CMUQ@Qatar:Using Rich Lexical Features for Sentiment Analysis on Twitter

Sabih Bin Wasi, Rukhsar Neyaz, Houda Bouamor, Behrang Mohit
Carnegie Mellon University in Qatar
{sabih, rukhsar, hbouamor, behrang}@cmu.edu

Abstract

In this paper, we describe our system for the Sentiment Analysis of Twitter shared task in SemEval 2014. Our system uses an SVM classifier along with rich set of lexical features to detect the sentiment of a phrase within a tweet (Task-A) and also the sentiment of the whole tweet (Task-B). We start from the lexical features that were used in the 2013 shared tasks, we enhance the underlying lexicon and also introduce new features. We focus our feature engineering effort mainly on Task-A. Moreover, we adapt our initial framework and introduce new features for Task-B. Our system reaches weighted score of 87.11% in Task-A and 64.52% in Task-B. This places us in the 4th rank in the Task-A and 15th in the Task-B.

1 Introduction

With more than 500 million tweets sent per day, containing opinions and messages, Twitter[1] has become a gold-mine for organizations to monitor their brand reputation. As more and more users post about products and services they use, Twitter becomes a valuable source of people's opinions and sentiments: what people can think about a product or a service, how positive they can be about it or what would people prefer the product to be like. Such data can be efficiently used for marketing. However, with the increasing amount of tweets posted on a daily basis, it is challenging and expensive to manually analyze them and locate the meaningful ones. There has been a body of recent work to automatically learn the public sentiments from tweets using natural language processing techniques (Pang and Lee, 2008; Jansen et al., 2009; Pak and Paroubek, 2010; Tang et al., 2014). However, the task of sentiment analysis of tweets in their free format is harder than that of any well-structured document. Tweet messages usually contain different kinds of orthographic errors such as the use of special and decorative characters, letter or word duplication, extra punctuation, as well as the use of special abbreviations.

In this paper, we present our machine learning based system for sentiment analysis of Twitter shared task in SemEval 2014. Our system takes as input an arbitrary tweet and assigns it to one of the following classes that best reflects its sentiment: positive, negative or neutral. While positive and negative tweets are subjective, neutral class encompasses not only objective tweets but also subjective tweets that does not contain any "polar" emotion. Our classifier was developed as an undergrad course project but later pursued as a research topic. Our training, development and testing experiments were performed on data sets published in SemEval 2013 (Nakov et al., 2013). Motivated with its performance, we participated in SemEval 2014 Task 9 (Rosenthal et al., 2014). Our approach includes an extensive usage of off-the-shelf resources that have been developed for conducting NLP on social media text. Our original aim was enhancement of the task-A. Moreover, we adapted our framework and introduced new features for task-B and participated in both shared tasks. We reached an F-score of 83.3% in Task-A and an F-score of 65.57% in Task-B. That placed us in the 4th rank in the task-A and 15th rank in the task-B.

Our approach includes an extensive usage of off-the-shelf resources that have been developed for conducting NLP on social media text. That includes the Twitter Tokenizer and also the Twitter POS tagger, several sentiment analysis lexica

[1] http://twitter.com

and finally our own enhanced resources for special handling of Twitter-specific text. Our original aim in introducing and evaluating many of the features was enhancement of the task-A. Moreover, we adapted our framework and introduced new features for task-B and participated in both shared tasks. We reached an F-score of 83.3% in Task-A and an F-score of 65.57% in Task-B. That placed us in the 4th rank in the task-A and 15th rank in the task-B.

2 System Overview

We participate in tasks A and B. We use three-way classification framework in which we design and use a rich feature representation of the Twitter text. In order to process the tweets, we start with a pre-processing step, followed by feature extraction and classifier training.

2.1 Data Pre-processing

Before the tweet is fed to the system, it goes through pre-processing phase that breaks tweet string into words (tokenization), attaches more information to each word (POS tagging), and other treatments.

Tokenization: We use CMU ARK Tokenizer (Owoputi et al., 2013) to tokenize each tweet. This tokenizer is developed to tokenize not only space-separated text but also tokens that need to be analyzed separately.

