Sentimental Analysis on Arcane

Hirotaka Tokura

Introduction to Digital Humanities: Final Report
tokura_h@toki.waseda.jp

1 Introduction

Arcane is a Netflix original animation series, produced by Riot games which famous for its game League of Legends. Arcane world is set in a League of Legends universe focusing on certain characters. This was the first animated series produced by Riot games and by doing sentimental analysis on review of such project, it will allow us to see if they should keep on making such product or not.

2 Background

For this research will be using NRC Word-Emotion Association Lexicon ver 0.92 by Saif M. Mohammad and Peter D. Turney (1). This allows to see if the word has any association with either positive or negative sentiments, and by applying this to data, it will allow us to see if the overall feedbacks are negative or positive. Neutral values are also observed however for analysis section it is excluded. I have used twint to scrape data from twitter. Twint (Twitter Intelligence Tool) allows python to aceess tweet without using Twitter's API that can be installed on python and can scrape tweets that has certain hashtags. Which for this research to gather tweets that include #Arcane it was one of the best tool to be used. Method is largely influenced by the page (2).

3 Data and Method

For this research I have scraped 10,000 tweets from Twitter, using twint. Settings of scraping was done as shown on top. Settings are changed so it

```
"""
# Configuration of the twint
c = twint.Config()
c.Store_csv = True
c.Output = "Arcanesample.csv"
c.Search = "#Arcane"
c.Limit = 10000
c.Lang = 'en'
"""
```

Figure 1: Configuration of the twint

store data in CSV file which named as Aecanesample.csv. Has set the Search value to Arcane so it only includes tweets which talks about arcane. And set the language to en (English) to try and gather as many English tweets which will allow me to process the tweet easier. This data is then taken in by using Pandas. Stored as data frame. In the data frame there are still tweet with language other than English which I have removed and now amount of tweets have decreased from 10,000 to 3686 which seems to be very small amount but I did not want to do father scraping because repetitive scraping can become big trouble. NRC Emotion Lexicon are loaded with only extracting positive and negative entries. These are then scored as 1 and -1 with positive and negative respectively. Words that are not included in this list are scored as 0 and are counted as neutral word. Score are given to each tweet and saved in list. Tweet and score are saved in different list and then it is combined into one data frame. Now in data frame add original tweet and we can see which words has sentiment score and original tweet with score given to the tweet. On the right of figure 2 there is section which changes score into 1, -1 or 0 making depending on if the score is positive, negative or neutral. Father analysis is done on if this model of Emotion Lexicon is suited for

	Words	Score	Sentence	Binary score
0	[good]	1.0	Just finished season 1 of #Arcane. So good!!! ...	1
1	[]	0.0	Chirean oc! His name is Glitch n he's a member...	0
2	[effort, doodle]	0.0	low effort jinx doodle anyone? just playing wi...	0
3	[shot]	-1.0	"Me, miss? not by a long shot." reposting casu...	-1
4	[proud, rusty]	0.0	Don't mind me being proud of this even tho I'm...	0

Figure 2: Data frame of tweets

this data. Using TF-IDF vectoriser it let us indicate which words has high frequency and low frequency and by identifying that, it will allow us to see which of the words have more meaning than others. And by using TF-IDF vectoriser, it can evaluate classification using different models. Which are Bernoulli naive Bayes model, Linear Support Vector Classification models, and Logistic Regression model from Scikit-learn library. As Bernoulli naive model, it needs binary data, so Binary score comes in play. Exclude binary score with 0 and it will end up only with 1 and -1 meaning it has only 2 values.a For Linear Support vector it is done similar way as it is in binary. It makes virtual line to distinguish which values are higher than the line or lower. Logistic regression is again uses Bernoulli distribution to build a model.

4 Results

```
print(max(binarylex_df['Score']))
print(min(binarylex_df['Score']))
```

```
7.0
-7.0
```

Figure 3: Maximum and minimum of score

Minimum value and max value has turned out to be 7 and -7 which is big in range. Figure 4 shows frequency against scores and as it shows there are

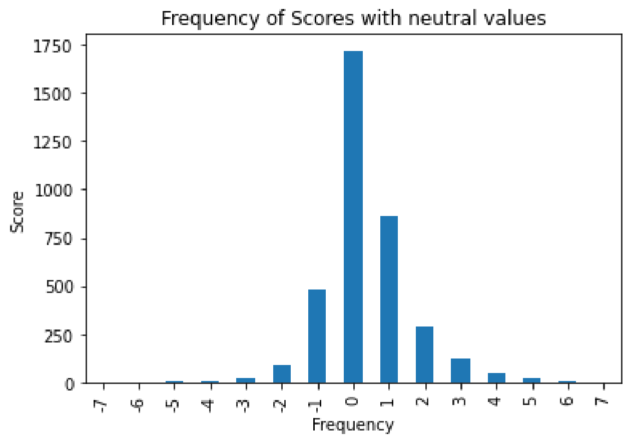

Figure 4: Bar graph with neutral value

lot of tweets with 0 scores and although these tweets may contain information for analysis to be carried we must at first ignore these values to see amount of positive and negative values we have obtained.

