Twitter sentiment blablabla

Yun Wu **Qiren Chen** **Xiaofan Lu**
University of Texas at Austin
Computer Science Department
{yun, qiren, xiaofan}@cs.utexas.edu

Abstract

This document contains the instructions for preparing a camera-ready manuscript for the proceedings of ACL-2014. The document itself conforms to its own specifications, and is therefore an example of what your manuscript should look like. These instructions should be used for both papers submitted for review and for final versions of accepted papers. Authors are asked to conform to all the directions reported in this document.

1 Introduction

Microblogging websites (such as Twitter, Weibo) have gained popularity in recent years. People can easily post real time, short message to express their opinions. Sentiment of microblog is of particular interest because such information is valuable to both consumer and manufactures.

Most of existing works are based on bag of words classifiers. People propose and evaluate different features to improve the performance of bag-of-words model. Such classifiers can work well in longer documents by relying on a few words with strong sentiment such as ”great” or ”awesome”. However, bag-of-words models have difficulties in handling in negation and comparisons, which involves the structure of sentence. To further improve the performance, new compositional models which capture the structure of sentences are needed.

Socher et al., recently recently proposed a new model called the Recursive Neural Tensor Network (RNTN) to capture the compositional effects with higher accuracy. Recursive Neural Tensor Networks take as input phrases of any length. They represent a phrase through word vectors and a parse tree and then compute vectors for higher nodes in the tree using the same tensor-based composition function. (Socher et al., 2013). They compared their model with standard recursive neural network (RNN), maxtrix-vector RNNs, Navie Bayes and SVM. Their experiment shows that RNTN has the best performance in predicating fine-grained sentiment for all nodes. They also claim that RNTN can accurately captures the sentiment change and scope of negation.

However, the data set used in the above paper is bunch of single sentences extracted from well formatted movie reviews, which is quite different from twitter message. One significant feature of tweet is noisy. It contains ungrammatical sentences, typos, urls, hash tag, target etc. In this paper, we explore the possibility of applying the RNTN model on twitter message. We collected twitter message from different sources and labeled some of them. We preprocessed the twitter corpus to better present the structure of the sentences and deleted some information which is not related the to the structure of the sentence such as urls.

Our experiment indicates bla bla bla.

2 Problem Definition and Algorithm

2.1 Task Definition

In this paper, we address the problem of sentiment analysis of Twitter message. To be more precise, given a message, we want to classify whether the message is of positive or negative (binary decision), or neutral sentiment (ternary decision). For messages conveying both a positive and negative sentiment, whichever is the stronger sentiment should be chosen. For the following two example message, we are expected to return positive on the first one and negative on the second.

> If u haven’t seen #Rio2 yet-GO! You need to meet Gabi! Great singer. Cute. Absolutely hysterical! @KChenoweth `pic.twitter.com/kkVBUKjqE3`

> The rio 2 has one of the worst soundtracks evvvvaaa. I’m at Alamo @Drafthouse Cinema. @marissanicole11 `http://4sq.com/1kKF8qE`

This task is interesting because sentiment of Twitter message can be used as a barometer for public mood and opinion in diverse areas such as entertainment, politics and economics. For example, Diakopoulos and Shamma(2010) use Twitter messages to provide information on the temporal dynamic of sentiment in reaction to the debate video between Barack Obama and John McCain. There is also a report on ”Berkshire Hathaway Stock Rises When Anne Hathaway Makes Headlines”[1], which indicates that sentiment toward public figure may have potential influence over stock market.

Working with these informal text genres presents challenges for natural language processing beyond those typically encountered when working with more traditional text genres, such as newswire data. Tweets and texts are short: a sentence or a headline rather than a document. The language used is very informal, with creative spelling and punctuation, misspellings, slang, new words, URLs, and genre-specific terminology and abbreviations, such as, RT for re-tweet and #hashtags, which are a type of tagging for Twitter messages. How to handle such challenges so as to automatically mine and understand the opinions and sentiments that people are communicating has only very recently been the subject of research

2.2 Algorithm Definition

Describe in reasonable detail the algorithm you are using to address this problem. A psuedocode description of the algorithm you are using is frequently useful. Trace through a concrete example, showing how your algorithm processes this example. The example should be complex enough to illustrate all of the important aspects of the problem but simple enough to be easily understood. If possible, an intuitively meaningful example is better than one with meaningless symbols.

2.2.1 RNTN

2.2.2 SVM

2.2.3 Pre-processing

- Twitter related features
- Slangs
- Emoticons
- Spelling correction
- Word Cluster

[1] `http://newsfeed.time.com/2011/03/06/star-power-success-berkshire-hathaway-stock-rises-when-anne-hathaway-makes-headlines/`

3 Experiment

3.1 Corpus

We conducted our experiment on three types of corpus.

