ENHANCED SENTIMENT ANALYSIS OF TWITTER DATA

Internship Report

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF TECHNOLOGY
NITK, Surathkal

By

Priyanka
(16CS13F)

Under the Guidance of
Dr. N Balakrishnan

DEPARTMENT OF SUPERCOMPUTER EDUCATION AND RESEARCH CENTRE

INDIAN INSTITUTE OF SCIENCE

BENGALURU – 560012

July 2017

D E C L A R A T I O N

I certify that the report on “ENHANCED SENTIMENT ANALYSIS OF TWITTER DATA” which is being submitted as record of my internship is a bonafide report of the work carried out by me. The material contained in this report has not been submitted to any University or Institution for the award of any degree.

Priyanka

C E R T I F I C A T E

This is to certify that the report titled "ENHANCED SENTIMENT ANALYSIS OF TWITTER DATA" submitted to the Department of S.E.R.C., Indian Institute of Science, by Priyanka is accepted as a record of the work carried out by her as part of Summer Internship.

N. Balakrishnan
(GUIDE)

ABSTRACT

In this report, I have presented the work done by me on Sentiment Analysis of Twitter data. Sentiment is nothing but polarity of a tweet, word or phrase. It can be positive, negative or neutral. Tweets have boundation that one tweet can be of 140 words. Micro-blogging sites gives opportunity to various organizations to analysis the sentiment of a particular word, product, elections and much more. Machine Learning is a field which is going to rule the world for so many years. So more the enhancement in this field is, more will be benefit to the society. I have performed sentiment analysis on Twitter data extracting various features. I have trained my system to perform sentiment analysis of each tweet.I have achieved a score of 73.0373 % through my sentiment analysis, which is best till date.

MOTIVATION

Many researchers have come up with brilliant ideas to do sentiment analysis. One of the student with N. Balakrishnan from SERC, IISC came up with sensor fusion technique where she combined results from various machine learning algorithms to enhance the accuracy of sentiment analysis. Also many researchers have tried to perform sentiment analysis and came up with various algorithms. No doubt many of their ideas were brilliant and gave sufficiently good performance.
They performed sentiment analysis with the available machine learning algorithms like SVM, neural networks, decision tree but the features upon which these algorithms were run are different.I will be doing the twitter sentiment analysis as many researchers tried, but with improvement. Also as this is one of the hot topic in current research, I would like to contribute a little to it.

PROBLEM STATEMENT

Sentiment analysis helps to process the opinion of public and hence help in making predictions for future. It is defined as finding emotional polarity of the text. These days micro-blogging websites help public to share their opinion freely and this becomes a huge source of data which can be processed. Sentiment Analysis try to identify the view point of the person who has text-ed (T), on topic (O). It identifies if the text is positive, negative or neutral towards topic, O. It not only identifies the sentiment of sentence or a tweet but can also identify for a document as a whole or for a word.
So I will be extracting different features and will make use of them to full extent. That is, Evaluating which features gives better result with a particular data. Also I will be using SVM (Support Vector Machine) as machine learning algorithm and will be using labeled SemEval2013 dataset.

LITERATURE SURVEY

1. Twitter Sentiment Classification using Distant Supervision, 2009
Alec Go, Richa Bhayani, Lei Huang from Stanford University

Methodology : Twitter messages were classified as positive or negative. They considered emoticons as noisy data, as same sentence can have both sad and happy emoticons and they were affecting the result of SVM and MaxEnt.

Data used for feature extraction : Collected data from twitter API themselves, they collected tweets with emoticons and removed emoticons from them. They took the first 800,000 tweets with positive emoticons, and 800,000 tweets with negative emoticons, for a total of 1,600,000 training tweets. For testing, a set of 177 negative tweets and 182 positive tweets manually marked. Twittratr's list - 174 positive words and 185 negative words.

Features reduced : mentions by USERNAME, links by URL and removed repeated letters.
Features extracted : unigrams, bigrams, unigrams and bigrams, and unigrams with part of speech tags.

Classifier : Naive Bayes, Maximum Entropy, and SVM.

Performance: (accuracy)

Features/MLA	Naive Bayes	MaxEnt	SVM
unigram	81.3%	80.5%	82.2%
unigram & bigram	82.7%	82.7%	81.6%

Shortcomings: They did not include neutral class, does not include emoticons for feature extraction, sentiment analysis on only English language. They found general purpose POS tagger and bigram alone feature not much useful to them.

