Twitter Sentiment Classification using Machine Learning Techniques

Zhichao Cao
Department of Computer Science
Drexel University
Philadelphia, PA 19104
zc77@drexel.edu

Abstract

Finding out what other people are thinking is a one of the most important parts in information gathering and retrieval. With the increasing level of available online resources of opinions and reviews in blogs, there is a potential opportunity of digging out and understanding the ideas behind these opinions and reviews. Sentiment acts as a key role to determine the attitude of a speaker or a writer with respect to some specific topics. The attitude can be his or her judgment, evaluation or emotional state of the author while writing.

In this paper, I introduce a practical approach of automatically classifying sentiment of the Twitter messages, and the overall evaluation on results produced by different machine learning algorithms (Naive Bayes, Maximum Entropy, and Support Vector Machine) used in the program. I consider the problem of classifying not by the topic of the messages, but by the overall sentiment determining whether a tweet is positive or negative. Training the classifiers and making them able to determine the sentiment of Twitter messages is meaningful, it helps both customers and companies to research the market and products. The training data consists of twitter messages with different topics, messages are easily obtained through automated methods by calling API functions that are provided by Twitter.

1 Introduction

Nowadays, the information based Internet has very high availability of online documents. Twitter, one of the most famous online social networking service, enables users to send and read short 140-character messages called "tweets". Registered users can read and post tweets from a variety of devices simultaneously, it is a very powerful communication base. The tweets contains opinions associated with various kinds of topics. The paper proposes a practical method to automatically extract sentiment from tweets, and classifies its sentiment in whether positive or negative. This method is helpful for the customers to research the products or services before they make a purchase, and also useful for companies to research public feedback of released products. This method makes them possible to deal with a great number of information and have a result. Moreover, sentiment classification would also be helpful in business intelligence and recommend system [11], as well as message filtering [12].

Twitter has some unique attributes which make sentiment extraction, analysis and classification difficult and different from other information carriers [1]:

1.Text Length, the maximum length of a tweet is only 140 characters, which is very different

from large text such as game review.

2.Topic Diversity, the topics of tweet cover a lot of domains, the diversity of topics makes training data set have high demand on total number.

3.Language Model, language model of tweets has high freedom, there is high frequency of misspelling and slang which makes training process difficult. Moreover, tweet has multiple languages not limited to English.

Sentiment analysis and classification is the field of study that analyzes people's opinions, sentiments, evaluations, appraisals, attitudes, and emotions towards entities such as products, services, organizations, individuals, issues, events, topics, and their attributes. It is using NLP (Natural Language Processing), statistics, or machine learning methods to extract, identify, or otherwise characterize the sentiment content of a text unit.[2]

In this paper, I introduce a practical approach of automatically extracting sentiment from the Twitter messages and classifying them into either positive and negative step by step, and then examine the effectiveness of applying machine learning techniques to the sentiment classification. I also state short evaluation on result of all machine learning techniques, and my thought for improvement of the program and result in the future. Thanks to the Twitter APIs, it is much easier to collect tweets rather than collecting them manually. I download a public data set posted by Stanford University.

2 Related Work

The program work is intended to implement the work of paper by Go, R. Bhayani, and L. Huang posted in 2009 [5]. There are many related work on sentiment classification of text-based document, Cecilia Ovesdotter Alm, Dan Roth and Richard Sproat introduce conception and idea of emotion prediction on text-based document [3]; Stephanie D. Husby, and Denilson Barbosa used the similar way to classify the topic of blog posts [7]; B. J. Jansen, M. Zhang, K. Sobel, and A. Chowdury have also analyzed the brand impact of microblogging [8].

Using machine learning techniques for sentiment extraction and classification is a practical way. Tony Mullen and Nigel Collier introduces an approach to sentiment analysis which uses support vector machines (SVM) to bring together diverse sources of potentially pertinent information [9].

3 Approach

My approach is using three different kinds of machine learning classifiers, which are Naive Bayes, Maximum Entropy (MaxEnt) and Support Vector Machine (SVM), and my feature selection algorithm with the help of Natural Language Toolkit (NLTK).

