KUNLPLab:Sentiment Analysis on Twitter Data

Beakal Gizachew Assefa
Koc Unversity
bassefa13@ku.edu.tr

Abstract

This paper presents the system submitted by KUNLPLab for SemEval-2014 Task9 - Subtask B: Message Polarity on Twitter data. Lexicon features and bag-of-words features are mainly used to represent the datasets. We trained a logistic regression classifier and got an accuracy of 6% increase from the baseline feature representation. The effect of pre-processing on the classifier's accuracy is also discussed in this work.

1 Introduction

Microblogging sites has become a common way of reflecting peoples' opinion. Unlike the regular blogs, the size of a message on a microblogging site is relatively small. The need to automatically detect and summarize the sentiment of messages from users on a given topic or product has gained the interest of researchers.

The sentiment of a message can be negative, positive, or neutral. In the broader sense, automatically detecting the polarity of a message would help business firms easily detect customers' feedback on their product or services. Which in turn helps them improve their decision making by providing information of user preferences, product trend, and user categories.(Chew and Eysenbach, 2010; Salethe and Khandelwal,2011). Sentiment analysis is also used in other domains.(Mandel et al.,2012).

Twitter is one of the mostly widely used microblogging web site with over 200 million users send over 400 million tweets daily(September 2013). A peculiar characteristic of a Twitter data are as follow: emoticons are widely used, the maximum length of a tweet is 140 character, some words are abbreviated, or some are elongated by repeating letters of a word multiple times.

The organizers of the SemEval-2014 has provided a corpus of tweets and posted a task to automatically detect their respective sentiments.

Sub task B of Task 9: Sentiment Analysis on Twitter is describe as follows

Task B - Message Polarity Classification

"*Given a message, classify whether the message is of positive, negative, or neutral sentiment. For messages conveying both a positive and negative sentiment, whichever is the stronger sentiment should be chosen.*"

This paper describes the system submitted by KUNLBLab for participation in SemEval-2014 Task 9 subtask B. Models were trained using the LIBLINEAR classification library (Fan et al., 2008). An accuracy of 66.11% is attained by the classifier by testing on the development set.

The remaining of the document is organized as follows: Section 2 presents a brief literature review on sentiment analysis on Twitter data. Section 3 discusses the system developed to solve the above task, characteristics of the dataset, prepressing on the dataset, and various feature representation. Section 4 illustrates the evaluation results. Section 5 presents conclusion and remarks.

2 Related Work

Sentiment analysis has been studied in Natural Language Processing. Different approaches have been implemented to automatically detect sentiment on texts (Pang et al., 2002; Pang and Lee, 2004; Wiebe and Riloff, 2005; Glance et al., 2005; Wilson et al., 2005).

There is also an active research on Sentiment analysis on Twitter data. (Go et al., 2009, Bermingham and Smeaton, 2010, and Pak and

Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pages 391–394,
Dublin, Ireland, August 23-24, 2014.

Paroubek 2010) consider tweets with good emoticons as positive examples and tweets with bad emoticons as negative examples for the training data, and built a classifier using unigrams and bigrams as features.

Barbosa and Feng (2010) classified the subjectivity of tweets based on traditional features with the inclusion of some witter specific clues such as retweets, hashtags, links, uppercase words, emoticons, and exclamation and question marks.

(Agarwal et al. 2011) introduced a POS-specific prior polarity features and used a tree kernel to obviate the need for tedious feature engineering.

3 System Description

3.1 Dataset

The organizer of SemEval-2014 have provided training and development sets. Table 1 bellow illustrates the characteristics of the dataset.

	Positive	Negative	Neutral
Train	3045	1,209	4004
Dev	575	340	739

Table 1. Dataset characteristics

3.2 Pre-processing

We employed two major pre-processing in the datasets. Converting terms to their correct representation, and stemming.

Mostly, in Twitter, words are not written in their correct/full form. For instance, love, looooove, looove convey the same meaning as the word love alone regardless of the extent of the emphasis intended to describe. Reducing this various representations of the same term to common word helps in better matching them even if they are written in different way. This is more problematic if our features are based on term matching and hence increase the number of unknown terms.

The second pre-processing we employed is stemming the terms in the dataset. In most cases, morphological variants of words have similar semantic interpretations and can be considered as equivalent. The advantage of stemming is twofold. Primarily it reduces the number of OOVs (Out Of Vocabulary) terms. The second one is feature reduction.

3.3 Features

There are two main categories of features used in the development of this system. Bag-of-Words and sentiment lexicon features.

