Top Streaming Services Based on Twitter Sentiment Analysis

Nathan Thomas
York University
Toronto, Ontario
nofun100@my.yorku.ca

Dominik Sobota
York University
Toronto, Ontario
sobotad@my.yorku.ca

Kirill Kresling
York University
Toronto, Ontario
kkres@my.yorku.ca

Tong Le
York University
Toronto, Ontario
letony28@my.yorku.ca

Soo Kim
York University
Toronto, Ontario
sykim88@my.yorku.ca

ABSTRACT

To evaluate the performance of shows and streaming networks, Tweets were collected on a select list of popular shows and their respective networks. The Twitter data was collected over a month period from the Twitter's real-time data and amounted to 1.1 GBs of data. These Tweets were evaluated on performance based on sentiment analysis and based on the number of Tweets. A number of techniques were used to clean and categorize Tweets with the proper show or network such as tf-idf and regular expressions. Data was served through a MySQL server to aggregate results on shows or networks. With sentiment analysis, the top performing show in our list was Chernobyl and the worst performing show in our list was The Walking Dead. The most tweeted about show was Watchmen. Shows hosted by Netflix had the highest average sentiment. With Tweets about the network themselves, Hulu had the highest average sentiment.

1 INTRODUCTION/MOTIVATION

Coupled with the rise of online streaming services, has come a plethora of television content for consumers to enjoy. In these last couple of years, streaming services and television networks now, more than ever, need to be aware of what viewers feel about their content. It is easy for a new show being released on a platform to be lost in a sea of other content and other services. In order to determine what shows to invest money into acquiring or creating, networks and streaming services, need to be able to gauge the public interest and opinion of the shows they are investing into. If a show is not attracting enough interest, or is being received poorly, the service may be better off pulling funding and investing in other shows. Alternatively, well engaging shows with a positive reception may be worth a production and marketing budget increase. Data collected from Twitter relating to television shows can act as a microcosm of the general opinion of these shows, which can fluctuate depending on a multitude of factors that are of interest to the services they are on. The data for this project will consist of text data extracted from Twitter, from Tweets relating to various pre-selected shows during the month of November in 2019. Some examples of these shows being The Walking Dead, Bojack Horseman, The Good Place, The Boys and many more. The selected shows are present on the following streaming platforms: Netflix, Amazon Prime Video and Hulu. This data was rigorously analyzed to monitor the trends of sentiment over the aforementioned time period, and the results are presented later in this report. From this, a ranking of shows based on this analyzed sentiment was able to be constructed. The project set out to answer the following questions based on the data and analyzed sentiment from said data. What are the best performing television shows? Worst? What shows are talked about the most? How do the different networks fare in overall Twitter sentiment? This analysis is important as it is a tool that can be used to determine if a show is performing well or not. As some other websites offer a ranking system via a user inputted score, our project is seeking to use their actual written opinion on twitter as our basis of determining sentiment. As stated above, the sentiments and engagements on Twitter can give an overall feeling of how a show is performing in comparison to other shows. An application of these collected sentiments can be used by the creators of the selected shows to determine if the show meets their expectation of user engagement and sentiment. By having a visual representation of this data, they have more tools at their disposal in order to determine if it is worth continuing the show on to the next season, and what types of shows they could start creating in the future. Another application is in the competition of streaming services. The analysis could offer insight into the performance of different content types of each streaming service and allow them to make better data driven decisions on what to create in the future.

2 DATA AND DATA ANALYSIS

2.1 Twitter Data

The data prevalent in this project is text data obtained from Twitter tweets, relating to a curated set of television shows that are being analyzed. This, along with the number of tweets per show, are the main criteria for the results given later in this report. To perform some pre-processing, the tweets are cleared of any smileys, emojis and links found within its text. These pre-processed tweets are then categorized into the shows they are in reference too. These categorized and cleaned tweets are then ready to be analyzed by the Natural Language Toolkit, where the resulting cleaned text, category and sentiment scores are then stored as csv files. The architecture for this process is described in further detail in the Architecture section.

