TV Shows Muster and Analysis

Abhimanshu Mishra
Binghamton University
amishr11@binghamton.edu

Aditya Bhagwat
Binghamton University
abhagwa1@binghamton.edu

Sharvari Joshi
Binghamton University
sjoshi14@binghamton.edu

Vinit Bhosale
Binghamton University
vbhosal1@binghamton.edu

ABSTRACT

Online entertainment has become a crucial part of life in the modern age. In recent times, the amount of content and the number of content producers has increased exponentially. With such an increased consumption of content, everyone has opinions about different TV shows and water cooler discussions have become commonplace. People take to different social media forums to express what they think and connect with the world at large. This provides data scientists and researchers a unique opportunity to study the pulse of the public about a specific show easily. We analyse reviews and tweets talking about TV shows by performing sentiment analysis, topic modeling and social network analysis. We see that shows with positive reviews and chatter also have a higher rating. However, it is not necessary that a more talked about TV show is necessarily popular.

1 INTRODUCTION

Our project aims to collect data about TV Shows from different social media platforms like Twitter and IMDb like tweets and ratings on a large scale and draw patterns about the popularity of a show, it's influence on users from different age groups, the amount of engagement it receives based on retweets, likes and replies as well as how the show trends in different parts of the world. The data sources we have selected are IMDb and Twitter. IMDb is a movie/tv show rating and review website which is very popular around the world and is trusted in most circles for recommendations of content. Twitter is one of the largest social media forums in existence and has become the platform of choice for people to express their feelings and broadcast it on a universal scale.

To find the relation between IMDb and twitter we performed sentiment analysis, topic modeling and primitive social network graph modeling. First, we perform simple textual analysis on the reviews for the TV shows and find that the most frequent words are generic ones. Some of them do capture a particular aspect of the show. Through sentiment analysis, we found a strong correlation between IMDb rating and the opinions people posted online, even on a different forum. However, it is clear that there is no relationship between popularity of a show on social media and its ratings. Finally, through topic modeling, we are able to cluster reviews distinctly and also analyse the salient words which influence a cluster.

2 DATA SOURCES

2.1 Twitter

Twitter [1] is a micro-blogging and social networking service where users post and interact with messages, "tweets," restricted to 280 characters. Information can be shared using photos, videos and links as and when they're happening, enabling insightful and real-time search results. Tweets are fetched using the Official Twitter API which allows up to 50 requests per hour when the data fetched is from a percent of sampled stream of the total tweets collected in real-time. We use the bearer token for authentication, provided by twitter API when requested using the developer account. We filter and scale the search request according to the fields necessary to perform analysis and the user related data. Tweets collected are provided in JSON format which is collected and stored on our remote storage. Tweets collected may contain some noise or unnecessary data which can be easily removed before storage. We do so by storing the tweets locally in a python list where they're stored in a dictionary format thus creating a list of dictionaries where each entry in the list (a dictionary) represents an individual tweet received. The JSON response (tweet) consists of various fields like text field containing text of status update, user field containing information of user who posted the information, coordinates field containing Geographic information of the user, retweet count field containing number of times tweet has been retweeted, favorite count field containing number of times tweet has been liked by any twitter user. Based on the extracted field values, analysis can be made to decide whether a TV show has a good review or not based on the keywords in the tweet, the number of times it has been retweeted, the likes and replies it received and also the number of times it trended at a particular location in the world. This information also helps in determining how negative or positive tweet can easily influence other social media users in reviewing a TV show. Since the python script to perform this data collection is to be performed everyday at regular intervals of time. We use cron-job to achieve this task. This cron-job runs every hour with an interval of 10 minutes everyday. As the cron-job schedules the script to run after every 10 minutes, the stream is again refreshed so no duplicates arrive while fetching. At an approximate , it can be seen that around 250-350 meaningful tweets arrive after every call. So on an hourly basis we achieve up-to 1500-1600 tweets per hour, which approximates to be about 30,000+ tweet per day. The arrival of such amount Of tweets per day was monitored using the ROBO 3T application made for keeping a tab on the mongo Databases and also to query and retrieve data from it. As the data stored in the database was separated by TV shows into collections, every meaningful tweet of a show ended up getting into the correct collection in JSON format. We gathered around 229,000+ tweets up-to 26th November, 2020 and we stopped the data collection as the data collected was enough to derive meaningful insights. The major contributors to the twitter data collected were the TV Shows