POS tagging: We use CMU ARK POS Tagger (Owoputi et al., 2013) to assign POS tags to the different tokens. In addition to the grammatical tags, this tagger assigns also twitter-specific tags like @ mentions, hash tags, etc. This information is used later for feature extraction.

Other processing: In order to normalize the different tokens and convert them into a correct English, we find acronyms in the text and add their expanded forms at the end of the list. We decide to keep both the acronym and the new word to ensure that if the token without its expansion was the word the user meant, then we are not losing any information by getting its acronym. We extend the NetLingo[2] top 50 Internet acronym list to add some missing acronyms. In order to reduce inflectional forms of a word to a common base form we use WordNetlemmatizer in NLTK (Bird et al., 2009)[3]. This could be useful for the feature extraction, to get as much matches as possible between the train and test data (e.g., for bag-of-words feature).

Tweet	”This is so awesome @henry:D! #excited”
Bag of Words	”This”:1, ”is”:1, ”so”:1, ”awesome”:1, ”@henry”:1, ”:D”:1, ”!,”:1, #excited”:1
POS features	numHashTags:1, numAdverb:1, numAdjective:1
Polarity features	positiveWords:1, negWords:0, avgScore: -0.113
Task-B specific features	numCapsWords:0, numEmoticons:1, numUrls:0

Table 1: Set of Features demonstrated on a sample tweet for Task-B.

2.2 Feature Extraction

Assigning a sentiment to a single word, phrase or a full tweet message requires a rich set of features. For this, we adopt a forward selection approach (Ladha and Deepa, 2011) to select the features that characterize to the best the different sentiments and help distinguishing them. In this approach, we incrementally add the features one by one and test whether this boosts the development results. We heavily rely on a binary feature representation (Heinly et al., 2012) to ensure the efficiency and robustness of our classifier. The different features used are illustrated in the example given in Table 1.

Bag-of-words feature: indicates whether a given token is present in the phrase.

Morpho-syntactic feature: we use the POS and twitter-specific tags extracted for each token. We count the number of adjectives, adverbs and hashtags present in the focused part of the tweet message (entire tweet or phrase). We tried adding other POS based features (e.g., number of possessive pronouns, etc.), but only the aforementioned tags increased the result figures for both tasks.

Polarity-based features: we use freely available sentiment resources to explicitly define the polarity at a token-level. We define three feature categories, based on the lexicon used:

[2] http://www.netlingo.com/top50/popular-text-terms.php

[3] http://www.nltk.org/api/nltk.stem.html

	Task-A			Task-B		
	Dev	Train	Test	Dev	Train	Test
Positive	57.09 %	62.06%	59.49%	34.76%	37.59%	39.01%
Negative	37.89%	33.01%	35.31%	20.56%	15.06%	17.15%
Neutral	5.02%	4.93%	5.21%	44.68%	47.36%	43.84%
All	1,135	9,451	10,681	1,654	9,684	8,987

Table 2: Class size distribution for all the three sets for both Task-A and Task-B.

- *Subjectivity:* number of words mapped to "positive" from the MPQA Subjectivity lexicon (Wilson et al., 2005).

- *Hybrid Lexicon:* We combine the Sentiment140 lexicon (Mohammad et al., 2013) with the Bing Liu's bag of positive and negative words (Hu and Liu, 2004) to create a dictionary in which each token is assigned a sentiment.

- *Token weight:* we use the SentiWordNet lexicon (Baccianella et al., 2010) to define this feature. SentiWordNet contains positive, negative and objective scores between 0 and 1 for all senses in WordNet. Based on this sense level annotation, we first map each token to its weight in this lexicon and then the sum of all these weights was used as the tweet weight.

Furthermore, in order to take into account the presence of negative words, which modify the polarity of the context within which they are invoked, we reverse the polarity score of adjectives or adverbs that come within 1-2 token distance after a negative word.

Task specific features: In addition to the features described above, we also define some task-specific ones. For example, we indicate the number of capital letters in the phrase as a feature in Task-A. This could help in this task, since we are focusing on short text. For Task-B we indicate instead the number of capital words. This relies on the intuition that polarized tweets would carry more (sometimes all) capital words than the neutral or objective ones. We also added the number of emoticons and number of URL links as features for Task-B. Here, the goal is to segregate fact-containing objective tweets from emotion-containing subjective tweets.

