Twitter Movie Sentiment

Jack Lotkowski, Gedalia Kott, Paige Nelson, Danny Shaw

May 11, 2015

ABSTRACT: Movie opinions of the general population are somewhat hard to come by. In this paper, we look at a way to take advantage of the numerous movie opinions posted on Twitter. Using our own classifier, we evaluate tweets about movies as positive, negative, or neutral, and determine an aggregate score for each movie based on the tweets about it. This score can then be compared to the critic score of that movie to get an idea of how effective our tool is. In addition, we look at how SentiWordNet [1] can be used to help in the analysis of tweets as positive, negative, and neutral.

Keywords: Twitter*;* Sentiment Analysis*;* Movie Reviews*;*

1 Introduction

Our current standard for what makes a good movie review is based on critic's opinions and those who avidly participate in the movie review community. The most popular movie review sites follow this standard: Rotten Tomatoes solely shows critic reviews and account member reviews and IMDB solely shows account member reviews. These reviews are based on the opinions of a small portion of the population and are oftentimes delayed. Meanwhile, we have an enormous amount of people tweeting their opinions of various movies immediately after leaving the theaters. Tweets such as "Boyhood was much better than I expected" and "I don't understand why people like Big Hero 6" provide insight into the general population's opinion of a given movie. By aggregating these opinions into one average percent positive score, we can provide a real time "score" for a movie that represents the opinions of the many Twitter users viewing that movie.

Our general method involved three datasets: a list of movies, the thousands of tweets about these movies, and the movies' Rotten Tomatoes scores. We started by aggregating the tweets about each movie. Using our own classifier, we determined if each of these tweets, in general, was positive or negative. After marking each tweet and ignoring neutral tweets, we determined what percent of the tweets were positive. This percent gave a score for the movie that we were the able to compare to the Rotten Tomatoes movie score, which gives a percent of positive critic reviews. To take our project further we ran SentiWordNet [1] Analysis on our dataset of tweets. SentiWordNet evaluates terms on a spectrum of positive, negative, and objective. By evaluating tweets with SentiWordNet's dictionary, we were able to get another score for movies that represent the positivity of tweets about that movie.

2 Method

2.1 Data Sets

To test our classifier on tweets from Twitter we needed:

- Lists of movies to get tweets about
  - 100 movies ranked as good
  - 100 movies ranked as bad
  - 100 movies that were recently released
- Tweets on each of the above to classify
  - 1,000 tweets per movie
- Pre-determined Rankings of the movies from the above lists for comparison with the classifiers results

This makes for a total of 300 movies and their rankings and 300,000 tweets.

2.1.1 Movies and Rankings Data

We collected the movies from Rotten Tomatoes and the Internet Movie Database (IMDb). Rotten Tomatoes and IMDb had lists of movies ranked by categories matching our exact needs: good, bad, and recent. Our sources are included in our datasets, where we list the URLs used to obtain the movie titles and their rankings. We scraped the HTML for the movie names and their associated scores. We then saved the data to text files in JSON format for later processing. Scores from the individual sites were in different scales. IMDb ranks movies on a scale of 1 to 10 while Rotten Tomatoes ranks movies by percentages. We normalized all scores to be on a scale of 1 to 10.

2.1.2 Twitter Tweets Collection

We collected 1,000 tweets per movie so we could classify the movie with our own classifier. This collection process was accomplished using the Twitter REST API (GET search/tweets). We provided the same style of query for each movie in our requests. The following is the query used to collect tweets on Ex Machina:

```
#ex_machina OR @ex_machina OR ex_machina OR #exmachina
OR @exmachina OR exmachina OR ex machina OR Ex Machina
...
OR "#Ex_Machina" OR "@Ex_Machina" OR "#ExMachina"
OR "@ExMachina" OR "ExMachina"
```

The code that generates the query is provided with the project. The queries were tested informally to see what would provide the best results without skewing the results for any particular movie. We found this to be the best option. Further testing and post-processing/analysis of tweets can be utilized in the future to get better tweets and remove tweets that do not actually reference the movie. Tweets found were stored in JSON format for later consumption by the classifier. Only recent tweets in the English language were requested being that we can only use English text and we wanted to avoid pulling popular tweets intentionally so they wouldn't skew the data.

2.2 Classifying Tweets

In addition to gathering tweets, we also worked on creating a classifier to analyze and differentiate between tweets which had positive, negative or neutral sentiment toward a movie. (Although we used the 2011 article, Target-dependent Twitter Sentiment Classification [2] as inspiration for our project, our methods deviated significantly as we attempted to produce a working prototype for this project).

