Project Final Report: Movie on Tweets

EECS 349 Machine Learning

Northwestern University

Zhe Chen, Chenxing Wu, Hanhong Lu

1. Introduction

Movie on Tweets is a IMDb movie score predictor, using movie's tweets retrieved from Twitter. Our task is to analyze the feedback of a movie (either positive or negative) based on tweets that have mentioned the movie, for example, using hashtags.

Twitter is one of the most popular social media where people post comments about their everyday life. Users share their reviews about a movie on Twitter which possibly encourage or discourage their followers' motivations of watching the movie. In addition, because of its huge influences, many movies have started to include a hashtag in their trailers that tell people to tweet about them to attract social attention. Therefore, Twitter has become a great platform to examine people's feedback on a movie. By analyzing the sentiment of tweets by people who have watched the film, a general review of the movie can be made. We believe that this piece of information will be a helpful guideline.

2. Experiment

I. System Training

We use Large Movie Review Dataset from Stanford University http://ai.stanford.edu/~amaas/data/sentiment/ as our training dataset. It contains 50,000 movie reviews, 25,000 positive and 25,000 negative. We randomly chose 1,000 positive reviews and 1,000 negative reviews to train our naive bayes classifier. To improve the classifier, we then tried to increase our training dataset by adding another 2,000 positive and negative reviews.

We built our system using naive bayes algorithm. Each example is stored in one txt file in the form *review-id_number-of-stars.txt*. We extract the filename of each review to categorize them into positive and negative examples. To simplify the problem, we define IMDb score from 0 to 4 stars as negative, and 7 to 10 starts as positive. We did not include neutral reviews to achieve better training result. Each review is then segmented into single words. The word's frequencies in positive files and negative files are recorded. We also added another feature, bigram, to our system to compare the performance.

Here is the performance result on two systems using 10-fold cross validation.

Table 1. Accuracy of training result

Features		Single Word	Single Word + Bigram
Positive	Precision	0.855773972033	0.920307652833
	Recall	0.888353508419	0.797222480181
	F-measure	0.870217714806	0.853164955523
Negative	Precision	0.882733702769	0.821094050964
	Recall	0.85220049602	0.931654782064
	F-measure	0.865545492618	0.871933113784

II. Testing

We retrieved tweets comments of 3 different movies, *Kingsman* as one recent movie, *Fifty shades of grey* with a low IMDb score, *The Imitation Game* with a high IMDb score, using Twitter API to be our test dataset. Each movie has at least 2000 examples. The basic attributes are single words from each review, and we also tried adding bigram as another attribute.

Data processing: In python, we use language identification module, langid (https://github.com/saffsd/langid.py), to filter languages other than English. Then we got rid of samples with links and "RT" (retweet sign) to avoid duplicated tweets and unnecessary information, such as promotion tweets and ads. After filtering, we finally got around 500 samples for each movie.

Table 2. Number of data before and after preprocessing

	Before preprocessing	After preprocessing
Kingsman	2000	403
Fifty Shades of Grey	2000	681
The Imitation game	2000	433

3. Result

After applying our system to three tweets datasets, we get the number of positive and negative tweets on two systems. Results are shown in the following tables.

Table 3. Testing result with 2,000 training examples

	Single word		Single word + bigram	
	Positive	Negative	Positive	Negative
Kingsman	230	173	189	214
The Imitation Game	229	204	188	245
50 Shades of Grey	258	423	255	416

Table 4. Testing result with 4,000 training examples

	Single word		Single word + bigram	
	Positive	Negative	Positive	Negative
Kingsman	253	128	182	221
The Imitation Game	294	139	181	252
50 Shades of Grey	68	613	167	514

Table 5. Movie Evaluation with real IMDb score

	Predicted score		Actual score from IMDb
	2,000 test examples single / single+bigram	4,000 test examples single / single+bigram	
Kingsman	5.63/5.04	6.15/5.01	7.9
The Imitation Game	5.41/4.83	6.23/4.80	8.1
50 Shades of Grey	4.59/4.47	3.04/3.85	4.2

4. Conclusion

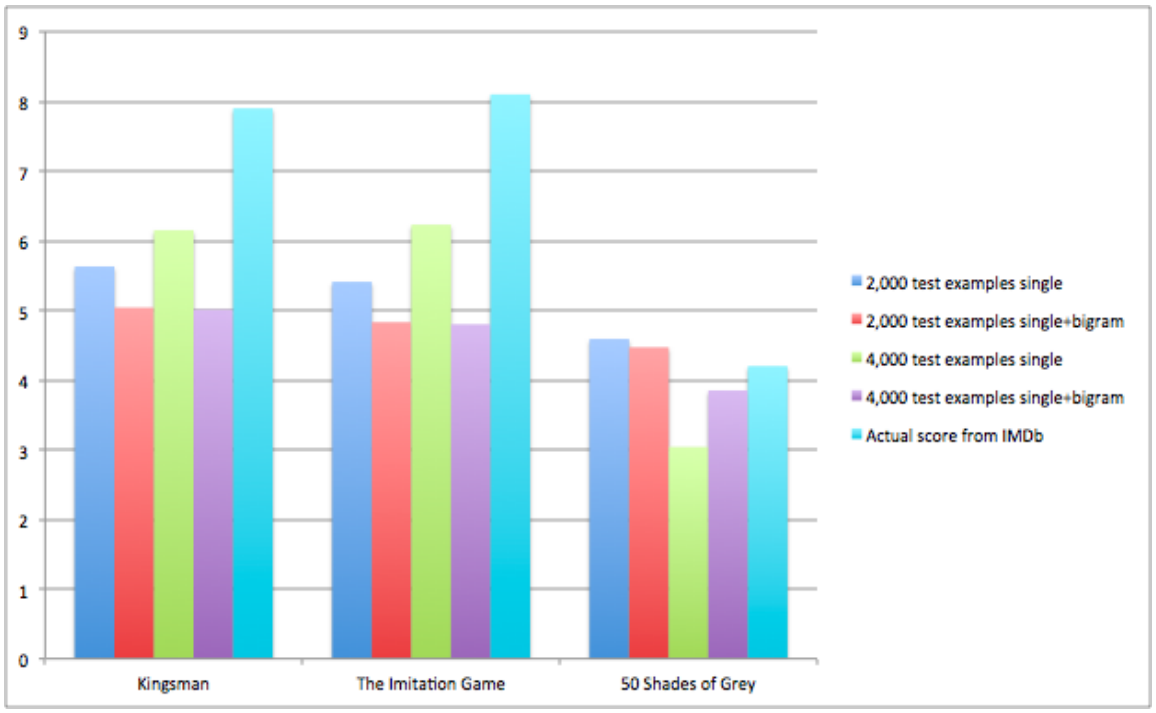

Figure1. Histogram of three movie evaluation

The Naive Bayes Network Classifier has relatively good performance. In comparison, the systems perform better with more testing examples. Adding bigram feature does not improve the system as we expected. Its performance actually becomes worse the only using single word. This is because when adding features, most of them are useless. These irrelevant features create noise to the classifier, and thus not improving the accuracy.

Another problem is that we cannot get more than 2,000 tweets due to the Twitter API's rate limitation. The lack of testing examples may decrease the accuracy of Naive Bayes Network Classifier.

5. Future Work

In the future, we would like to implement logistic regression algorithm to build our system. Logistic regression may have better performance on sentiment analysis. Another thing we can try is finding better attributes to predict, for instance, stems, number of capitalization, punctuation, or we can add weights to certain words to improve performance.