2.3 Classifier

We use a Support Vector Machine (SVM) classifier (Chang and Lin, 2011) to which we provide the rich set of features described in the previous section. We use a linear kernel and tune its parameter C separately for the two tasks. Task-A system was bound tight to the development set with C=0.18 whereas in Task-B the system was given freedom by setting C=0.55. These values were optimized during the development using a brute-force mechanism.

	Task-A	**Task-B**
LiveJournal 2014	83.89	65.63
SMS 2013	88.08	62.95
Twitter 2013	89.85	65.11
Twitter 2014	83.45	65.53
Sarcasm	78.07	40.52
Weighted average	**87.11**	**64.52**

Table 3: F1 measures and final results of the system for Task-A and Task-B for all the data sets including the weighted average of the sets.

3 Experiments and Results

In this section, we explain details of the data and the general settings for the different experiments we conducted. We train and evaluate our classifier for both tasks with the training, development and testing datasets provided for the SemEval 2014 shared task. The size of the three datasets we use as well as their class distributions are illustrated in Table 2 . It is important to note that the total dataset size for training and development set (10,586) is about the same as test set making the learning considerably challenging for correct predictions. **Positive** instances covered more than half of each dataset for Task-A while **Neutral** were the most popular class for Task-B. The class distribution of training set is the same as the test set.

	Task-A	Task-B
all features	87.11	64.52
all-preprocessing	80.79(-6.32)	59.20(-5.32)
all-ARK tokenization	83.69(-3.42)	60.61(-3.91)
all-other treatments	85.06(-2.05)	62.19(-2.33)
only BOW	81.69(-5.42)	57.85(-6.67)
all-bow	82.05(-5.06)	52.04(-12.48)
all-pos	86.92(-0.19)	64.31(-0.21)
all-polarity based features	81.80(-5.31)	57.95(-6.57)
all-SVM tuning	80.82(-6.29)	21.41(-43.11)
all-SVM c=0.01	84.20(-2.91)	59.87(-4.65)
all-SVM c=selected	87.11(0.00)	64.52(0.00)
all-SVM c=1	86.39(-0.72)	62.51(-2.01)

Table 4: F-scores obtained on the test sets with the specific feature removed.

The test dataset is composed of five different sets: *Twitter2013* a set of tweets collected for the SemEval2013 test set, *Twitter2014*, tweets collected for this years version, *LiveJournal2014* consisting of formal tweets, *SMS2013*, a collection of sms messages,*TwitterSarcasm*, a collection of sarcastic tweets. The results of our system are shown in Table 3. The top five rows shows the results by the SemEval scorer for all the data sets used by them. This scorer took the average of F1-score of only positive and negative classes. The last row shows the weighted average score of all the scores for Task A and B from the different data sets.

Our scores for Task-A and Task-B were 83.45 and 65.53 respectively for Twitter 2014.

Our system performed better on Twitter and SMS test sets from 2013. This was reasonable since we tuned our system on these datasets. On the other hand, the system performed worst on sarcasm test set. This drop is extremely evident in Task-B where the results were dropped by 25%. To analyze the effects of each step of our system, we experimented with our system using different configurations. The results are shown in Table 4 and our analysis is described in the following subsections. The results were scored by SemEval 2014 scorer and we took the weighted average of all data sets to accurately reflect the performance of our system.

We show the polarities values assigned to each token of a tweet by our classifier, in Table 5.

Tokens	POS Tags	Sentiments	Polarity
This	O	Neutral	-0.194
Is	V	Neutral	-0.115
So	R	Neutral	-0.253
Awesome	A	Positive	2.351
@Henry	@	-	-
#excited	#	Positive	1.84

Table 5: Polarity assigned using our classifier to each word of a Tweet message.

3.1 Preprocessing Effects

We compared the effects of basic tokenization (based on white space) against the richer ARK Twitter tokenizer. The scores dropped by 3.42% and 3.91% for Task-A and Task-B, respectively. Other preprocessing enhancements like lemmatization and acronym additions also gave our system performance a boost. Again, the effects were more visible for Task-B than for Task-A. Overall, the system performance was boosted by 6.32% for Task-A and 5.32% for Task-B. Considering the overall score for Task-B, this is a significant change.