2.3 Wrangling Training/Testing Data

To train our classifier, we opted to use the training/testing approach to teach our classifier. Since this problem was not researched in the past, training on a nicely curated set of tweets about movies was not an option. Instead we researched different datasets related to movies and narrowed it down a dataset from the data science website Kaggle. This dataset featured 221,000 movie reviews from RottenTomatoes broken down by sentence and labeled using crowdsourcing by Amazon's Mechanical Turk. Each labeled entry ranged in length from one word to no more than a sentence, allowing our classifier to become more adept at spotting small clues present in tweets which may indicate sentiment. After writing a script to split and reformat the data we ended up with a 3 class (positive, negative and neutral), and 5 class (positive, somewhat positive, negative, somewhat negative and neutral) data files, each with 156,000 training reviews and 65,000 testing reviews. The final format was as follows:

```
3   occasionally
3   amuses but none of which amounts to much of a story
5   amuses
1   but none of which amounts to much of a story
3   but
1   none of which amounts to much of a story
3   none
```

2.4 Building the Classifier

To build the classifier, we looked into a couple implementations of machine learning, focusing on NLP, to get the greatest accuracy we could: Vowpal Wabbit and MetaMind. After attempting to work with Vowpal Wabbit it was clear we needed a more approachable solution for the project given our overall lack of knowledge into the field of ML: this where MetaMind came in. MetaMind is a machine learning API developed by Stanford PhD graduate Richard Socher and developed by such minds as Ruslan Belkin, formally of Salesforce, RelateIQ and Twitter. MetaMind has done extremely well in text classification: scoring a 83.4 Pearson Correlation Coefficient Score in the SemEval 2014 Task 1 competition [3] beating out top attempts by University of Texas, Stanford and Illinois. Building on this, we wrote an application in Python which utilized the MetaMind API to create a classifier from our

RottenTomatoes training data. Testing on our testing dataset showed our classifier predicted at 67% accuracy.

3 Experimental Results

3.1 Classifying Tweets

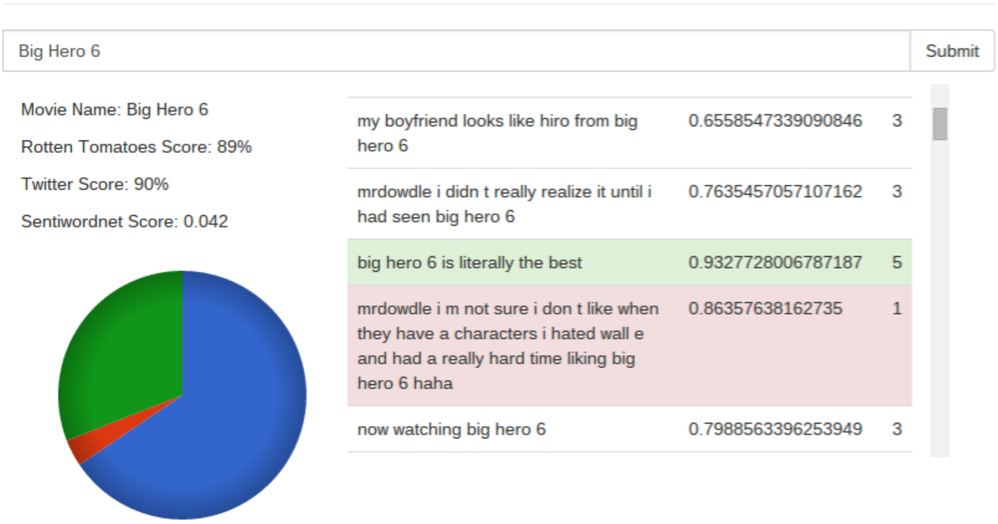
To run our newly created classifier on our library of tweets we wrote a script to reformat and standardize our tweets using RegEx to remove irrelevant and non-standard compliant characters. Additionally, our culminating program consolidated and fed tweets into our classifier. After a few hours, we had JSON formatted output of tweets along with our predicted sentiment formatted as follow:

```
[
  {u’user_value’: u"Furious7 was the worst movie I’ve ever seen.
   Period.", u’probability’: 0.819143650919281, u’label’: u’1’},
  {u’user_value’: u’I loved Skyfall. Brilliant!’, u’probability’:
    0.9619668949957287, u’label’: u’5’}
]
```

With this data we were able to calculate a final metric of whether twitter users “rated” a movie as good (mostly positive) or bad (mostly negative).