2.2 Sentiment Analysis

Sentiment Analysis is ideal for the questions posed in this project as "sentiment analysis presents an efficient and effective evaluation of consumer opinions in real time" [10]. Social media platforms gives networks and streaming services a data driven means of determining the success of their television shows via viewer opinions. To analyze sentiment, a tutorial on sentiment analysis with Reddit news headlines was followed [9]. This tutorial used VADER sentiment analysis tools from the Natural Language Toolkit [8], specifically the polarity_scores method. This method assigns text four different scores. These four scores are floats representing the compound, positive, neutral and negative sentiment. The compound score ranges between -1 (negative sentiment) and 1 (positive sentiment) while the positive, neutral and negative scores range between 0 and 1.

3 ARCHITECTURE OF PROPOSED SOLUTION

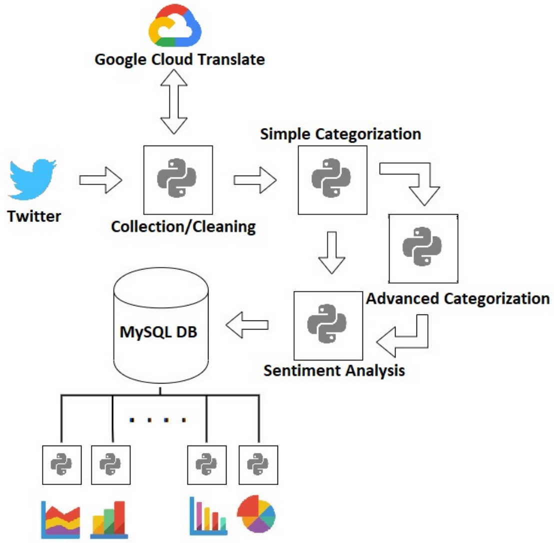

Figure 1: Analysis architecture

3.1 Data Ingestion/Storage

No pre-existing dataset was readily available for the proposed analysis of Tweets cross-referenced with the show(s) in which the commenter is referring, therefore, the architecture begins with creating the dataset. The first module begins by connecting to Twitter using the official API Tweepy and streaming a selection of Tweets [11]. To determine which Tweets are streamed and which are ignored a file was created storing strings related to each show and network. The strings in this file were all loosely related to the chosen shows and were used as a filter to determine which Tweets were streamed. Tweepy returns a complex JSON, with extraneous data that was not helpful to the analysis, therefore each Tweet was parsed and only select attributes were stored. The desired attributes were stored on disk in CSV format with only the selected columns represented in the data. In total, roughly 6 million Tweets were collected over a 14 day period totaling approximately 1.1 GBs of data. At each intermediate step between modules the data is always stored as a CSV though the structure of the CSV structure changes after the various processing steps.

3.2 Data Processing

3.2.1 Cleaning. The first step in processing the data is removing unwanted characters which may hamper the analysis of each Tweet's sentiment. The process begins by identifying which Tweets are non-English and connecting to Google Cloud Platform to translate the Tweets, however due to the financial cost of translation ($1/50000 characters) only a portion of the non-English data could be translated as a proof of concept. Once translation is done than unwanted characters can be removed. An unwanted character is any encoding string whether it be Unicode, HTML, or a symbol. The encoding strings are removed or replaced using a series of regular expressions. Once all the encoding strings are removed, one last substitution is performed to remove any character which is not one of either a word, number, @, # or whitespace character.

3.2.2 Categorization. Once the records have been cleaned the Tweets were categorized and segregated into their respective shows. To perform the initial show categorization, a simple keyword match is implored, matching Tweets which contain certain keywords including unique characters, hashtags, and mentions. The keywords are chosen manually by reading through Tweets referencing the show and using online sources to find unique identifying names referring to the shows. This method however only captures about 60% of the Tweets which were initially gathered, furthermore, it is impossible to know every keyword which will be important, therefore further analysis of the categorized shows is performed. The shows which have been successfully sorted are analysed using the natural language processing library Gensim. The corpus used to identify the important words for each show are the Tweets which were sorted in the previous step. With each of these corpora, we perform tf-idf analysis on the documents within each corpus to determine which new keywords uniquely identify the shows [5]. Once analysis has been performed the best words are manually chosen by the implementer based not only on tf-idf ranking but also how unique that keyword is to the show. After new identifying words have been discovered, the initial categorization module is rerun using the new keywords.

