Sentiment Analysis Using Machine Learning Technique on Movie Review and Public Tweets

605.433 Social Media Analytics

Final Paper

Ko, Chin-Ting
Johns Hopkins Engineering for Professionals
Computer Science Master Program,
Whiting School of Engineering, Baltimore, United States.
Email: cko17@jhu.edu

***Abstract*—In this paper, I used numbers of sentiment classifier to perform sentimental analysis on movie reviews. The techniques used includes python programming language with NLTK library and Twitter REST APIs. The datasets includes IMDB movie review, and public tweets from Twitter. I also compared different machine learning tools and data sets to perform the best efficiency sentiment analysis results, and used sentiment analysis result to predict box office sales of movies.**

Keywords—Sentiment Analysis, Twitter, Machine Learning, Text Classification, data analysis.

I. INTRODUCTION

Sentiment analysis refers to uses of natural languages processing techniques to extract information from source materials. Sentiment analysis determined the attitude of speaker or writer for specific topic. Sentiment analysis from social media applied widely for variety of applications. This paper we focus on movie reviews and predict box office sales of movies and public tweets from social media platform Twitter.

II. DATASET

In this project, I used two different training data sets: large movie review dataset v1.0 and Sentiment140 corpus. The reason is to compare training set difference impact on classier accuracy rate. The test data is from large movie review dataset, and public tweets from Twitter REST APIs. The details of each datasets are explained below:

A. Large Movie Review Dataset v1.0

http://ai.stanford.edu/~amaas/data/sentiment/

This dataset contains movie reviews as benchmark for sentiment classifications. It contains 50,000 reviews split evenly into 25k training and 25K test sets. The overall distributions is balanced 25K positive reviews, and 25K negative reviews. No more 30 reviews are allowed for any given movie.

B. Sentiment140 Corpus

http://help.sentiment140.com/for-students

This Sentiment140 dataset is uniques because the training data was automatically created, as opposed to having humans manual annotate tweets. This dataset is a CSV format with emoticons removed. Data file format has 6 fields:

0 - the polarity of the tweet (0 = negative, 2 = neutral, 4 = positive)

1 - the id of the tweet (2087)

2 - the date of the tweet (Sat May 16 23:58:44 UTC 2009)

3 - the query (lyx). If there is no query, then this value is NO_QUERY.

4 - the user that tweeted (robotickilldozr)

5 - the text of the tweet (Lyx is cool)

C. Public Tweets from Twitter REST APIs

https://dev.twitter.com/rest/public

The dataset is from public Tweets from Twitter REST APIs. The formats is JSON file. The data is queried by movie title (year).

III. DATA PREPROCESSING

To process the data I collected and build a model for sentiment classified, I used Python 2.7 with NLTK (Natural Language Toolkit) library. Before performing any training algorithm for sentiment classifier, cleaning the data is necessary to remove any irrelevant information. It is also improve the accuracy result and computing efficiency.

A. Cleaning the data

The movie review dataset does not serve as sentiment analysis use. It contains punctuations and html tag which I decided to remove. If the word length is less than three, it is also removed.

I also tried to apply stemming techniques snowball stemmer (Porter stemming algorithm) to replace words with

it's root. This is to improve classification accuracy for sentiment analysis. More information is available http://snowball.tartarus.org/

Example code:

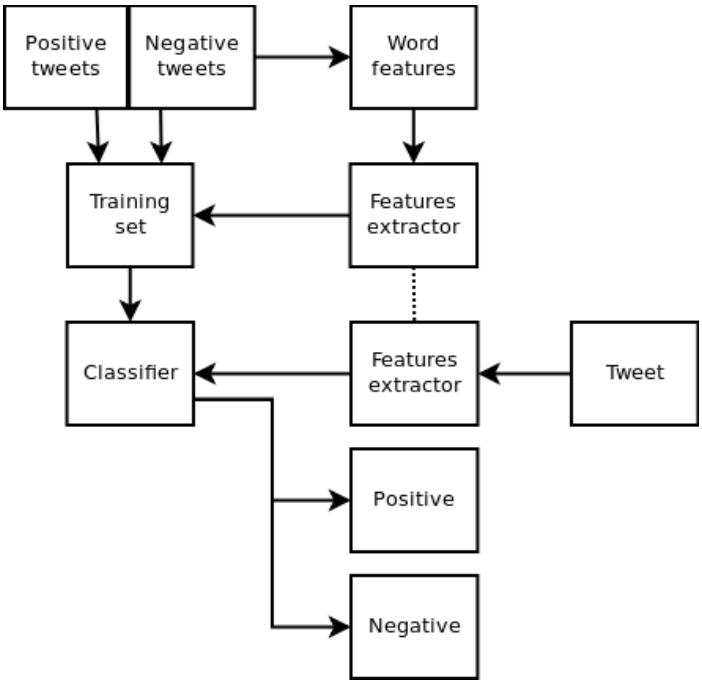
```
def do_stemming(filtered):
    stemmed = []
    for f in filtered:  stemmed.append(SnowballStemmer('english').stem(f))
    return stemmed
stop_words = set(stopwords.words('english'))

for (words, sentiment) in neg_tweets:
    words_filtered = [e.lower().decode('ISO-8859-1')
        for e in words.split() if ((len(e) > 3) and e not in stop_words)]
    words_filtered_stemmed = do_stemming(words_filtered)
    neg_tweets_filtered.append((words_filtered_stemmed, sentiment))
```

B. *Model Concept*

This sentiment classifier model has to exacted word features from the movie review dataset. It is a list with distinct words ordered by term frequency. The 'training set' is a tuples with each topple contain feature and sentiment string. Then I feed the features to train the sentiment classifier.