3.2 Analyzing Results

After collecting the necessary data and tweets, we created a web application which can be found at `http://leonardowater.herokuapp.com/twitter-sentiment`. Our web application allows users to search for a movie and after selecting a movie, the corresponding tweets are shown with positive tweets highlighted in green, negative tweets highlighted in red, and neutral tweets not highlighted. Along with the tweets, our web application displays the Rotten Tomatoes score, our own computed Twitter Score, our computed SentiWordNet Score, as well as a dynamically-generated pie chart that visually represents the distribution of positive (green), negative (red), and neutral tweets (blue).



Movie Name: Big Hero 6

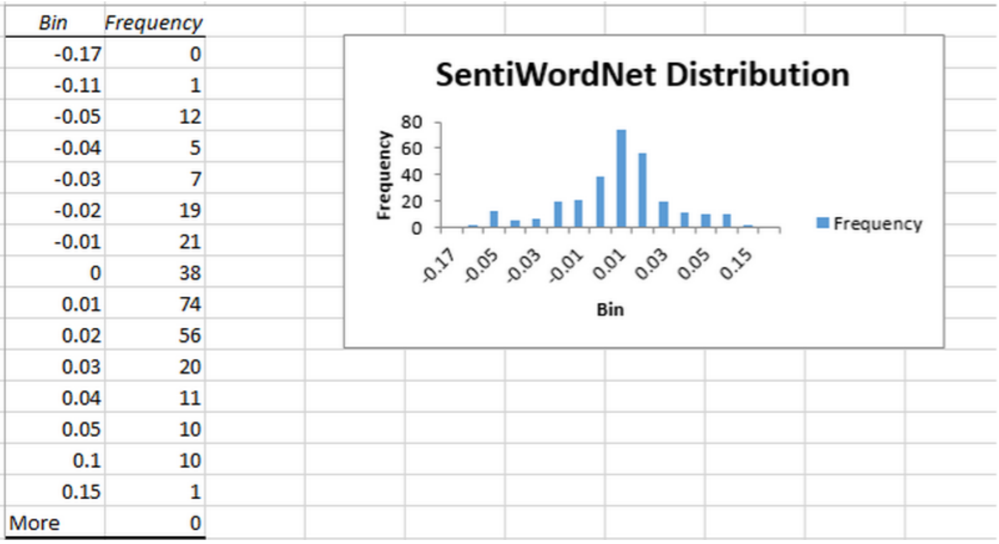
Rotten Tomatoes Score: 89%

Twitter Score: 90%

Sentiwordnet Score: 0.042

my boyfriend looks like hiro from big hero 6	0.6558547339090846	3
mrdowdle i didn t really realize it until i had seen big hero 6	0.7635457057107162	3
big hero 6 is literally the best	0.9327728006787187	5
mrdowdle i m not sure i don t like when they have a characters i hated wall e and had a really hard time liking big hero 6 haha	0.86357638162735	1
now watching big hero 6	0.7988563396253949	3

To calculate a movie's SentiWordNet score, we first created a hash that mapped words to their respective SentiWordNet sentiment score. Next, we iterated through the tweets of each movie and found the weighted average of all of the words in the tweet and eventually found the weighted average of the sentiment scores of the entire collection of tweets of a specific movie. Words that aren't identified by the SentiWordNet dictionary were disregarded in our algorithm. After finding the SentiWordNet scores of every movie, we created a score distribution histogram which is shown below. Scores of 0 represent objectivity while positive scores represent positivity and negative scores negativity. Interestingly enough, the SentiWordNet scores tended to be very close to 0 with a slight edge towards positivity. Although some of the SentiWordNet scores seemed to reflect what Rotten Tomatoes and our own Twitter score showed, there did not appear to be an obvious correlation between the SentiWordNet scores which may have been attributed to the fact that many of the words in tweets were slang and not actually words which would've been found in the SentiWordNet word network/dictionary.



Bin	Frequency
-0.17	0
-0.11	1
-0.05	12
-0.04	5
-0.03	7
-0.02	19
-0.01	21
0	38
0.01	74
0.02	56
0.03	20
0.04	11
0.05	10
0.1	10
0.15	1
More	0

References

[1] S. Baccianella, A. Esuli, and F. Sebastiani Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. *Proceedings of the 7th conference on International Language Resources and Evaluation (LREC '10)*, 2200-2204, 2010.

[2] L. Jiang, M. Yu, M. Zhou, X. Liu, T. Zhao Target-dependent Twitter Sentiment Classification. *Proceedings of the 49th Annual Meeting of the Association for ComputationalLinguistics (ACL-2011)*, 2011.

[3] SemEval-2014 Task 1. *http://alt.qcri.org/semeval2014/task1/*, 2014.