like The Mandalorian , The Queen's Gambit, ,The Crown , The BlackList and The Schitt's Creek. The data collection saw an upward trend during the afternoon and night where the approximate data collection per hour would almost be increased by twice. The total data size of each collected tweet was 0.02 MB and each tweet collected varied in the incoming number of fields from about 14 to 16. We required a total VB is a stupid fucker storage of 1.5 GB for storing this data. After the collection process ended, the data was exported using the Studio 3T application to get the data stored on Drive storage. We consider this to be meaningful data since the data and noise cleansing has already been done.

2.2 IMDb

IMDb (Internet Movie Database) [2] offers an extensive database of information related to films, television programs, video games, and streaming content online – including crew details, plot summaries, trivia, fan and critical reviews. IMDb currently contains approximately 6.5 million titles (which includes movie and TV show titles),10.4 million personalities as well as 83 million registered users. Based on the launch of TV shows on worldwide streaming platforms, the popularity of these shows may vary with time which is one major factor of consideration for the task at hand [5]. After analysis we create a text file of popular Tv shows we then read each Tv show one by one and start collecting TV shows data by using OMDb API and scrapping IMDB website. Each call of OMDb API depends on either TV show title or the imdbID, we took the approach of using title of Tv shows to call the api and as a response we got all the attributes like "Title", "Year", "Rated", "Released", "Genre", "Writer", "Plot", "Language", "Ratings", "imdbId" and others. For user review of Tv show we scrape the imdb website. In scraping process, we encounter Load More button which loads all user reviews for the Tv show, so to get all the user reviews we used Selenium. Selenium automation was implemented to first click all the Load More button of the Tv show review webpage to load all the reviews. Once the automation is done, we then get the outerHTML of the webpage and convert it into lxml tree using lxml.etree.parse() then, with the help of xpath queries we parse the tree to fetch the exact node that will provide the user review data of the webpage and start storing those reviews. After storing all the reviews we append those with the OMDB api results and create a JSON collection object of each TV show which then get inserted into IMDb collection of MongoDB database on remote server and every TV shows associated data can be retrieved on demand for analysis.

3 DATA COLLECTION

We start by running a Daemon script on the virtual machine Written using cronjob which runs the python script every 15 minutes of an hour. This python script uses Twitter API to fetch the tweets of various TV shows in JSON format. The hashtags provided by us to the API results in the tweets to be fetched and will be stored in a list which creates list of dictionaries. Certain fields are extracted from this twitter data and those fields are stored into MongoDB into a single database with multiple collections. Each collection represents a different TV show in the database.

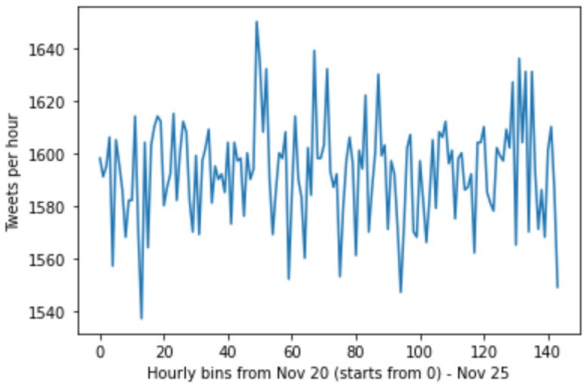

Figure 1: Hourly tweet collection over Nov 20-Nov 25

Figure 1 shows the number of tweets collected each hour in the given time period of November 20-25. It sums up to 229,000 tweets over this period of time.

