual comments to determine its state. When a hype comment comes in, the state machine increments it's 'hype_count' variable. When a ‘not hype’ comment comes in, the state machine decrements the ‘hype_count’ variable. Once the hype_count reaches a certain threshold, set at 10 for this experiment, the state changes to ‘hype’. The hype comments and not hype comments essentially cancel eachother out. This means that at a given point in time, there need to be at least 10 more hype comments than not-hype comments for the stream to be considered hype. The state transition diagram below shows this.

We found our model to accurately predict when something exciting was happening in a clip. For all of our test clips, we were able to identify the state of hype stream clips as ‘hype’, and not-hype stream clips as ‘not hype’.


hype_count=0
comment=='not hype'/
hype_count = max(0, hype_count -1)
hype_count < threshold
comment=='not hype'/
hype_count = max(0, hype_count -1)
not hype
hype
comment=='hype'/ hype_count +=1
hype_count >= threshold
comment=='hype'/ hype_count +=1

Results

The Comment Predictions table below shows the results of the predictions of our logistic regression classifier on the comments from a particular stream clip.

The Stream State table shows the results of our state machine on those same clips based on the predictions made by the logistic regression classifier.

Comment Predictions

Name	Stream Label	Majority Prediction	% Hype Comments	# Hype Comments
hyp-esl_csgo-11-22.txt	hype	hype	95.54140127	150
hyp-timthetatman-11-22.txt	hype	hype	85.82677165	109
hyp-moistcr1tikal-11-15.txt	hype	hype	80.52631579	153
hyp-GMHikaru-11-22.txt	hype	hype	89.93055556	259
hyp-eslcsgo2-11-22.txt	hype	hype	90.47619048	95
hyp-loltyler1-11-22.txt	hype	hype	93.38842975	113
reg-silentgarrett-11-21.txt	not hype	not hype	44.68085106	21

reg-gomisworld-11-21.txt	not hype	not hype	16.0	12
reg-aforestlife-11-21.txt	not hype	not hype	24.0	30
reg-keys-11-21.txt	not hype	not hype	28.46153846	37
reg-hologramdreams-11-21.txt	not hype	not hype	38.95348837	67
reg-coffeewithbee-11-21.txt	not hype	not hype	28.68852459	35
reg-sudarezz-11-20.txt	not hype	not hype	48.4375	31

Stream State

Name	Stream Label	Majority State	% of time hype (t=comments)	# of comments in hype state
hyp-esl_csgo-11-22.txt	hype	hype	91.08280255	143
hyp-timthetatman-11-22.txt	hype	hype	91.33858268	116
hyp-moistcr1tikal-11-15.txt	hype	hype	75.78947368	144
hyp-GMHikaru-11-22.txt	hype	hype	93.40277778	269
hyp-eslcsgo2-11-22.txt	hype	hype	91.42857143	96
hyp-loltyler1-11-22.txt	hype	hype	85.12396694	103
[redacted]				
reg-silentgarrett-11-21.txt	not hype	not hype	0	0
reg-gomisworld-11-21.txt	not hype	not hype	0	0
reg-aforestlife-11-21.txt	not hype	not hype	0	0
reg-keys-11-21.txt	not hype	not hype	0	0
reg-hologramdreams-11-21.txt	not hype	not hype	0	0
reg-coffeewithbee-11-21.txt	not hype	not hype	0	0
reg-sudarezz-11-20.txt	not hype	not hype	0	0

The following two plots show the state change over time for two of the streams. A ‘hype count’ > threshold indicates that the stream is in the ‘hype’ state. With a ‘hype count’ below the threshold the stream is in the ‘not hype’ state.

In the silentgarrett clip, while 44% of the comments were classified as ‘hype’, they did not occur close enough together for the stream as a whole to be considered a hype stream. This is depicted in Figure 3.. The moistcr1tikal clip depicted in Figure 4,on the other hand had a large number of hype comments coming in in a row, so the stream transitioned into the ‘hype’ state after about 45 comments.

Figure 3:

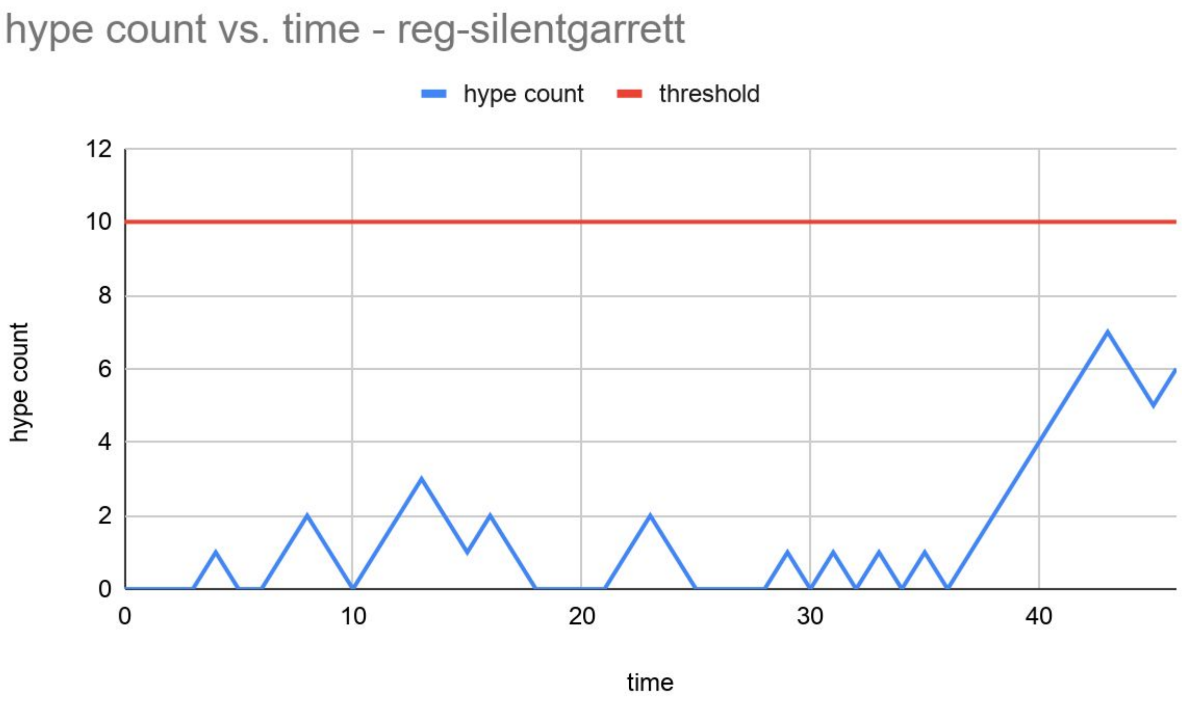
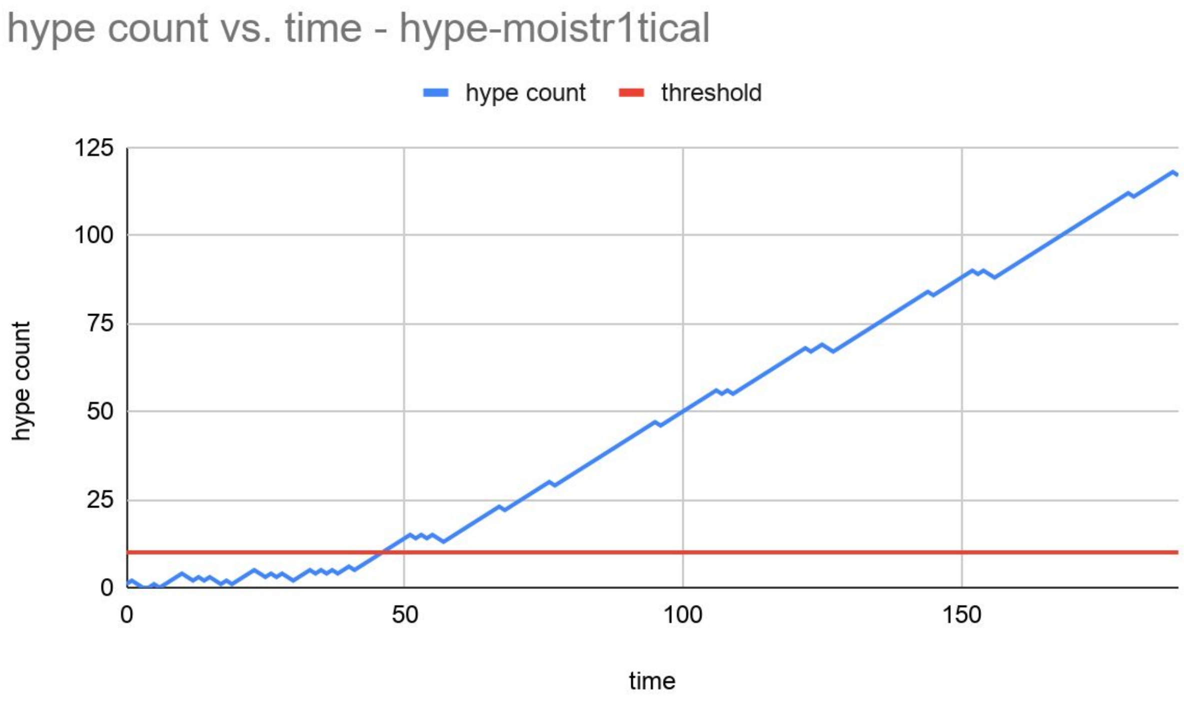


Figure 4:



What Was Learned

We have taken what we know about n gram features as well as classification methods to successfully carry out our project in a real life setting. Creating, training and testing a model was an integral part of this project as well as collecting fresh data for purposes of our project. This was extremely tedious but rewarding to know because we were able to collect and label meaningful data that helped the classifier perform as expected. Having a good data set to use for model training is important to have to be able to expect a model to perform well.

Future Improvements

Today, our system has the capability of testing chat data it's never seen and being able to classify it as hype or not hype. This partially solves the problem we are attempting to present a solution to. In future iterations, giving the capability to run on an actual live stream to test with live data. Other things to consider would be improving the list of features gathered to continue improving the classifier as well. A significant improvement would be the inclusion of an upper bound in our state machine, the reason for that being once a stream reaches that threshold of being hype it may be difficult to reduce it back down to being a normal non hype stream depending on the amount of comments that are presented. Some streams have 10s to 100s of comments that can be sent at the same time. However its worth noting that most of these streams will continually be exciting as some of the channels that have that kind of reach are among the top creators on twitch or are hosting some kind of event.

Citations

Koeze, Ella, and Nathanial Popper. “The Virus Changed the Way We Internet.” *The New York Times*, 7 Apr. 2020, www.nytimes.com/interactive/2020/04/07/technology/coronavirus-internet-use.html.