- The first corpus is the Stanford sentiment treebank released by Socher et. al. (2013). It is based on the dataset introduced by Pang and Lee (2005) and consists of 11,855 single sentences extracted from movie reviews. It was parsed with the Stanford parser (Klein and Mannning, 2003) and includes a total of 215,154 unique phrases from those parse trees, each annotated by 3 human judges. It is not twitter message, but would give us a comparison of RNTN and SVM on well formatted English. We refer to this corpus by *Sentiment Treebank* in the reset of paper.
- The second corpus is movie reviews on Twitter, which can be divided into two categories.
  - The first categoriy is single tweet movie review taken from two specialized review accounts (@FilmReviewIn140, @MovieTwoosh). We have 364 tweets in this corpus. Such reviews are mostly well formatted, usually consist of several sentences. The author rated each movie with A to F grades. And if a movie receives a grade no worth than B, we label the review as positive, otherwise, it is negative. We refer to this corpus by *moiveA* in the reset of paper.
  - The second category of movie reviews are collected by searching two currently popular movies names (Rio2 & Captain American2). Such tweets are published by the generally public and they have all the noisy feature of tweet. We manually labeled this corpus. We refer to this corpus by *moiveB* in the reset of paper.
- The third corpus is general tweet message. It is taken from SemEval-2013: Sentiment Analysis in Twitter Task B[2]. Each of the tweet messages has been manually labeled as positive, negative, or neutral. Out of all the 5,750 messages, 2,042 are positive, 855 are negative and 2853 are neutral. We refer to this corpus by *SemEval* in the reset of paper.

3.2 Single Sentence Sentiment

We firstly evaluate both models using *Sentiment Treebank*, which contains single sentence movie reviews extracted from `http://www.rottentomatoes.com/`. We used the same training/testing splits as in the original paper by Socher et. al. (2013).

Model	Accuracy		
	positive	negative	overall
RNTN	80.83	87.91	84.27
SVM1			
SVM2			

Table 1: Binary decision

We also evaluated the performance of using the emotional label as a feature to train the SVM classifier. ..

3.3 Multiple Sentences Sentiment

We then evaluates how different models works on deciding the sentiment of the whole twitter message. This won't affect bag-of-word method much because now we only need a larger bag. However, RNTN relies on the structural of single sentence so we need to combine the sentiment from multiple sentences within a single tweet. Here, we use the model trained on *Sentiment Treebank* and tested on *movieA* corpus.

As for the RNTN model, we evaluated two way to combine the sentiment of the whole tweet (multiple sentences). The first way is to make hard (binary) decision on single sentence (either positive or negative), then use majority vote to decide the sentiment of the whole sentence. Soft information (probability) is only used to break a tie. The second way is fully relying on soft (probability) information. For each sentence, we generate a 5 element vector for the probability of the having the corresponding sentiment (very negative, negative, neutral, positive, very positive). We add the vector for all the sentences

[2] `http://www.cs.york.ac.uk/semeval-2013/task2/index.php?id=data`

together and make final decision based on the combined vector.

Model	Accuracy(%)		
	positive	negative	overall
RNTN$_{hard}$	70.08	81.54	74.18
RNTN$_{soft}$	78.21	80.0	78.85
SVM1			
SVM2			

Table 2: Binary decision

From the result, we can see that hard decision combining has much worse performance than soft decision combining in positive sentiment but slightly better on negative sentiment. I guess this is because the RNTN model tend to label a sentence as negative. Soft combining outperform hard combining by more than 4% in the overall result. This is reasonable because more information is available in soft decision combining. We use soft mode in the following experiments.

3.4 Effect of Preprocessing

We also evaluates how much preprocessing contributed to our final performance. In this experiment, we still use the model trained on *Sentiment Treebank* but tested on *movieB* corpus, which contains noisy common twitter message collected by searching movie names. We evaluated both models with and without preprocessing the original corpus.

Model	Overall Accuracy(%)	
	with pre-processing	w/o pre-processing
RNTN		
SVM1		
SVM2		

Table 3: Effect of preprocessing

Comparing both table, we can see that preprocessing indeed helped in improving the performance.

3.5 General topic Tweet

We conducted three sets of experiment over the general topic twitter corpus *SemEval*.

- Exp 1: Training on 90% of *SemEval* and testing on the rest 10%.

Model	Accuracy(%)		
	positive	negative	overall
RNTN			
SVM1			
SVM2			

Table 4: Experiment 1

- Exp 2: Training on *Sentiment Treebank* and testing on *SemEval*.

Model	Accuracy(%)		
	positive	negative	overall
RNTN	69.83	70.17	69.93
SVM1			
SVM2			

Table 5: Experiment 2

- Exp 3: Training on *Sentiment Treebank*, label the training set of *SemEval*, retrain the model with both *Sentiment Treebank* and the testset of *SemEval*, and test the new model on the test set of *SemEval*.

Model	Accuracy(%)		
	positive	negative	overall
RNTN			
SVM1			
SVM2			

Table 6: Experiment 3

4 Related Work

5 Conclusion

Better parser needed.

Acknowledgments

We would like to thanks Dr. Mooney and our TA Zihao Zhou for their guidance and help on this course project.

References

[Aho and Ullman1972] Alfred V. Aho and Jeffrey D. Ullman. 1972. *The Theory of Parsing, Translation and Compiling*, volume 1. Prentice-Hall, Englewood Cliffs, NJ.

[American Psychological Association1983] American Psychological Association. 1983. *Publications Manual*. American Psychological Association, Washington, DC.

[Association for Computing Machinery1983] Association for Computing Machinery. 1983. *Computing Reviews*, 24(11):503–512.

[Chandra et al.1981] Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. 1981. Alternation. *Journal of the Association for Computing Machinery*, 28(1):114–133.

[Gusfield1997] Dan Gusfield. 1997. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, Cambridge, UK.