The sentiment features extract model is reference from http://www.laurentluce.com/posts/twitter-sentiment-analysis-using-python-and-nltk/



Mode Concept Diagram Summary

IV. CLASSIFIERS

I implemented two popular machine learning classifier: Naive Bayes classifier and SVM classifier.

A. *Naive Bayes Classifier*

Naive Bayes methods are a set of supervised learning algorithm based on Nayes theorem with naive assumption of independence between every pair of features. Since Naive Bayes by its nature assumed all the features are independent, it is expected may miss some of the dependencies which leads to less accuracy.

Example code:

```
nb_classifier =nltk.NaiveBayesClassifier.train(training_set)
print 'NB Accuracy:', nltk.classify.accuracy(nb_classifier, testing_set)* 100, "%"
```

B. *SVM Classifier*

Support vector machine (SVM) is one of the most common models for classification. SVM is widely used to solve text classification. Since it consider dependency, it is expected perform better result over Naive Baye classifier.

Example code:

```
svm_classifier = nltk.classify.SklearnClassifier(LinearSVC()).train(training_set)
print 'SVM Accuracy:', nltk.classify.accuracy(svm_classifier, testing_set)* 100, "%"
```

V. DETAILS OF EXPERIMENTS & ANALYSIS

In my initial experiments, is to trying to see how each classifier performs. Later on tried with different setting, then decide the best setting for sentiment classifier.

A. *Training Data Source*

It is expected that training dataset as a major factor of classifier prediction accuracy. I implemented two different training data set, and it is obviously the training data set source improve significantly (close to 20 %) for specific topic sentiment analysis.

B. *Training Set Data Size*

Training Dataset	Movie Review Accuracy	
	Naive Bayes	*SVM*
Movie Review Datasets v1.0 (1/10 of dataset)	69.6%	70.4%
Sentiment140 Corpus	50.4%	52.4%

As training dataset is crucial for sentiment analysis, the data size is expected to as much as possible to improve the accuracy. However in real application most cases we do not have enough dataset to optimize the result. To wisely choose reasonable size dataset is critical for sentiment analysis. Sometimes too many training data results in over trained to

classifier, the accuracy rate worse. In this experiment, I found 2/10 of dataset size performs the best efficiency prediction accuracy result.

Training Dataset	Movie Review Accuracy	
	Naive Bayes	*SVM*
Movie Review Datasets v1.0 (1/10 of dataset)	69.6%	70.4%
Movie Review Datasets v1.0 (2/10 of dataset)	64%	74.8%
Movie Review Datasets v1.0 (3/10 of dataset)	53.6%	72%
Movie Review Datasets v1.0 (4/10 of dataset)	56.8%	77.6%
Movie Review Datasets v1.0 (5/10 of dataset)	56%	76.8%

C. *Stemming and Stop words*

As mentioned in previous paragraph, cleaning data can remove irrelevant information which leads to better desirable accuracy. In some cases remove stop words and implement stemming algorithms increase model computing overhead, but doesn't result in better result. In this cases for smaller training dataset, these natural language processing techniques does not perform significantly accuracy improvement. To perform more efficiency, later experiments will ignore stop words and stemming algorithm techniques for sentiment analysis.

Training Dataset	Movie Review Accuracy	
	Naive Bayes	*SVM*
Movie Review Datasets v1.0 (2/10 of dataset) without Stemming without remove stop words	64%	74.8%
Movie Review Datasets v1.0 (2/10 of dataset) without Stemming removed stop words	63.2%	69.2%
Movie Review Datasets v1.0 (2/10 of dataset) with Stemming without remove stop words	64.4%	70.4%
Movie Review Datasets v1.0 (2/10 of dataset) with Stemming removed stop words	64.4%	72.8%

D. *Final Decision for Sentiment Classier*

Based on above experiments, I decides to use 2/10 data size of movie review datasets v1.0, without stemming without remove stop words as movie review sentiment classifier to perform sentiment analysis for public tweets.

VI. Classifier Testing with Twitter Data

Twitter is an online social networking platform service users post and read short messages (up to 140 character) called "tweets." Before I implement the classifier above perform sentiment analysis on public tweets, there are several Twitter training data set also available. These twitter training data is generic but not specific for movie review use. Although the movie review training dataset is not consistent format as twitter data such as message length and language usage, the topic specific training data is assumed performed outstanding accuracy. Future work is needed to address the difference.