3.2.3 Sentiment Analysis. After categorization, the sentiment of of each of the Tweets is ready to be calculated and appended to each record, but there is still some minor preprocessing which needs to be performed before analysis. It is potentially possible to perform sentiment analysis before sorting the Tweets into their respective shows, however, an issue arises when calculating the sentiment first; the issue being that many show titles are not sentiment neutral. A show such as 'Chernobyl' has a completely neutral title, therefore if the show title appears in the Tweet there is no bias when the sentiment score is calculated. Contrast 'Chernobyl' with 'The Walking Dead' which has an extremely negative score, leading any Tweet containing the show's name to be perceived more negatively than other shows. To alleviate this issue, a simple replacement was employed, replacing occurrences of the titles in the Tweets where the title of the show is determined to be non-neutral. This step is where it is helpful to know the name of the series prior to calculating the sentiment, as the sentiment of each show title can be calculated first and depending on whether it is neutral, a replacement in the text is made. Though this system was capable of producing drastic changes in the sentiment score from the initial calculations which did not implement a title substitution, this method is not a perfect fix. The first issue is that the substitution is a simple regular expression match that only looks for the show title in the string, therefore, it does not capture variations or substrings of the title (i.e 'Doctor Strange vs 'Dr. Strange). The second issue is that the replacement algorithm only searches for the title of the show that the Tweet is related to, meaning that if another show title is referenced in the Tweet it will not be replaced.

3.3 Data Serving

After exporting the sentiment data into a CSV file, it was ready to be queried and aggregated into meaningful and easy to understand data. MySQL was used for this purpose as it has advanced querying and works very effectively for aggregating data. The IDE, Datagrip, by JetBrains [1] was used to store and query the data from the sentiment analysis. The CSV file was converted into a MySQL table using Datagrip's file import functionality, which was well optimized and ran in linear time (importing over 2 million rows in seconds). Appending data using the same feature was just as simple, as discovered when missing data was added to sentiment table in the database.

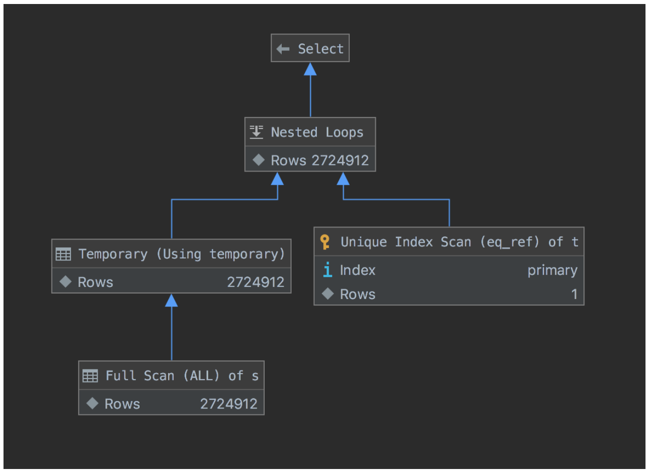

Figure 2: The schema of the database

The database consisted of four tables, including the sentiment table (Figure 2). The remaining three were used to store supplementary data for querying. A table named tv_shows was used to store the tv show values that were referenced for our data, also serving as a final layer of protection so invalid values would not be considered in the data. The network table stored the values for a select few networks that our chosen tv shows were hosted on. Finally a show_network table hosted a one-to-many relationship between the tv shows and their network(s), which were both foreign keys to their respective tables.

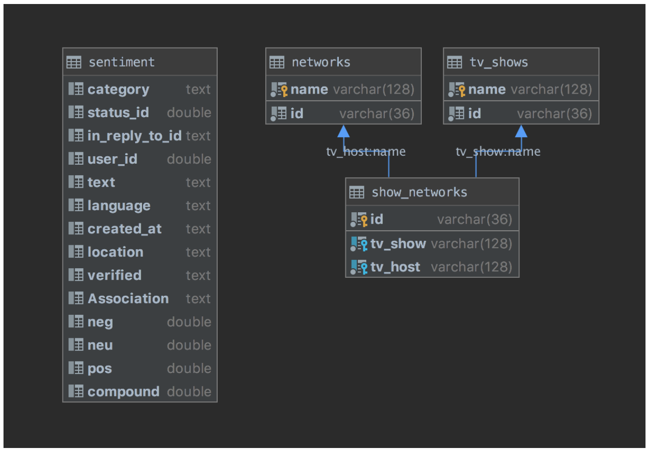

Figure 3: The query plan

Since the three supplementary data tables in the database were small and indexed on their names, the runtime of the queries was bound by the speed of processing the bulk data in the sentiment table. This was primarily done using a filescan and then a nested loop join on the appropriate table(s) as seen in Figure 3. Subsequent queries ran as fast as just under 200 milliseconds, proving that Datagrip stores the temporary file-scanned table; making sequential queries on the same bulk data efficiently.