2. Twitter as a Corpus for Sentiment Analysis and Opinion Mining, Alexander Pak, Patrick Paroubek, 2010
Dataset : They collected a corpus of 300000 text posts from Twitter evenly split automatically between three sets of texts: positive, negative and neutral emotions.

Filtering : removed URL's, mentions, special words, stopwords and emoticons. Salience and entropy is calculated for each ngram to discriminate redundant or unimportant ngrams from features.

Features: n-gram :- uni, bi and tri and TreeTagger for POS tagging.

Classifier : multinominal Naive Bayes, SVM and CRF.

Performance : For them Naive Bayes performed well. They calculated accuracy, decision and F-measure. They did not report the result, as they used different metrics of result.

3. NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets, 2013
Saif M. Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu

Methodology : they performed message-level and term-level sentiment analysis. They also made word-sentiment association lexicon, A set of 775,000 tweets(by pooling Twitter API) were used to generate a large word sentiment association lexicon. They analyzed the sentiment analysis by adding features one by one and then checking the performance. Ngram ans lexicon features contributed most to performance gain.

Dataset : SemEval2013 dataset for tweets as well as SMS.

Tokenization and POS tagging is done with the Carnegie Mellon University (CMU) Twitter NLP tool.

Lexicons used : NRC Emotion Lexicon , the MPQA Lexicon and the Bing Liu Lexicon.NRC Hashtag Sentiment Lexicon has entries for 54,129 unigrams and 316,531 bigrams(made by them, and now publicly available), Sentiment 140 lexicon.

Features : word ngrams, character ngrams, all-caps, presence of URL and hashtag POS ,hashtags, Lexicons,punctuations, emoticons, elongated words, clusters, term splitting, position and negations (The list of negation words was adopted from Christopher Potts' sentiment tutorial). Among them sentiment lexicon feature found to be most effective one.

Classifier : SVM with linear kernel C=0.005 and 10-fold cross validation is done.

Performance : F-score of 69.02 in the message-level task and 88.93 in the term-level task. A post-competition bug-fix in the bigram features resulted in a small improvement: F-score of 89.10 on the tweets set and 88.34 on the SMS set.

4. TeamX: A Sentiment Analyzer with Enhanced Lexicon Mapping and Weighting Scheme for Unbalanced Data, 2014, Yashuhide et al.

Pre-processing : Unicode normalization (NFKC), all caps to lowercase, links by "URL's" keyword, spelling corrector (Jazzy), POS tagger (Stanford POS tagger and CMU ARK POS tagger), word sense disambiguator (UKB), negation detector.

Enhancements done: spelling corrector, two POS taggers, sense disambiguator and weighing scheme.

Lexicons used : AFINN-111, Bing Liu's, general enquirer, MPQA subjectivity lexicon, NRC hashtag sentiment lexicon, Sentiment140 lexicon and SentiWordNet.

Features : word ngrams(1,2,3,4), character ngrams(3,4), lexicons, clusters and word senses.

Classifier : Logistic regression, LIBLINEAR with C=0.03, Wpos=2.4 and Wneg=3.3.

Performance : average F1 72.12 on Twitter2013, average F1 70.96 on twitter2014 and average F1 56.50 on twitter2014sarcasm.

Shortcoming: parameters can be tuned more better to enhance the performance.

5. CMUQ@Qatar: Using Rich Lexical Features for Sentiment Analysis on Twitter, Bin wasi et al.
Task A : sentiment of a phrase witin the tweet,
TaskB : sentiment of the whole tweet.

classifier used : SVM (linear kernel), C=0.18 (Task-A) and C=0.55(Task-B)

dataset : SemEval 2014 and Test data : Twitter2013,Twitter2014, LiveJournal2014,SMS2013 and TwitterSarcasm.

Preprocessing : 1. Tokenization and POS tagging: CMU ARK Tokenizer and POS Tagger
2. NetLingo and WordNetlemmatizer in NLTK
Features :
1. Bag of words (unigram, bigram, trigram)
2. POS features (num_hashtags, num_adverb, num_adjective)
3. Polarity features – polarity at token level (MPQA subjectivity lexicon-num_mapped_positive, sentiment140 + Bing liu's – positive and negative , SentiWordNet lexicon with WordNet)
4. num_capital, has_url, num_emoticon (for Task-B).

Performance : 87.11% in Task-A and 64.52% in Task-B.

Future work : more accurate and larger lexicons.

SYSTEM OVERVIEW

1. Data Pre-processing : It is needed so that features can be extracted more meaningfully. During preprocessing I have removed various unwanted texts in the tweet.