3.1 Machine Learning Classifiers

The machine learning classifiers that are used in my program are Naive Bayes, Maximum Entropy (MaxEnt) and Support Vector Machine (SVM).

3.1.1 Naive Bayes

One of my sentiment classifiers is based on Naive Bayes algorithm. Naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong independence assumptions between the features. And it works well on text categorization [15].

First the algorithm uses the Bayes rule to express P(label|features) in terms of P(label) and P(features|label) to find the probability for a label:

P(label|features) = (P(label) * P(features|label)) / P(features).

Second, the algorithm makes the 'naive' assumption that all features are independent:

P(label|features) = (P(label) * P(f1|label) * ... * P(fn|label)) / SUM[l](P(l) * P(f1|l) * ... * P(fn|l)).

Finally, the algorithm calculates the denominator for each label, and normalizes them and sums them to one:

P(label|features) = (P(label) * P(f1|label) * ... * P(fn|label)) / (SUM[l](P(l) * P(f1|l) * ... * P(fn|l))).

3.12 Maximum Entropy

One is my sentiment classifiers is based on Maximum Entropy algorithm. Maxi-mum entropy is a probability distribution estimation technique widely used for a variety of natural language tasks, such as language modeling, part-of-speech tagging, and text segmentation [10]. The algorithm considers all of the probability distributions that are empirically consistent with the training data; and chooses the distribution with the highest entropy. I use 10 iterations for total iterative scaling algorithm.

3.1.1 Support Vector Machine

Support Vector Machines (SVM) is supervised learning models with associated learning algorithm that analyzes data and recognizes patterns, used for classification and regression analysis [16]. For SVM in program, I use linear kernel classifier and 0s and 1s to indicate what label (positive or negative) the data from training data set belongs to.

3.2 Natural Language Toolkit (NLTK)

The Natural Language Toolkit (NLTK) is a suite of libraries and programs for symbolic and statistical natural language processing (NLP) for the Python programming language. NLTK's code base of 100,000 lines of Python code includes support for corpus access, tokenizing, stemming, tagging, chunking, parsing, clustering, classification, language modeling, semantic interpretation, unification, and much else besides. [14]

3.3 Feature Reduction

In order to increase the performance of feature selection process and decrease the affect of useless words which do not have much information on sentiment, such as "and", "or" and so forth; And I also exclude or fix the noisy words which affect the accuracy of classifiers such as meaningless symbols, duplicate characters in a word. I set up a filtering rule for pre-processing both training data and testing data.

4 Process and Results

I mainly use NLTK of python for the whole framework of project. The purpose of the implementation is to be able to automatically classify a tweet as a positive or negative tweet sentiment. In order to achieve that, I used a public data set which has been already labeled for training and a set of manually labeled data for testing, and make careful feature selection process to improve performance of the features extracted from both training data and testing data. Finally I calculate and compare result accuracy of three machine learning classifiers. The whole process of project is shown in Figure 1.