3.4 Data Visualization

Upon aggregation of the data, it is then ready for the final step in the process, visualization. When the data regarding Tweets is aggregated it is visualized using the matplotlib library in Python to graph the data (8). Data referencing the sentiment for each show was represented using bar graphs which were sorted on the compound value from NLTK analysis.

4 EVALUATION/RESULTS

4.1 TV Shows

4.1.1 Sentiment. For the overall sentiment of our shows, there was an average positive score of 0.100658, an average neutral score of 0.86429, an average negative score of 0.035049, and an average compound score of 0.149235. The average Tweet was neutral.

The top ten shows based on compound sentiment in descending order were Chernobyl, Stranger Things, The Family Man, Lost in Space, The Haunting of the House, Cartel Crew, Castle Rock, Living with Yourself, Modern Love and The Man in the High Castle. See Figure 4. The average compound sentiment score of the top ten shows is 0.254574 which is 70.6% higher than the average compound score of all shows collected. The bottom ten shows based on sentiment analysis in descending order were Bojack Horseman, Tom Clancy's Jack Ryan, Watchmen, Big Mouth, American Vandal, Peaky Blinders, Mr. Robot, The Witcher, Black Mirror and The Walking Dead. See Figure 5.The average compound score of these shows was 0.023759 which is 84.1% lower than the average compound score of all of our shows.

4.1.2 Tweet Popularity. The amount of people talking about show can also help gauge the performance of a show. The top ten shows are listed in Table 1. Cross referencing top shows based on sentiment with top shows based on number of tweets can give new information. For example, the shows that are in both the top ten