1.1. Tokenization: CMU ARK tokenizer is used to perform tokenization. Though this I was able to analyze each token separately and removed unwanted ones. I did further processing in tokens.

1.2. POS Tagging: CMU ARK POS tagger is used for POS tagging. Each token is marked with a tag. Accuracy of this tagger is 93%. The various tags are shown in table below:

Tag	Description	Examples	%
Nominal, Nominal + Verbal			
N	common noun (NN, NNS)	books someone	13.7
O	pronoun (personal/WH; not possessive; PRP, WP)	it you u meeee	6.8
S	nominal + possessive	books' someone's	0.1
^	proper noun (NNP, NNPS)	lebron usa iPad	6.4
Z	proper noun + possessive	America's	0.2
L	nominal + verbal	he's book'll iono (= *I don't know*)	1.6
M	proper noun + verbal	Mark'll	0.0
Other open-class words			
V	verb incl. copula, auxiliaries (V*, MD)	might gonna ought couldn't is eats	15.1
A	adjective (J*)	good fav lil	5.1
R	adverb (R*, WRB)	2 (i.e., *too*)	4.6
!	interjection (UH)	lol haha FTW yea right	2.6
Other closed-class words			
D	determiner (WDT, DT, WP$, PRP$)	the teh its it's	6.5
P	pre- or postposition, or subordinating conjunction (IN, TO)	while to for 2 (i.e., *to*) 4 (i.e., *for*)	8.7
&	coordinating conjunction (CC)	and n & + BUT	1.7
T	verb particle (RP)	out off Up UP	0.6
X	existential *there*, predeterminers (EX, PDT)	both	0.1
Y	X + verbal	there's all's	0.0
Twitter/online-specific			
#	hashtag (indicates topic/category for tweet)	#acl	1.0
@	at-mention (indicates another user as a recipient of a tweet)	@BarackObama	4.9
~	discourse marker, indications of continuation of a message across multiple tweets	RT and : in retweet construction RT @user : hello	3.4
U	URL or email address	http://bit.ly/xyz	1.6
E	emoticon	:-) :b (: <3 o_O	1.0
Miscellaneous			
$	numeral (CD)	2010 four 9:30	1.5
,	punctuation (#, $, '', (,), ,, ., :, ``)	!!! .... ?!?	11.6
G	other abbreviations, foreign words, possessive endings, symbols, garbage (FW, POS, SYM, LS)	ily (*I love you*) wby (*what about you*) 's ♫ --> awesome...I'm	1.1

1.3. Dependency parsing : TweeboParser is used for establishing dependency between tokens of each tweet. This was helpful for negations. Example, "not good" in this good is positive but due to not in front of it as a whole this phrase becomes negative. Dependency parser helps in this. Its accuracy is 80% on twitter data.

1.4. For ngram: removing mentions, digits, URL's, hashtags, punctuations, emoticons, one and two length words, also made all the words to small before ngram feature extraction. The before mentioned are removed by me to avoid unnecessary formation of unigrams and bigrams.

2. Feature Extraction : After preprocessing the data, features can be extracted efficiently as we have removed unwanted data from the dataset upon which analysis is to be performed. For my system, I have collected the below mentioned features:

2.1. n-grams: In this frequency of each gram is represented in frequency vector. It is been proved by many researchers that this is one of the most useful feature to perform sentiment analysis. Mohanmmad et al. (2013) mentioned in its paper that Bag of Words and Lexical features are highly useful for better sentiment analysis.

2.2. Presence of hashtag, mention, URL : Their presence itself is found important. It is been noticed that their count did not contributed much to the performance, so only their presence is taken as one of the feature.

2.3. Num_of_Adverbs and Num_of_Adjectives : It is easy to notice that mainly adjectives and adverbs show the polarity in a sentence. For example, "Rita is so beautiful" in this sentence beautiful is adjective. So I have taken count of adjectives and adverbs as one of the feature.

2.4. Lexicon : I have used AFFIN-111, Bing liu's lexicons as one of the feature vector. There are various other Lexicons available like MPQA, Sentiment140, NRC hashtag. I am going to add even them in future. Many researcher have proved that lexicon features are highly effective in improving performance.

2.5. Emoticons : They do contribute to confusion but are again highly effective in giving sentiment to a text. Some positive smiles can also contribute to sarcasm. I will be working on this in near future.