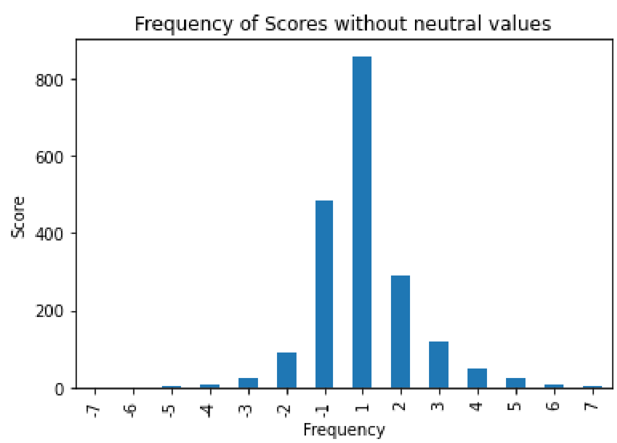

Figure 5: Bar graph without neutral value

And on figure 5, it has excluded bar with 0 and as it shows, there are more positive values than negatives and mode value here is 1 with 860 tweets. Second most is -1 with 484 and third with 2 and has 290 tweets.
Average of this data turned out to be 0.37 which is positive but because there

```python
avrscore = sum(score_list) / len(score_list)
counts = 0
total = 0
for i in score_list:
  if i != 0:
    total+=i
    count+=1
zerolessavr = total/count

print(f'Average is {avrscore : .2f}') #calculates avearge here

print(f'Average with out neutral is {zerolessavr : .2f}')
```

```
Average is  0.37
Average with out neutral is  0.70
```

Figure 6: Average value which top is with 0s and bottom without

are lots of 0s. By excluding 0 we get average if 0.7 which again is positive value. And so here we can assume that there are overall positive reviews than the negatives.

With first try because I have replaced url and #tags with URL tag and @user with USER they have showed up commonly.By removing them we get

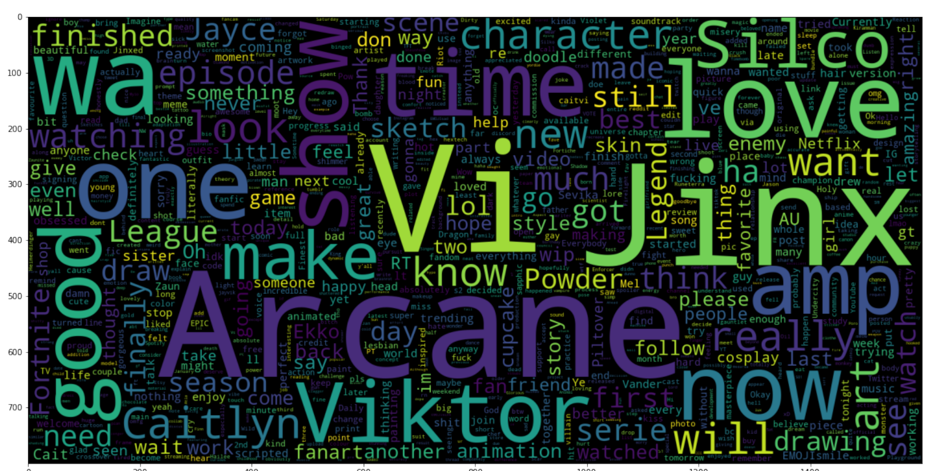

Figure 7: Word cloud with raw words

Word cloud on figure 7 which we can clearly see common word which some are name of characters but we can also see positive words such as good and love being commonly used in tweet and for last;Word cloud above which we can clearly see common word which some are name of characters but we can also see positive words such as good and love being commonly used in tweet and for last;Word cloud above which we can clearly see common word which some are name of characters but we can also see positive words such as good and love being commonly used in tweet and for last;