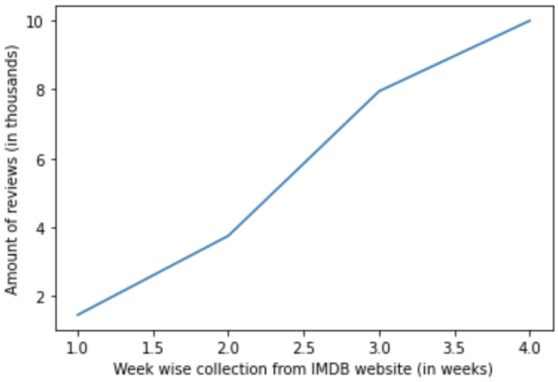

Figure 2: A graph showing the collection of data weekly from the IMDB website.

The above graph plots the number of reviews we collected from IMDb against a particular week. We expect to collect approximately 10,000 reviews from the IMDb website from which we estimate around 5k-6k to be meaningful. We would be able to obtain a fair graph from sample based on these estimations

4 DATA ANALYSIS

Source	Total Data	Clean Data
IMDb	25,228	25,228
Twitter	229,416	101,291

Table 1: Dataset statistics

We collected 25,228 reviews for the 19 TV shows we want to study, using the OMDb API [3]. Additionally, we scraped Twitter with hashtags related to these TV shows and amassed 229,416

tweets out of which 101,291 were unique. These statistics can be seen in Table 1.

On the reviews, we performed sentiment analysis and topic modeling. We also performed sentiment analysis on all the tweets.

4.1 IMDb reviews

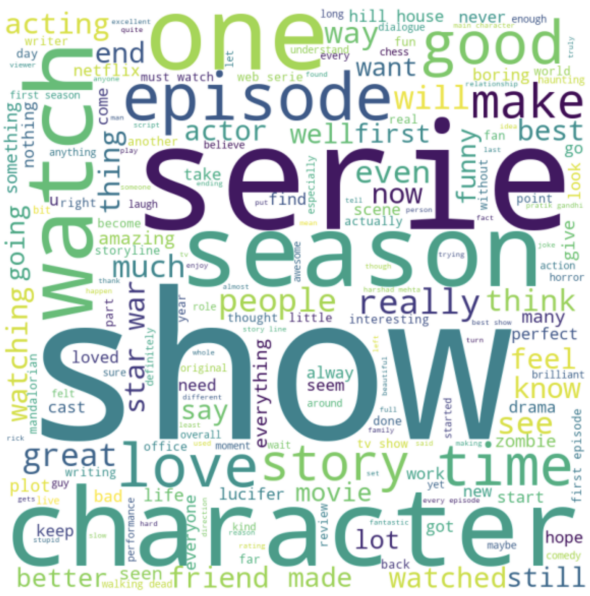

Figure 3: Word cloud of the text in reviews

In Figure 3, we show a wordcloud of all the reviews that have been accumulated. The most common words in the review seem to be reference to the shows themselves, common nouns for the same.

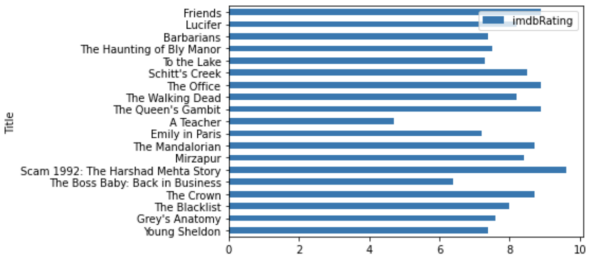

Figure 4: IMDb rating (scale of 0 to 10) of all TV shows

Figure 4 shows the comparative ratings of all the TV shows in consideration. We see that Scam 1992: The Harshad Mehta Story is rated the highest.