3. Feature Selection Evaluation Feature : I used Gain Ratio Attribute feature reduction method using Weka. It is been proved in Kavitha et al.(2012) paper that Gain ratio is one of the best method to do feature reduction. So i used this and manually set the parameter to get the output as only certain count of features. As mentioned below:

n-gram	count
unigram	650
bigram	850
trigram	475

4 . Classifier

I choose SVM as classifier as from my theoretical study on various Machine Learning algorithms, I found this best among the other classifiers. I also noticed this practically while training my data through various machine learning algorithms like Random Forest, Naive Bayes, MaxEnt. I achieved maximum performance through SVM with default setting. I am yet to experiment with various attributed of SVM algorithm.

SVM : Support Vector Machines are the supervised leaning models that are associated with some learning algorithms. These models analyze the data and can be used for classification or regression analysis.

1. Goal of SVM : Find the optimal separating line which maximize the margin of training data.
2. Margin : Given a particular hyperplane, we can find the distance between the hyperplane and the closest data point, if we double it, we will get margin.
3. These closest data points are called the SUPPORT VECTORS.
4. Basically margin is no man's land.
5. There will never be any data point in the margin.

Since I have multi class data and SVM is by default a binary classifier and there is no standard way for dealing with multi-class problems. So the basic idea to apply multi classification to SVM is to decompose the multi-class problem into several two class problem that can be addressed directly using several SVM's. This approach only I have done with SVM.

EXPERIMENTAL APPROACH

With default setting I ran SVM and got a good performance of 73.0373 %.

Time taken to build and train my model is given below:

Time taken to build model: 353.64 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0.56 seconds

Confusion Matrix obtained is:

=== Confusion Matrix ===

CLASS	**positive**	**negative**	**neutral**
positive	792	29	444
negative	50	243	250
neutral	105	39	1449

Output of SVM with Default Settings:

=== Summary ===

Correctly Classified Instances	2484	73.0373 %
Incorrectly Classified Instances	917	26.9627 %
Kappa statistic	0.5404	
Mean absolute error	0.2974	
Root mean squared error	0.3859	
Relative absolute error	72.3918 %	
Root relative squared error	85.0996 %	
Total Number of Instances	3401	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.626	0.073	0.836	0.626	0.716	0.597	0.807	0.687	positive
	0.448	0.024	0.781	0.448	0.569	0.538	0.794	0.516	negative
	0.910	0.384	0.676	0.910	0.776	0.543	0.765	0.659	neutral
Weighted Avg.	0.730	0.211	0.753	0.730	0.721	0.563	0.785	0.647	

CONCLUSION

I successfully performed sentiment analysis and achieved a performance of 73.0373%. I have studies various ways to enhance the feature and also implemented a few of them. Also during this system making I understood various Machine Learning algorithms.

FUTURE WORK

While making the above system I came across various other lexicons, so in future I will be including even them in my feature vector. Also now I just applied default settings of SVM, so in future I will be learning to vary the kernel setting of SVM and various other attributed to enhance my system further.

REFERENCES

[1] Mohammad, S., Kiritchenko S., and Zhu, X. 2013. NRCCanada: Building the state of the art in sentiment analysis of tweets. In Proceedings of the Seventh international workshop on Semantic Evaluation Exercise,SemEval2013, pages 321–327, Atlanta, Georgia, USA.

[2] Nielsen, F. Å. (2011). A new ANEW: Evaluation of a word list for sentiment analysis in microblogs.arXivpreprint arXiv:1103.2903.

[3] Yasuhide, M. (2014). TeamX: A Sentiment Analyzer with Enhanced Lexicon Mapping and Weighting Scheme for Unbalanced Data. In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014).Association for Computational Linguistics.

[4]Kavitha, A. S., Kavitha, R., & Viji Gripsy, J. (2012). Empirical Evaluation of Feature Selection Technique in Educational Data Mining. ARPN Journal of Science and Technology 2 (11).

[5] Kong, L., Schneider, N., Swayamdipta, S., Bhatia, A., Dyer, C., Smith, N. A. (2014). A Dependency Parser for Tweets. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).

[6] Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., ... & Smith, N. A. (2011, June). Partofspeech tagging for twitter: Annotation, features, and experiments. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papersVolume 2 (pp. 4247). Association for Computational Linguistics.

[7] Bin wasi, S., Neyaz, R., Bouamor, H., Mohit, B. (2014). CMUQ@Qatar: Using Rich Lexical Features for Sentiment Analysis on Twitter. In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014). Association for Computational Linguistics

[8] CMUQ@Qatar: Using Rich Lexical Features for Sentiment Analysis on Twitter, Bin wasi et al.