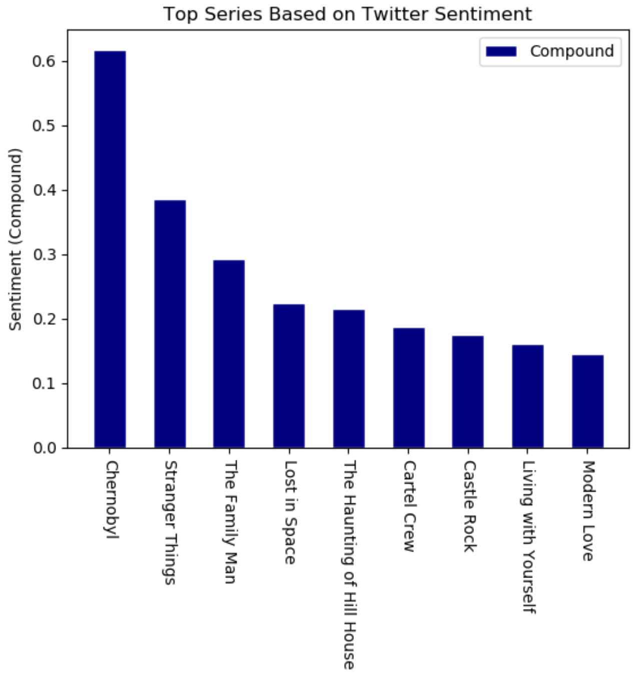

Figure 4: Top ten shows based on average Twitter sentiment

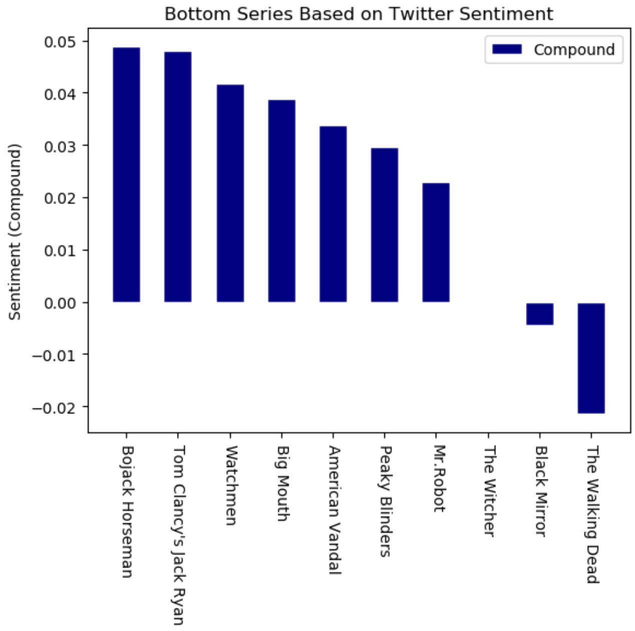

Figure 5: Bottom ten shows based on average Twitter sentiment

Show	Number of Tweets
Watchmen	57566
Stranger Things	57233
Game of Thrones	40215
The Good Place	32830
The Witcher	28084
Chernobyl	27728
The Walking Dead	23880
Silicon Valley	22408
Peaky Blinders	21063
Black Mirror	15994

Table 1: Top ten shows by number of Tweets

Show	Average Rating	Number of Reviews
Chernobyl	9.5	389823
Stranger Things	8.8	684971
The Family Man	8.7	13571

Table 2: The IMDB Ratings of our top three shows by sentiment

Show	Average Rating	Number of Reviews
The Walking Dead	8.2	808593
Chernobyl	8.8	365894
The Witcher	n/a	n/a

Table 3: The IMDB Ratings of our bottom three shows by sentiment

in sentiment and Chernobyl and Stranger Things. A large number of tweets talk positively about these shows, signalling great success. However there are also shows that are low in sentiment and were tweeted about a lot. These shows are Watchmen, Black Mirror and The Walking Dead. This signals that these shows are not performing well as much discussion is negative.

4.1.3 IMDB Reviews. Here we compare our data with more historic data from IMDB [3]. Ideally we would have liked to augment our data on top of SQL data on top as part of the data project. However, only a subset of the data is available and there are multiple shows that share the same name requiring manual work to match shows. Instead some ratings were gathered from searching on the IMDB site. See Table 2 and Table 3. When looking at the IMDB ratings of these select shows, there is not much difference. The second top performing show by sentiment analysis has the same rating as the second bottom performing show.

4.2 Network Sentiment

One show that was tracked turned out to be hosted exclusively on a network that did not host many of our other shows. This network has been excluded in the following results. The average compound sentiment of the networks based on their shows is 0.155236. Netflix had the highest compound sentiment value at 0.277989, higher than the average by 79%. See Figure 6. Average sentiment values differ greatly when analyzing tweets about the networks themselves. See Figure 7. Netflix has the lowest compound sentiment at 0.096958 lower than the average network sentiment of 0.293376 by 67%.

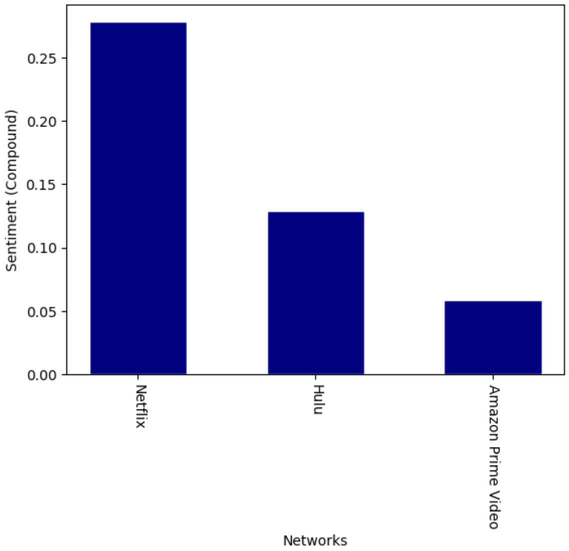

Figure 6: Average network sentiment based on shows

Instead the top network here is Hulu with 0.565411, higher than the average by 93%.