Figure 5 shows the average sentiment score of all the reviews available for a TV show on IMDb. We used TextBlob to generate a sentiment score for each text review. It is clear that Scam 1992: The Harshad Mehta Story has the most positive reviews. This is in line with the IMDb ratings. In fact, the correlation between ratings

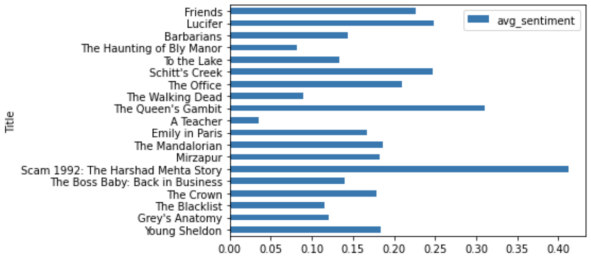

Figure 5: Sentiment scores (scale of 0 to 1) of IMDb reviews for all TV shows

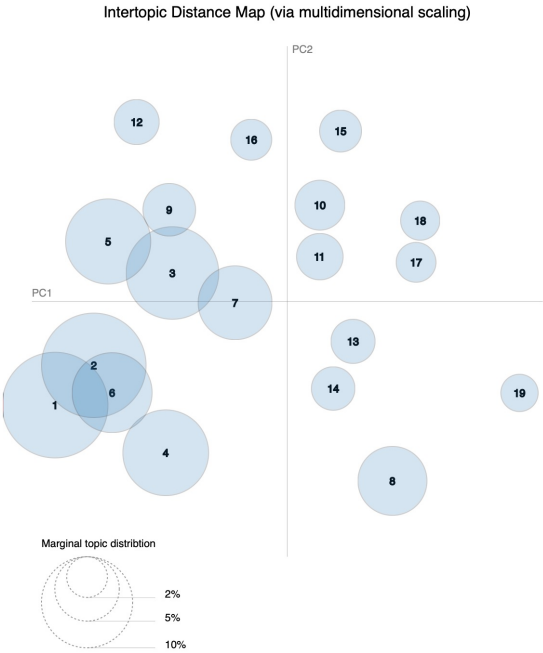

Figure 6: Topic clusters of reviews

and sentiment score is 0.7353, which shows that they are highly correlated.

We also train a topic model using LDA [4] on all the reviews and visualise the clusters it creates in Figure 6. Upon looking at the salient terms for each topic, we understand that only some topics are made up of distinct, non-function words. Others are influenced mostly by generic words like 'I' and 'The'. This shows that most reviews are built up of the same words and it is difficult to learn an effective topic model where clusters signify something tangible.

4.2 Tweets

Out of 101,291 tweets, 97,750 tweets are not tagged with a geolocation. We hypothesize that, despite the privacy issues of social media, people prefer to not share their location data and are mindful of it being used somewhere.

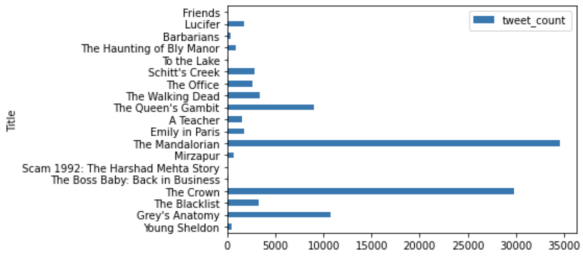

Figure 7: Number of tweets for each TV show

Figure 7 shows the number of tweets extracted for each TV show. It is clear that The Mandalorian is the most talked about show on Twitter. However, 'Scam 1992: The Harshad Mehta Story' barely has any chatter about it.

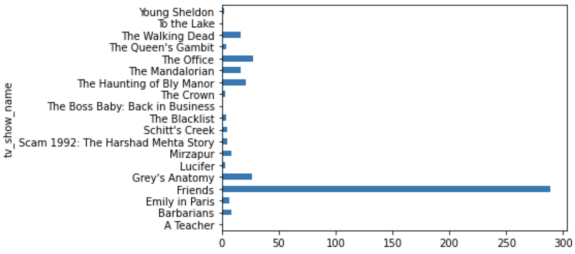

Figure 8: Popularity of tweets about a TV show

Figure 8 depicts the popularity of tweets about a particular TV show. The popularity of a tweet is calculated by adding the number of times it has been liked and the number of times it has been retweeted. This signifies that a user is likely to increase their standing by talking about 'Friends'.

