Visual Tweets Sentimental Analysis

Lilong Jiang (jiang.573)

April 24, 2014

1 Introduction

In this project, I build a visual online tweets sentimental analysis tool, which allows users to do the sentimental analysis for a certain topic. I also implement two classifiers and compare both classifiers on accuracy.

2 Dataset

We get our dataset from the Sentiment140[1]. The training dataset in this dataset is collected by Twitter API. The tweets whose content contain the emoticons :) and :(are collected. The test dataset in this dataset is manually annotated.

3 Preprocessing

There are several challenges for the tweets processing: firstly, the tweets are very short, limited as 140 characters. Secondly, the tweets are very sloppy and it contains a lot of misspellings, slangs, abbreviations, emoticons, urls, etc. So the preprocessing step is very important for the later classification. We preprocess all the tweets as follows:
(1) Remove usernames and urls.
(2) I use an emoticon dictionary, in which each emoticon is labeled as emotion(positive or negative). The emoticons in the tweets are replaced by happy or sad.
(3) Replace all sequences of repeated characters by three characters, for example, convert goooood to goood.
(4) Expand abbreviations with a dictionary[2]. For example, *lol* is converted into laughing out loud.
(5) NLTK[3] is used to tokenize the tweet, remove the stop words, punctuations, and do the lemmatization.
(6) PyEnchant[4] is used to check whether the token is a correct English word.

[1] http://www.sentiment140.com/
[2] http://www.noslang.com/
[3] http://www.nltk.org/
[4] http://pythonhosted.org/pyenchant/

Only correct English words are retained.
(7) Replace all the negations (e.g. not, no, never, cannot, etc.) by *never*.

4 Classifiers

Two classifiers are implemented in this project. One is the naive Bayes classifier and the other is the SVM.

4.1 Naive Bayes Classifier

The Naive Bayes classifier is a simple probalistic classifier which is based on Bayes theorem. Each tweet will be classified into the class with the highest posterior probability. In order to avoid the underflow, I use the log likelihood.

$$C_{map} = argmax_{c \in C}(logP(c) + \sum_{1 \leq k \leq n_d} logP(t_k|c))$$

The conditional probability of a particular word given a class as the relative frequency of term t in documents belongs to class c in the multinomial naive bayes model while in the binarized multinomial naive bayes model, each term has the same frequency which is 1:

$$P(t|c) = \frac{T_{ct}+1}{\sum_{t \in V}(T_{ct}+1)} = \frac{T_{ct}+1}{\sum_{t \in V}(T_{ct})+B}$$

where B is the number of documents.
I use add-one smoothing to address the problem that a particular word doesn't appear in a particular class.

4.2 Feature Selection

For the naive bayes classifier, I implement two feature selection methods. One is the frequency-based feature selection. The other is based on the mutual information.

4.2.1 Frequency-based Feature Selection

All the adverb and adjective tokens are retained since it is more related to the polarity and for the verb and noun tokens, only tokens whose frequency is greater than 4 are retained.

4.2.2 Mutual Information

Mutual Information is used to measure how much a term t contributes to the class c.

$$I(U;C) = \frac{N_{11}}{N} \log_2 \frac{NN_{11}}{N_{1.}N_{.1}} + \frac{N_{01}}{N} \log_2 \frac{NN_{01}}{N_{0.}N_{.1}} + \frac{N_{10}}{N} \log_2 \frac{NN_{10}}{N_{1.}N_{.0}} + \frac{N_{00}}{N} \log_2 \frac{NN_{00}}{N_{0.}N_{.0}}$$

where N_{10} is the number of documents that contain t and are not in class c. N_{11} is the number of documents that contain t and are in class c. N_{01} is the number of documents that don't contain t and are in class c. N_{00} is the number of documents that don't contain t and are not in class c. $N_{1.}$ is the number of documents that contain t. $N_{0.}$ is the number of documents that don't contain t. $N_{.0}$ is the number of documents which are not in class c. $N_{.1}$ is the number of documents in class c. The top k feature with high I(U; C) value will be selected. Add-1 smoothing is used for each count to avoid the divided by 0.

4.3 SVM Classifier

Libsvm[5] is a library which implemented SVM algorithm. Libsvm is used to classify the tweets. I use C-SVM and set the parameter of the cost to 4. In the SVM classifier, only the unigram feature is used.

4.4 Experiment

I compare these two classifier on the performance and accuracy. 359 tweets are used to test the classifier. In the training dataset, half are positve tweets and half are negative tweets. If the data size is 1000, it means the data set contains 1000 positive tweets and 1000 negative tweets.

4.4.1 Preprocessing

There are a lot processing in this project. In this experiment, three processing steps are checked to see whether they are helpful to increase the accuracy. Remove pyenchant means the incorrect words will be left. Remove Expanding Abbr means the abbrivations are not expanded. Remove Replacing Emoticon means the emoticons are not replaced with the relevant words. The result is shown in Figure 1. As we can see, when the data size is large, some processing steps can't increase the accuracy or even hurt the accuracy. However, when the data size is small, it can help to increase the accuracy. The reason behind this is that when the data size is large, the noisy words have little effect on the accuracy since it can be reduced in the feature selection step.

4.4.2 Accuracy

I compare the accuracy of three different classifiers on different data size, which is shown in Figure 2. As we can see, with the increase of data size, the accuracy is increased. And in most cases, the binarized multinomial naive bayes model performs the best and the multinomial naive bayes model presents more variability. This makes sense, since in the sentiment analysis task, where it does not really matter how many times someone mentions the word but rather only the fact that he does.