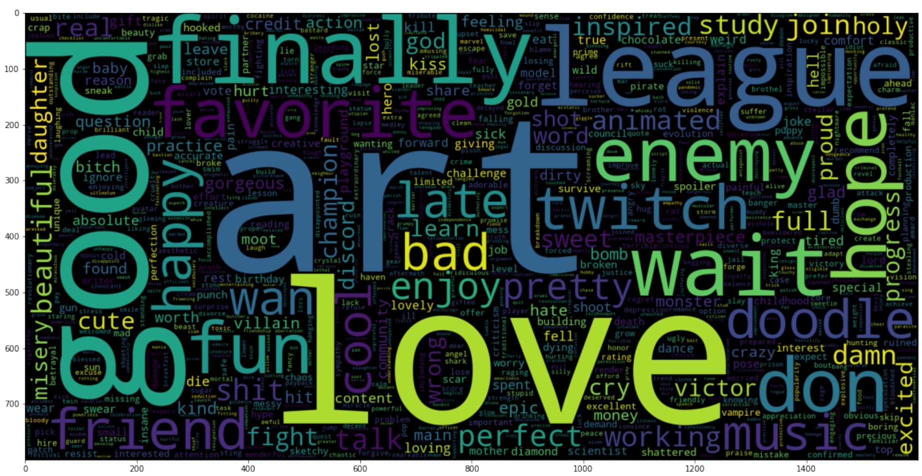

Figure 8: Word cloud with sentiment words

Word cloud with only words that has sentimental values. We can clearly see love and good taking big space meaning it is commonly used with other words such as art being used. there are also negative words as well such there is bad we can clearly see in the middle for example. There are also words that could be related to the game League of Legends which there is league and twitch that is noticeable however, it is uncertain that if those words are not related to sentimental scores and it would take way too long to go through every single tweet to examine the words so for now it will stay in the list.

Moving on to using TF-ID vectoriser. By removing data with 0 scores we are left with 1965 samples.

	precision	recall	f1-score	support
-1	0.75	0.11	0.19	27
1	0.75	0.99	0.85	72
accuracy			0.75	99
macro avg	0.75	0.55	0.52	99
weighted avg	0.75	0.75	0.67	99

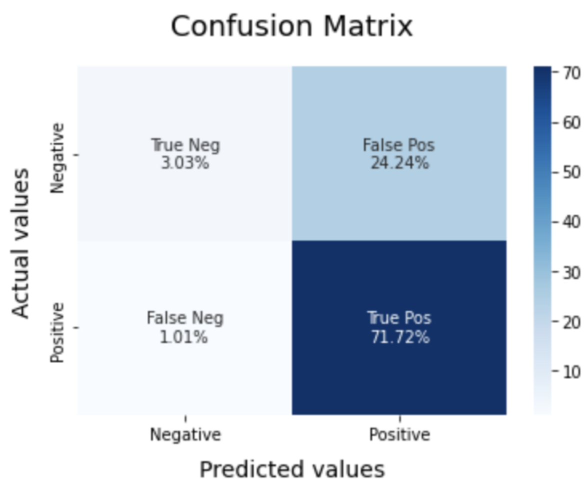

Figure 9: Confusion matrix with Bernoulli naive Bayes model

First with Bernoulli naive Bayes model. With this model previsions are 0.75 for both negative and positive with samples that are more into positive area. This is because sample includes more sentiment with positives rather than negatives which leading skewed confusion matrix to be created. We can also see that precision score are same for both negative and positive values. By having around 95% in positive areas this suggest the data may not be suitable to be used for this confusion matrix.

	precision	recall	f1-score	support
-1	0.90	0.67	0.77	27
1	0.89	0.97	0.93	72
accuracy			0.89	99
macro avg	0.89	0.82	0.85	99
weighted avg	0.89	0.89	0.88	99

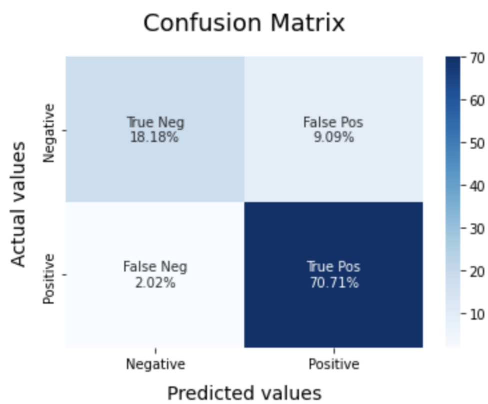

Figure 10: Confusion matrix with Linear support vector classification model

Next is with Linear Support vector classification model. With this one accuracy score turned out to be higher than the previous one with having same amount of samples. Again percentage in confusion matrix is skewed towards positives but this time there are less false positives and negatives. Where percentage of true positive did not change but has increased numbers in true negatives.
And for last with Logistic regression model on figure 11 below. For this model, accuracy here is lower than last one with sharing similar numbers in matrix. As these three analysis done by different models have processed through skewed samples which could lead to having biases in numbers. And indeed with confusion matrix there larger number in confusion matrix so I have made another try with balanced sample but this meant less samples and that leads to having less accuracy.

```
LRmodel = LogisticRegression(C = 2, max_iter = 5000, n_jobs=-1)
LRmodel.fit(X_train, y_train)
model_Evaluate(LRmodel)
```

```
              precision    recall  f1-score   support

          -1       0.93      0.48      0.63        27
           1       0.84      0.99      0.90        72

    accuracy                           0.85        99
   macro avg       0.88      0.73      0.77        99
weighted avg       0.86      0.85      0.83        99
```