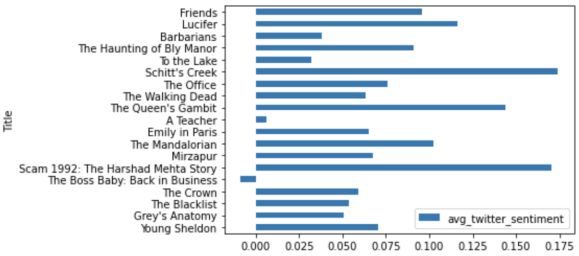

Figure 9: Sentiment score (scale of 0-1) of tweets about TV shows

Figure 9 shows the average sentiment score of all tweets grouped by TV shows. It is clear that Schitt's Creek is the favorite of the Twitterati. It is also interesting to note that 'The Boss Baby: Back in Business' is the only TV show that receives negative reviews overall.

The correlation between IMDb review sentiment and tweets' sentiment is 0.7948 . The correlation between IMDb rating and tweets' sentiment score is 0.7408. Both these numbers indicate a strong correlation between the respective pairs above.

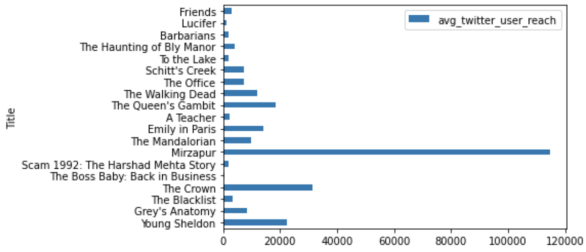

Figure 10: Reach of users talking about a TV show

Figure 10 shows the reach of the users who talk about a particular TV show. User reach is defined as the number of followers a user has. It can be seen that users' with a large following are talking about Mirzapur the most. However, despite this, Figure 7 shows us that there aren't a lot of tweets about that TV show in comparison to others.

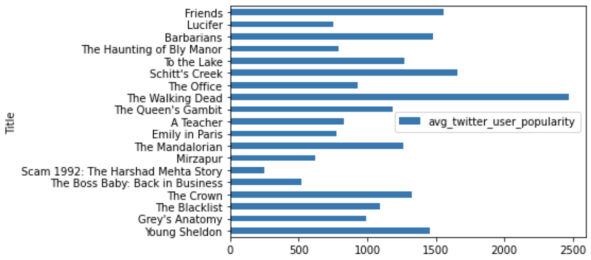

Figure 11: Popularity of users talking about a TV show

Figure 11 describes the popularity of users talking about a particular TV show. We interpret the number of friends a user has, i.e. the number of accounts they follow, to be their popularity. 'The Walking Dead' has the most popular users talking about it.

5 CONCLUSION

We built various pipelines to extract data from two social forums - IMDb and Twitter. After collecting a sufficient amount of data, we performed sentiment analysis, topic modeling and primitive social network graph modeling on it. We find a strong correlation between our sentiment analysis system and the IMDb rating for the TV shows, which also demonstrates the strength of simple, out-of-the-box systems. We build a topic model aimed at clustering tweets about one TV show in one cluster i.e. associating one topic with one TV show. However, due to the use of generic words in the review, we find that thought the clusters are almost distinct, they do not represent a tangible entity like a TV show. We also analyse how the popularity of a user affects the popularity of that show on Twitter. It is surprising to find no clear correlation between them. Going forward, it would be interesting to model the demographics of the people interested in a TV show to see which show appeals to whom.

REFERENCES

[1] [n.d.]. https://developer.twitter.com/en/docs/twitter-api
[2] [n.d.]. https://www.imdb.com/
[3] [n.d.]. http://www.omdbapi.com/
[4] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *J. Mach. Learn. Res.* 3, null (March 2003), 993–1022.
[5] Adam Nyberg. 2018. Classifying movie genres by analyzing text reviews. *CoRR* abs/1802.05322 (2018). arXiv:1802.05322 http://arxiv.org/abs/1802.05322