3.2 Feature Engineering Effects

To analyze the effect of feature extraction process, we ran our system with different kind of features disabled - one at a time. For Task-A, unigram model and polarity based features were equally important. For Task-B, bag of words feature easily outperformed the effects of any other feature. However, polarity based features were second important class of features for our system. These suggest that if more accurate, exhaustive

and social media representative lexicons are made, it would help both tasks significantly. POS based features were not directly influential in our system. However, these tags helped us find better matches in lexicons where words are further identified with their POS tag.

3.3 Classifier Tuning

We also analyzed the significance of SVM tuning to our system. Without setting any parameter to SVMutil library (Chang and Lin, 2011), we noticed a drop of 6.29% to scores of Task-A and a significant drop of 43.11% to scores of Task-B. Since the library use poly kernel by default, the results were drastically worse for Task-B due to large feature set. We also compared the performance with SVM kernel set to C=1. In this restricted setting, the results were slightly lower than the result obtained for our final system.

4 Discussion

During this work, we found that two improvements to our system would have yielded better scores. The first would be lexicons: Since the lexicons like Sentiment140 Lexicon are automatically generated, we found that they contain some noise. As we noticed a drop of that our results were critically dependent on these lexicons, this noise would have resulted in incorrect predictions. Hence, more accurate and larger lexicons are required for better classification, especially for the tweet-level task. Unlike SentiWordNet these lexicons should contain more informal words that are common in social media. Additionally, as we can see our system was not able to confidently predict sarcasm tweets on both expression and message level, special attention is required to analyze the nature of sarcasm on Twitter and build a feature set that can capture the true sentiment of the tweet.

5 Conclusion

We demonstrated our classification system that could predict sentiment of an input tweet. Our system performed more accurately in expression-level prediction than on entire tweet-level prediction. Our system relied heavily on bag-of-words feature and polarity based features which in turn relied on correct part-of-speech tagging and third-party lexicons. With this system, we ranked 4th in SemEval 2014 expression-level prediction task and 15th in tweet-level prediction task.

Acknowledgment

We would like to thank Kemal Oflazer and the shared task organizers for their support throughout this work.

References

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, pages 2200–2204, Valletta, Malta.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media, Inc.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27.

Jared Heinly, Enrique Dunn, and Jan-Michael Frahm. 2012. Comparative Evaluation of Binary Features. In *Proceedings of the 12th European Conference on Computer Vision*, pages 759–773, Firenze, Italy.

Minqing Hu and Bing Liu. 2004. Mining and Summarizing Customer Reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 168–177, Seattle, WA, USA.

Bernard J Jansen, Mimi Zhang, Kate Sobel, and Abdur Chowdury. 2009. Twitter Power: Tweets as Electronic Word of Mouth. *Journal of the American society for information science and technology*, 60(11):2169–2188.

L. Ladha and T. Deepa. 2011. Feature Selection Methods and Algorithms. *International Journal on Computer Science and Engineering (IJCSE)*, 3:1787–1797.

Saif Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 321–327, Atlanta, Georgia, USA.

Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. 2013. SemEval-2013 Task 2: Sentiment Analysis in Twitter. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 312–320, Atlanta, Georgia, USA.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A. Smith. 2013. Improved Part-of-Speech Tagging for Online Conversational Text with Word Clusters. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–390, Atlanta, Georgia.

Alexander Pak and Patrick Paroubek. 2010. Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, pages 1320–1326, Valletta, Malta.

Bo Pang and Lillian Lee. 2008. *Opinion Mining and Sentiment Analysis*, volume 2. Now Publishers Inc.

Sara Rosenthal, Preslav Nakov, Alan Ritter, and Veselin Stoyanov. 2014. SemEval-2014 Task 9: Sentiment Analysis in Twitter. In *Proceedings of the Eighth International Workshop on Semantic Evaluation (SemEval'14)*, Dublin, Ireland.

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 1555–1565, Baltimore, Maryland.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing Contextual Polarity in Phrase-level Sentiment Analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 347–354.