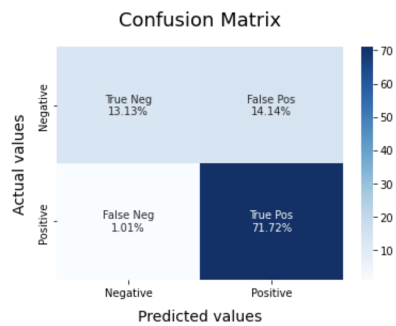

Figure 11: Confusion matrix with Logistic Regression model

```
              precision    recall  f1-score   support

          -1       0.73      0.82      0.77        68
           1       0.72      0.60      0.65        52

    accuracy                           0.73       120
   macro avg       0.72      0.71      0.71       120
weighted avg       0.72      0.72      0.72       120
```

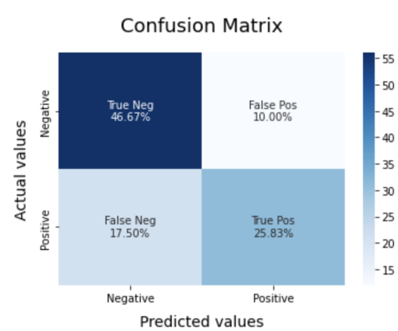

Figure 12: Confusion matrix with BNB model with balanced sample

This time I have balanced numbers so there are 600 samples which includes positives and 600 with negatives with total of 1200 samples. I have picked these samples from original using random variable.

BNB model with balanced sample on figure 12. Here precision is 0.73 which is lower than skewed sample but it is expected. As confusion matrix shows there are lot more true negatives with close to half of the samples are in True negatives. True positive has decreased by lot. By balancing out number of samples visualisation with confusion matrix comes in play giving clear idea what data may look like.

	precision	recall	f1-score	support
-1	0.81	0.75	0.78	68
1	0.70	0.77	0.73	52
accuracy			0.76	120
macro avg	0.76	0.76	0.76	120
weighted avg	0.76	0.76	0.76	120

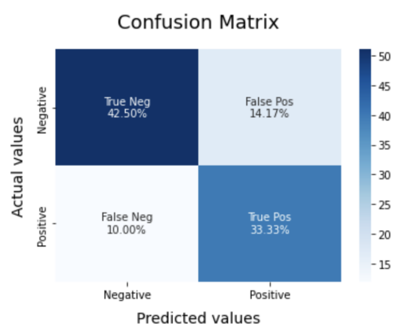

Figure 13: Confusion matrix with Logistic Regression model with balanced sample

Weirdly result for Linear SVC model and Logistic Regression Model were the same. With having accuracy score of 0.76 which is higher than the BNB model. This time there are more true positives and false positives. Amount of True negatives have not changed much but false negative has decreased. Meaning this model has judged that some are rather in positives in negatives compared to that in BNB model.

5 Conclusion and Discussion

By doing this analysis we can know how well NRC word lexicon sentiments are by comparing with other 3 models. And in the end 0.73 accuracy score is quite decent score for sampling and there are problems with this way of scoring as it only checks for words, which if there are words in front or after which could make the meaning opposite to what it is intended. This sample also ignores score with 0 this include if the tweet has word with score of 1 but it also has word with -1 score which canceling out the score leading to 0. Doing this analysis will let us know how well NRC word lexicon sentiments are by comparing with other 3 models. And in the end 0.73 accuracy score is quite decent score for sampling and there are problems with this way of scoring as it only checks for words, which if there are words in front or after which could make the meaning opposite to what it is intended. This sample also ignores score with 0 this include if the tweet has word with score of 1 but it also has word with -1 score which canceling out the score leading to 0.

6 Future Work

There are lots of future works to be done here. Number of samples were very small, although it had 10,000 samples at start, by extracting out useful data we were only left with 1200 samples to be analysed. As noted before, this type of analysis will only look though words that are used in sentence instead of giving score to whole sentence. Which lead to some samples having score of 0 even tho it included words with sentiment scores. Sentence based sentiment score could be applied instead of word based for the next research.

7 List of tools and methods used

This section lists all the tools and methods used.

7.1 Tools and software

- Python3
  - NLTK
  - re

  - whatthelang
  - twint

- nest-asyncio
- pandas
- sklearn
- random

7.2 Regular expressions

- To find user tags: \’@[\ˆs]+’
- To find hashtags : \r”#[\ˆs]+”
- TO find URL : \r”((http://)[ˆ]*—(https://)[ˆ]*—(www˙)[ˆ]*)”

References

[1] Saif M. Mohammad and Peter D. Turney, NRC Word-Emotion Association Lexicon ver 0.92

[2] Nikit Periwal, Twitter Sentiment Analysis for Beginners, (2021)