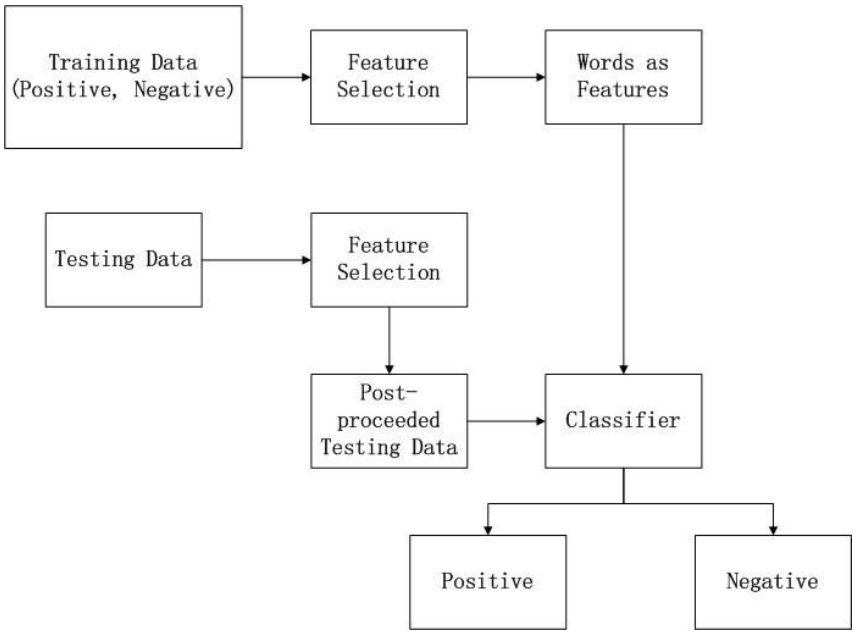

Figure 1: Overall Work-flow of Program.

4.1 Processing and Selection

The tweets are pre-processed before applying feature selection, the noisy words and useless features are excluded in advance, the rule is as follows:

1.Strip punctuation out of the text, such excellent!!! is replaced with excellent;

1.Discard words whose length is less than 3;

2.Strip out @username and URL links;

3.Replace #word with word, such as #Microsoft is replaced with Microsoft;

4.Replace 2 or more repetitions of character with the character itself, and delete additional white spaces, such as gooooooooooood is replaced with good.

After processing, I put the processed words from text into my feature set together with their label at the same time. For example, the tweet "@Angel it is so lucky I found a new wallet. " will be ([luck, found, new, wallet], 'positive'). After that, I extract the list of word features from the tweets, and finally I get a total list of word features together with their occurrence and apply the features to my classifiers. I do the same steps for the my testing data.

4.2 Classification

Classification is the task of choosing the correct class label for a given input. In my project, each input is considered in isolation from all other inputs, and the set of labels is defined in advance. After the previous processing, I instantiate my classifiers, train them with my training data that has been processed and classify test tweets. After applying my classifiers, I can check my most informative features in my program as displayed in Figure 2. For example, the word feature sick is the most informative because about 94% of its occurrences are in negative sentiment tweets.

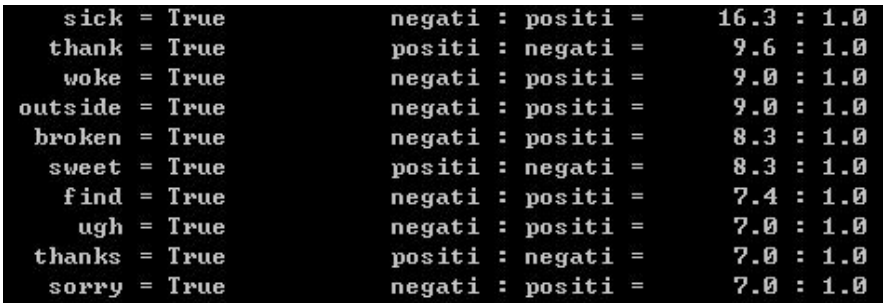

Figure 2: List of Informative Features

4.3 Evaluation

My training data set has 1,600,000 tweets from the time period between April 6, 2009 to June 25, 2009. 800,000 of them are labeled as positive ones, and the other 800,000 are labeled as negative ones. My testing data set has 360 tweets which also has been labeled as half of them are positive and other half are negative, since I want to check the accuracy of my classifiers. While doing experiments, I did not take all training data set into my feature extraction and training process due to its huge size makes program takes super long time. Instead, I randomly took thousands of tweets from my training data set, and made sure the ratio of positive and negative tweets is 1:1; however I took all of my testing data. After applying 5 groups which have tweets with different size and content from my total training data set, I got the result accuracy as displayed in Table 1.

	Group 1 4000 Train 360 Test	Group 2 2000 Train 360 Test	Group 3 1000 Train 360 Test	Group 4 5000 Train 360 Test	Group 5 4000 Train 360 Test
Naive Bayes	76.67%	73.33%	70.13%	78.89%	77.50%
MaxEnt	74.44%	72.83%	69.33%	77.78%	75.56%
SVM	71.1%	69.39%	68.06%	72.33%	70.44

Table 1: Five Groups of Experiment Data

The training data of Group 5 is different from one of Group 1 although both of them have same quantity. From the table of 5 groups, the overall trend is that the performance of Naive Bayes is higher than Maximum Entropy, and the Maximum Entropy has better performance than SVM. However the overall accuracy reaches what I expected although lower than the one of A. Go, R. Bhayani, and L. Huang [5]. My algorithm is able to classify neutral sentiment although it is displayed in my experiment, because my training data and testing data only consist of positive and negative labels.