Bag-of-Words features takes a given input text and extracts the raw words as features independent of one another. One issue in using this feature is how to represent negations. In the texts "I *like the movie*. ", and "*I do not like the movie*.", the sentiment of the words in the two texts is opposite since the two statements are negations of one another. One way of representing the negated word is by appending the tag **_NOT** (Chen (2001) and Pang et al. (2002). The **_NOT** tag suffixes all words between the negation word and the first punctuation mark after the negation word. In the above example the second text is transformed to " *I do like_**NOT** the_**NOT** movie **_NOT***". In representation of the negations, we employ the above approach. Lee Becker et al. (2013) directly integrated the polarized word representation in their system. One disadvantage of this representation is the number of features doubles in worst case.

Sentiment lexicons are words, which have association with positive or negative sentiments. Unlike the Bag-Of-Words, instead of taking the raw word as a feature, every word has a score, which is a measure of how much positive or negative sentiment the lexicon has. In this work we use the NRC Hashtag Sentiment Lexicon, and Sentiment140 Lexicon (Mohammad 2013). Both list of lexicons are used in the SemEval 2013 by NRC-Canada team.

The ***NRCHashtag Sentiment Lexicon*** is based on the common practice that users use the # symbol to emphasis on a topic or a word. The hashtag lexicon was created from a collection of tweets that had a positive or a negative word hashtag such as #good, #excellent, #bad, and #terrible (Mohammad 2012). It was created from 775,310 tweets posted between April and December 2012 using a list of 78 positive and negative word hashtags. They have provided unigram, bigram, and trigram datasest. In this work however, we used the unigram features which contains 54,129 terms.

The ***Sentiment140*** is also a list of words with associations to positive an negative sentiments. It has the same format as the NRC Hashtag Sentiment Lexicon. However, it was created from the sentiment140 corpus of 1.6 million tweets, and emoticons were used as positive and negative labels (instead of hashtagged words).

In order to investigate the effect of the features listed above, we have used various combination of them. Table 2 shows 12 kinds of features used for the system we have developed.

The converted versions of the features are the ones where the enlongated words are shortened to

their normal form and terms with less than 5 occurances in the training set are ignored.

Code	Features
F1	RawBag-Of-Word
F2	Bag-Of-WordStemmed
F3	ConvertedStemedBag-Of-Word
F4	Hashtag
F5	Sentiment140
F6	CombinedLexicons
F7	ConvertedHashtag
F8	ConvertedSentiment140
F9	ConvertedNegatedHashtag
F10	ConvertedNegatedSentiment140
F11	ConvertedStemmeLexicon
F12	AllCombined

Table 2. Code of features and their names

The description of the features is as follow, F1 is a raw Bag-Of-Word features in which terms with more than five frequency are taken as features. F2 takes the stem of the words whereas F3 applies both stemmig and shortening of elongeted words to the corpus then takes Bag-Of-Word features of the converted corpus.

F4 and F5 are sentiment lexcon features hashtag. F6 is a combined Sentiment140, and Hashtag features. F7 and F8 are applications of the sentiment lexicons after applying shortening and steming. Negative message representation is included in features F9 and F10. F11 is the combination of a preprocessed corpus by applicaiton of stemming and short represenation of elnogated terms, negative message representation, and extracting a combined sentiword140 and hash tag features.

Feature F12 is the combination of all the features. If a term after being preprocessed is found in one of the lexicon features, the lexicon polarity measure is taken as feature value.Otherwise; we resort to the Bag-Of-Word feature.

3.4 The classifier

For this task, we have used L2 regularized logistic regression and used the LIBLINEAR implementation (Rong-En Fan et al.).To estimate the hyper parameters, we applied a 10 fold cross validation on the training set. Liblinear implementation of a L2 regularized logistic regression takes a single cost C parameter. The value of the cost C parameter decides the weight between the L1 regularization term and L2 regularization term. If the value of C is less than one, it means the more weight it given to the L1 regularization term. On the other hand C values more than one gives more weight to the L2 regularizing term. The cost parameter C=1 gives the best result on the cross validation test. The same value is used to train our model.

4 Evaluation Results

As described in Table 2 of section 3.3, the major features used in this work are bag-of-word and sentiment lexicon features. In addition to the feature representation, pre-processing has been done on the datasets.

F1 is a baseline feature (raw Bag-Of-Word), with a total accuracy of 60.16. Simply converting the elongated terms to their normal form and applying stemming on the corpus increase the accuracy from 60.16 to 64.92 (**4.76%**).