[5] http://www.csie.ntu.edu.tw/ cjlin/libsvm/

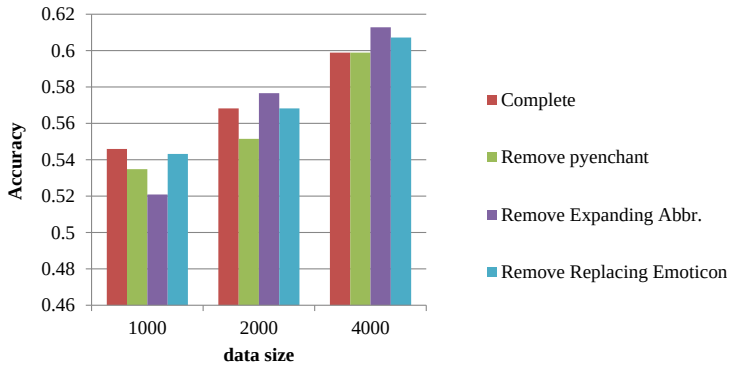

Figure 1: Effect of preprocessing on accuracy

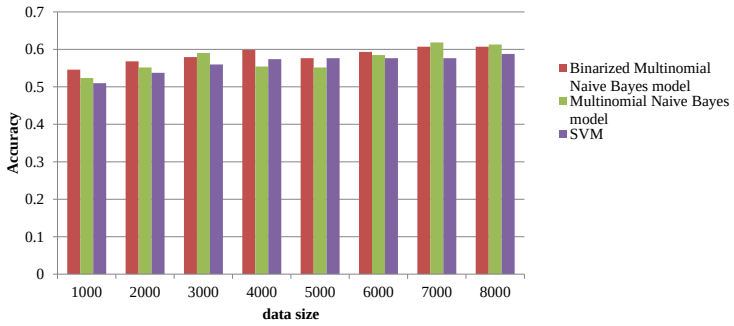

Figure 2: Accuracy with the data size between different classifiers

5 Online Sentimental Analysis

5.1 Architecture

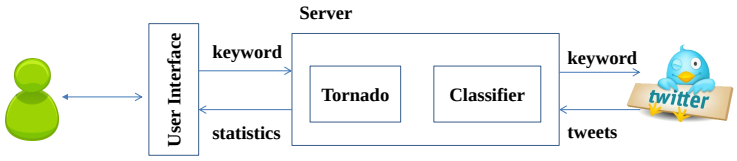

Figure 3: Architecture

In the front end, D3[6] is used to visualize the final result and Tornado[7] works as a web server. When users submit a topic, the server gets 500 tweets from twitter through twitter api and analyzes each tweet and returns the statistics

[6] http://d3js.org/
[7] http://www.tornadoweb.org/

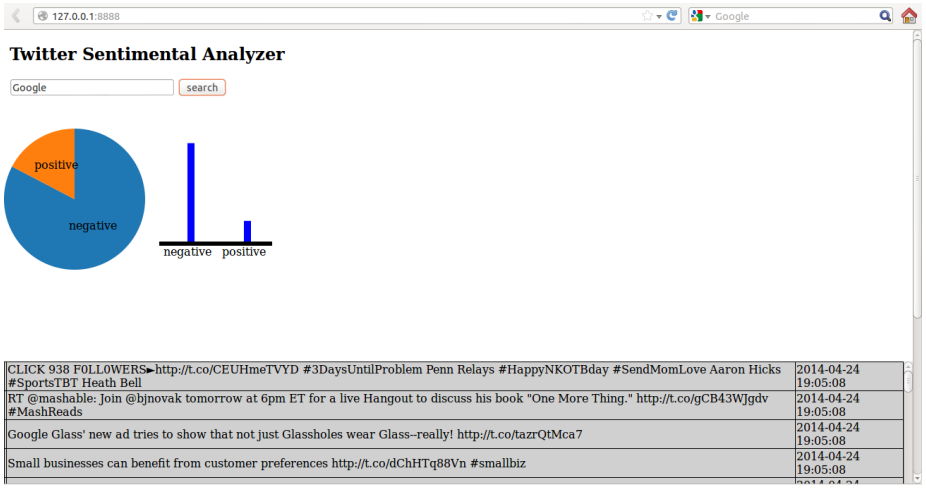

Figure 4: User Interface

information and tweets to the client. The user interface of this system is shown in Figure 4.

5.2 Data Collection

TwitterSearch[8] is a python-based interface which implements the Twitter Search API. I use it to collect the relevant tweets.

5.3 Online Analysis

We analyzes each tweet using the naive bayes classifier and show the final statistics information in the pie chart and bar chart. The tweets will be shown in the table, where the negative tweet will shown in light gray while the positive tweet will be shown in light blue.

6 Future Work

I don't try changing the parameters of SVM to see how the accuracy will be affected. Also the mutual information selection method is needed to compare with the feature-based selection method.

References

[1] Alec Go, Richa Bhayani, Lei Huang. *Twitter Sentiment Classification using Distant Supervision*. Stanford, Technical Report.

[8] `http://twittersearch.readthedocs.org/en/latest/`

[2] Agarwal Apoorv, Xie Boyi, Vovsha Ilia, Rambow Owen, Passonneau Rebecca. *Sentiment analysis of twitter data*. Association for Computational Linguistics, Proceedings of the Workshop on Languages in Social Media.

[3] Pang Bo, Lee Lillian, Vaithyanathan Shivakumar. *Thumbs up?: sentiment classification using machine learning techniques*. Association for Computational Linguistics, Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10.

[4] Qi Han, Junfei Guo, Hinrich Schuetze. *CodeX: Combining an SVM Classifier and Character N-gram Language Models for Sentiment Analysis on Twitter Text*. Joint Conference on Lexical and Computational Semantics.

[5] Christopher D. Manning, Prabhakar Raghavan and Hinrich Schtze. *Introduction to Information Retrieval*. Cambridge University Press.