A. *IMDB Review Top 10 Ranking*

The most challenge tasks for this part is to extract relevant movie review tweets from Twitter REST APIs. Some of the data extracted doesn't accurate enough as movie reviews. For the Godfather as example, some tweets are regarding to movies plot rather than movie review. That leads to different sentiment analysis results. It is assumed the top ranking IMDB movies are older movies which with less users posts their opinion for social media platform, so the quality of tweets is not relevant enough for sentiment classifier I used.

IMDB Rating	Sentiment Analysis
The Godfather (1972)	Negative
The Shawshank Redemption (1994)	Negative
Schindler's List (1993)	Positive
Raging Bull (1980)	Negative
Casablanca (1942)	Negative
Citizen Kane (1941)	Negative
Gone with the Wind (1939)	Negative
The Wizard of Oz (1939)	Positive
One Flew Over the Cuckoo's Nest (1975)	Negative
Lawrence of Arabia (1962)	Negative

B. *All Time Box Office Top 10*

The sentiment analysis result is not accurate as expected neither. Again it's related to the quality of the public tweets. For example the movie "Frozen", the tweets actually extracts weather related tweets. When I use each IMDB movie review posts as testing data to test sentiment classifier, the result is positive as expected.

All Time Box Office Top 10	Sentiment Analysis
Avatar	Negative
Titanic	Negative
Star Wars: The Force Awakens	Negative
Jurassic World	Negative
Marvel's The Avengers	Negative
Furious 7	Negative
Avengers: Age of Ultron	Positive
Harry Potter and the Deathly Hallows Part 2	Negative
Frozen	Negative
Iron Man 3	Negative

C. *Recent Release Movies (Dec 09, 2016)*

Compared to previous two experiments, recent released movies perform better predictions accuracy. The sentiment analysis result could be one of the movie rating reference and box office sales predictions.

Recent Release Movies	Sentiment Analysis
La La Land (2016)	Positive
Office Christmas Party (2016)	Positive
Befikre (2016)	Negative
Miss Sloane (2016)	Positive
Harry Benson: Shoot First (2016)	Negative
Moana (2016)	Positive
Fantastic Beasts and Where to Find Them (2016)	Negative
Arrival (2016)	Positive
Allied (2016)	Positive
Doctor Strange (2016)	Negative

VII. CONCLUSION

In this project I have implemented and investigated several machine leaning and natural language processing techniques to perform sentiment analysis on movie reviews. I used two different training data, and tried different setting to conduct a best efficiency sentiment classifier for public tweets. The result of sentiment analysis can be improved but still can be used for variety of applications such as predict box office sales of recent released movies.

VIII. FUTURE WORK

After the detailed experiments and analysis, the sentiment analysis is still not accuracy enough compared to public results. The accuracy of Tweets on movie review also requires special data query and preprocessing techniques. There are many popular machine learning algorithms and open source library such as deep learning and Google tensorflow library. It is worth to invest effort to perform deeper sentiment analysis on both robust code development and machine learning algorithm research.

REFERENCES

1. Pang, B., Lee, L., & Vaithyanathan, S. (2002, July). Thumbs up?: sentiment classification using machine learning techniques. In Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10 (pp. 79-86). Association for Computational Linguistics.
2. https://en.wikipedia.org/wiki/Sentiment_analysis
3. Potts, Christopher. 2011. On the negativity of negation. In Nan Li and David Lutz, eds., Proceedings of Semantics and Linguistic Theory 20,636-659.
4. Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. (2011). Learning Word Vectors for Sentiment Analysis. The 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011).
5. http://www.laurentluce.com/posts/twitter-sentiment-analysis-using-python-and-nltk/
6. http://snowball.tartarus.org/
7. http://www.enggjournals.com/ijet/docs/IJET15-07-06-027.pdf
8. https://pdfs.semanticscholar.org/c521/80a8fe1acc99b4bf3cf3e11d3c8a38e2c7ff.pdf
9. http://www.dlib.si/stream/URN:NBN:SI:DOC-UXY6UBCY/46d0c222-03ce-4995-8399-07913fa72442/PDF
10. http://www.cs.tau.ac.il/~kfirbar/mlproject/project-ml.pdf
11. http://www.datumbox.com/machine-learning-api/
12. http://textminingonline.com/fasttext-for-fast-sentiment-analysis
13. http://www.wildml.com/2015/12/implementing-a-cnn-for-text-classification-in-tensorflow/
14. http://domkaukinen.com/sentiment-analysis-with-tensorflow/
15. http://andybromberg.com/sentiment-analysis-python/
16. http://streamhacker.com/2010/05/17/text-classification-sentiment-analysis-precision-recall/
17. https://www.cs.cornell.edu/people/pabo/movie-review-data/
18. https://en.wikipedia.org/wiki/Twitter
19. https://dev.twitter.com/rest/public
20. http://www.imdb.com/list/ls055592025/
21. http://www.boxofficemojo.com/alltime/world/
22. http://www.imdb.com/movies-in-theaters/