	Positive	Negative	Neutral	Total
F1	61.71	52.48	60.55	60.16
F2`	61.71	51.43	61.18	60.36
F3	67.64	62.86	63.64	64.92
F4	66.67	52.94	60.10	61.65
F5	67.91	54.72	61.00	62.54
F6	64.86	55.24	61.47	61.94
F7	67.72	60.42	63.07	63.51
F8	70.29	58.93	63.02	64.17
F9	70.27	56.12	62.28	63.36
F10	71.73	59.29	62.86	64.65
F11	67.25	62.89	63.14	64.52
F12	**71.12**	**61.4**	**64.13**	**66.11**

Table 3. Results of the evaluation on the development set

F6 (the combined lexicon feature- sentiword140 and hashtag) yields an accuracy of 61.94. Applying conversion, negative representation and stemming raises the accuracy to 64.52 (**F11**)

Testset	MacroF1
LiveJournal2014	63.77
SMS2013	55.89
Twitter2013	58.12
Twitter2014	61.72
Twitter2014Sarcasm	44.60

Table 4. Evaluation result on test set

The accuracy of identifying negative sentiment is the least in all features. This shows that we need a better representation of negated messages.

A test dataset was also provided by the organizer of semEval-2014. Table 4 show the accuracy of the KUNPLab classifier.

Our model has performed poorly on the Twitter2014Sarcasm test set (44.60%). The performance of our classifier on LiveJournal2014 is similar to the development set test performance.

5 Conclusion

The performance of a classifier depends on feature representation, hyperparameter optimization and regularization. In this work, we mainly used bag-of-word features and sentiment lexicon features. We trained a L2 regularized logistic regression model. Two major features are used to represent the datasets; Bag-of-Word features and Lexical features. It has been shown that stemming the terms increases accuracy of the classifier in either case. The accuracy of the classifier on development set and training set is reported and has shown an increase of 6% in accuracy form the baseline with 95% confidence interval..The evaluation of our system on SemEval-2014 test data is also shown with an F measure of 44.60 to 63.77%.

6 Acknowledgement

I would like to acknowledge Ass.Prof. Dr. Deniz YURET for his advice, guidance, encouragement and inspiration to participate in SemEval-2014. I also like to thank Mohammad Khuram SALEEM, and Mohamad IRFAN for proof reading this document.

Reference

Cynthia Chew and Gunther Eysenbach. 2010. Pandemics in the Age of Twitter: Content Analysis of Tweets during the 2009 H1N1 Outbreak. PLoS ONE, 5(11):e14118+, November.

Marcel Salath´e and Shashank Khandelwal. 2011. Assessing vaccination sentiments with online social media: Implications for infectious disease dynamics and control. PLoS Computational Biology, 7(10).

Benjamin Mandel, Aron Culotta, John Boulahanis, Danielle Stark, Bonnie Lewis, and Jeremy Rodrigue. 2012. A demographic analysis of online sentiment during hurricane irene. In Proceedngs of the Second Workshop on Language in Social Media, LSM'12, pages 27–36, Stroudsburg, PA, USA. Association for Computational Linguistics.

Sanjiv Das and Mike Chen. 2001. Yahoo! for amazon: extracting market sentiment from stock message boards. In Proceedings of the 8th Asia Pacific Finance Association Annual Conference.

Lee Becker, George Erhart, David Skiba and Valentine Matula. 2013. AVAYA: Sentiment Analysis on Twitter with Self-Training and Polarity Lexicon Expansion. Seventh International Workshop on Semantic Evaluation (SemEval 2013)

Saif Mohammad. 2012. Emotional Tweets. In Proceedings of the First Joint Conference on Lexical and Computational Semantics (*SEM), pages 246–255, Montr´eal, Canada. Association for Computational Linguistics.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment Classification Using Machine Learning Techniques. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)

Mohammad, Saif and Kiritchenko, Svetlana and Zhu, Xiaodan 2013. NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013).

Go, A., Bhayani, R., Huang, L.: Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford (2009)

Barbosa, L., Feng, J.: Robust sentiment detection on Twitter from biased and noisy data. In: Proceedings of COLING. pp. 36–44 (2010)

Agarwal, A., Xie, B., Vovsha, I., Rambow, O., Passonneau, R.: Sentiment analysis of Twitter data. In: Proc. ACL 2011 Workshop on Languages in Social Media. pp. 30–38 (2011)

Adam Bermingham and Alan Smeaton. 2010. Classifying sentiment in microblogs: is brevity an advantage is brevity an advantage? ACM, pages 1833–1836.

Alexander Pak and Patrick Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining. Proceedings of LREC.

Glance, N., M. Hurst, K. Nigam, M. Siegler, R. Stock ton, and T. Tomokiyo. 2005. Deriving marketing intelligence from online discussion. In Proceedings of the eleventh ACM SIGKDD, pages 419–428. ACM.

Wiebe, J. and E. Riloff. 2005. Creating subjective and objective sentence classifiers from unannotated texts. Computational Linguistics and Intelligent Text Processing, pages 486–497.

R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A Library for Large Linear Classification, Journal of Machine Learning Research 9(2008), 1871-1874