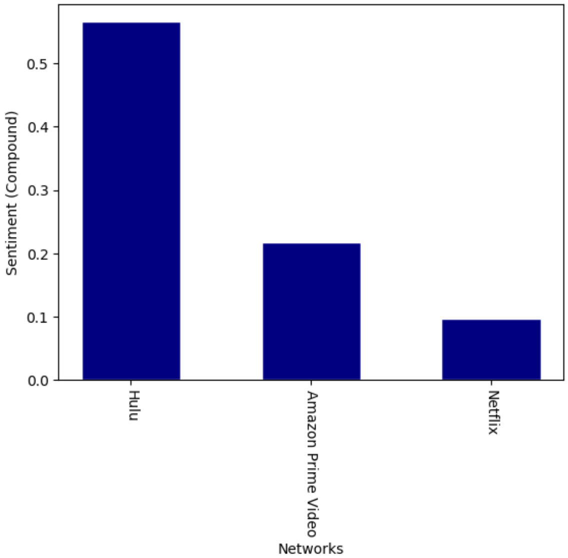

Figure 7: Average network sentiment

5 CONCLUSION

There are many limitations with the architecture beginning with the dataset itself. The data collected can never be complete because not all users who comment on a certain topic use keywords and hashtag representative of the topic. Likewise it is not a trivial task to map Tweets when the context of the Tweet has been lost and there is no easy way to identify which topic the Tweet was referring. As well as organizing the topics, calculating the sentiment can be a challenge as well. There are some words and phrases which may lead to undue negative or positive evaluation such as 'The Walking Dead' and 'The Good Place' which have a negative and positive sentiment respectively. Likewise when calculating sentiment, a show with the title 'Chernobyl', which itself is not negative, but references a disastrous event which has negative connotations can skew the results against it. The highlights of the analysis were that sentiment of the Netflix network were low compared to other networks, but the shows hosted by Netflix had the highest sentiment. Also comparing our bottom sentiment rankings to historical data from IMDB presents us with polarizing results. There are many refinements and improvements that could be made at each stage of the architecture, including steps which were not implemented due to limitations of both time and Twitter. During data ingestion, the Tweet ID the user is responding to is recorded, this was supposed to be used after categorization to potentially connect the replied Tweet to the show as well. Another feature that was left out due to design choices of Twitter was incorporating the location. The original plan included breaking down the results by location, the issue with this is that Twitter allows users to enter any string as their location resulting in the location field being either empty, in a non-standard format, or jokes.

6 LIBRARIES

Tweepy [11], NTLK [8], matplotlib [4], tweet-preprocessor [7], pandas [6], Gensim [2].

REFERENCES

[1] JetBrains n.d. *Data Grip: The Cross-Platform IDE for Databases & SQL.* JetBrains. Retrieved December 14, 2019 from https://www.jetbrains.com/datagrip/

[2] n.d. *gensim.* Retrieved December 14, 2019 from https://pypi.org/project/gensim/

[3] IMDB n.d. *IMDB Reviews.* IMDB. Retrieved December 14, 2019 from https://www.imdb.com/

[4] n.d. *Matplotlib: Python plotting.* Retrieved December 14, 2019 from https://matplotlib.org/

[5] gensim n.d. *models.tfidfmodel – TF-IDF model.* gensim. Retrieved December 14, 2019 from https://radimrehurek.com/gensim/models/tfidfmodel.html

[6] n.d. *Python Data Analysis Library.* https://pandas.pydata.org/,lastaccessed=

[7] n.d. *tweet-preprocessor.* Retrieved December 14, 2019 from https://pypi.org/project/tweet-preprocessor/

[8] C.J. Hutto and E.E Gilbert. 2014. *VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text.* Eighth International Conference on Weblogs and Social Media (ICWSM-14). Retrieved December 10, 2019 from https://www.nltk.org/_modules/nltk/sentiment/vader.html

[9] Brendan Martin and Nicos Koufos. 2018. *Sentiment Analysis on Reddit News Headlines with Python's Natural Language Toolkit (NLTK).* Retrieved November 15, 2019 from https://www.learndatasci.com/tutorials/sentiment-analysis-reddit-headlines-pythons-nltk/

[10] Meena Rambocas and João Gama. 2013. *Marketing Research: The Role Of Sentiment Analysis.* Retrieved December 1, 2019 from https://ideas.repec.org/p/por/fepwps/489.html

[11] Joshua Roesslein. 2009. *Streaming With Tweepy.* Retrieved November 15, 2019 from https://tweepy.readthedocs.io/en/latest/streaming_how_to.html