5 Conclusion and Future Work

Machine learning techniques perform well for classifying sentiment in tweets. I believe the accuracy could be improved. I ran into difficulties while I was taking training and evaluating on the data set due to its size. It is impossible for me to train whole data set which has 1,600,000 tweets, and even when I took more than ten thousands of tweets randomly from data set, it took me couple of hours on my own PC machine. This issue prevents me from finishing several experiments that I was hoping to run. This is also the reason why my training data displayed in Table 1 only has couple of thousands, and why I believe the results could be improved.

While there also exists some feasible way for me to improve accuracy further as well as the performance of training process:

1.Improve and extend my filtering rule, increase the purity of informative word features.

2.Use some algorithm of Natural Language Processing, such as Name Entity Recognizer which is able to locate and classify words from text into pre-defined categories.

3.Try to precisely quantify the benefit of using the tags of subsequent words as features.

4.Try to train and evaluate other languages than English with different sentiment recognition processes and syntactic constructions.

References

[1] Milstein, S., Chowdhury, A., Hochmuth, G., Lorica, B., and Magoulas, R. Twitter and the micro-messaging revolution: Communication, connections, and immediacy - 140 characters at a time. O'Reilly Media, 2008.

[2] Bing Liu. Sentiment Analysis and Opinion Mining, Morgan & Claypool Publishers, May 2012.

[3] Cecilia Ovesdotter Alm, Dan Roth, and Richard Sproat. Emotions from text: machine learning for text-based emotion prediction. In Proceeding of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP), 2005.

[4] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? Sentiment Classification Using Machine Learning Techniques. In Proceeding of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 79-86, 2002.

[5] A. Go, R. Bhayani, and L. Huang. 2009. Twitter sentiment classification using distant supervision. Processing, pages 1-6.

[6] http://en.wikipedia.org/wiki/Twitter. Link to Twitter of Wikipedia.

[7] Stephanie D. Husby, and Denilson Barbosa. Topic Classification of Blog Posts Using Distant Supervision. Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, pages 28 - 36, Avignon, France, April 23 - 27 2012.

[8] B. J. Jansen, M. Zhang, K. Sobel, and A. Chowdury. Micro-blogging as online word of mouth branding. In CHI EA '09: Proceedings of the 27th international conference extended abstracts on Human factors in computing systems, pages 3859{3864, New York, NY, USA, 2009. ACM.

[9] Tony Mullen, and Nigel Collier. 2004. Sentiment analysis using support vector machines with diverse information sources. In Proceedings of EMNLP, 412 - 418.

[10] K. Nigam, J. Lafferty, and A. Mccallum. Using maximum entropy for text classification. InIJCAI-99 Workshop on Machine Learning for Information Filtering, pages 61 - 67, 1999.

[11] Junichi Tatemura. 2000. Virtual reviewers for collaborative exploration of movie reviews. In Proc. of the 5th International Conference on Intelligent User Interfaces, pages 272–275.

[12] Ellen Spertus. 1997. Smokey: Automatic recognition of hostile messages. InProc. of Innovative Applications of Artificial Intelligence (IAAI), pages 1058–1065.

[13] B. Pang and L. Lee. Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval, 2(1-2):1-135, 2008.

[14] Bird, Steven; Klein, Ewan; Loper, Edward; Baldridge, Jason (2008). "Multidisciplinary instruction with the Natural Language Toolkit". Proceedings of the Third Workshop on Issues in Teaching Computational Linguistics, ACL.

[15] C. D. Manning and H. Schutze.Foundations of statistical natural language processing. MIT Press, 1999.

[16] N. Cristianini and J. Shawe-Taylor. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press, March 2000.