uOttawa: System description for SemEval 2013 Task 2 Sentiment Analysis in Twitter

Hamid Poursepanj, Josh Weissbock, and Diana Inkpen
School of Electrical Engineering and Computer Science
University of Ottawa
Ottawa, K1N6N5, Canada
{hpour099, jweis035, Diana.Inkpen}@uottawa.ca

Abstract

We present two systems developed at the University of Ottawa for the SemEval 2013 Task 2. The first system (for Task A) classifies the polarity / sentiment orientation of one target word in a Twitter message. The second system (for Task B) classifies the polarity of whole Twitter messages. Our two systems are very simple, based on supervised classifiers with bag-of-words feature representation, enriched with information from several sources. We present a few additional results, besides results of the submitted runs.

1 Introduction

The Semeval 2013 Task 2 focused on classifying Twitter messages ("tweets") as expressing a positive opinion, a negative opinion, a neutral opinion, or no opinion (objective). In fact, the neutral and objective were joined in one class for the requirements of the shared task. Task A contained target words whose sense had to be classified in the context, while Task B was to classify each text into one of the three classes: positive, negative, and neutral/objective. The training data that was made available for each task consisted in annotated Twitter message. There were two test sets for each task, one composed of Twitter messages and one of SMS message (even if there was no specific training data for SMS messages). See more details about the datasets in (Wilson et al., 2013).

2 System Description

We used supervised learning classifiers from Weka (Witten and Frank, 2005). Initially we extracted simple bag-of-word features (BOW). For the submitted systems, we also used features calculated based on SentiWordNet information (Baccianella et al., 2010). SentiWordNet contains positivity, negativity, and objectivity scores for each sense of a word. We explain below how this information was used for each task.

As classifiers, we used Support Vector Machines (SVM) (SMO and libSVM from Weka with default values for parameters), because SVM is known to perform well on many tasks, and Multinomial Naive Bayes (MNB), because MNB is known to perform well on text data and it is faster than SVM.

2.1 Task A

Our system for Task A involved two parts: the expansion of our training data and the classification. The expansion was done with information from SentiWordNet. Stop words and words that appeared only once in the training data were filtered out. Then the classification was completed with algorithms from Weka.

As mentioned, the first task was to expand all of the tweets that were provided as training data. This was doing using Python and the Python NLTK library, as well as SentiWordNet. SentiWordNet provides a score of the sentient state for each word (for each sense, in case the word has more than

*Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 380–383, Atlanta, Georgia, June 14-15, 2013. ©2013 Association for Computational Linguistics

one sense). As an example, the word “want” can mean “a state of extreme poverty” with the SentiWordNet score of (Positive: 0 Objective: 0.75 Negative: 0.25). The same word could also mean “a specific feeling of desire” with a score of (Positive: 0.5 Objective: 0.5 Negative: 0). We also used for expansion the definitions and synonyms of each word sense, from WordNet.

The tweets in the training data are labeled with their sentiment type (Positive, Negative, Objective and Neutral). Neutral and Objective are treated the same. The provided training data has the target word marked, and also the sentiment orientation of the word in the context of the tweeter message. These target words were the ones expanded by our method. When the target was a multi-word expression, if the expression was found in WordNet, then the expansion was done directly; if not, each word was expanded in a similar fashion and concatenated to the original tweet. These target words were looked up in SentiWordNet and matched with the definition that had the highest score that also matched their sentiment label in the training data.

Original Tweet	The great Noel Gallagher is about to hit the stage in St. Paul. Plenty of room here so we're 4th row center. Plenty of room. Pretty fired up
Key Words	Great
Sentiment	Positive
Definition	very good; "he did a bully job"; "a neat sports car"; "had a great time at the party"; "you look simply smashing"
Synonyms	Swell, smashing, slap-up, peachy, not_bad, nifty, neat, keen, groovy, dandy, cracking, corking, bully, bang-up
Expanded Tweet	The great Noel Gallagher is about to hit the stage in St. Paul. Plenty of room here so were 4th row center. Plenty of room. Pretty fired up swell smashing slap-up peachy not_bad nifty neat keen groovy dandy cracking corking bully bang-up very good he did a bully job a neat sports car had a great time at the party you look simply smashing

Table 1: Example of tweet expansion for Task A

The target word’s definition and synonyms were then concatenated to the original tweet. No additional changes were made to either the original tweet or the features that were added from SentiWordNet. An example follows in Table 1. The test data (Twitter and SMS) was not expanded, because there are no labels in the test data to be able to choose the sense with corresponding sentiment.

2.2 Task B

For this task, we used the following resources: SentiwordNet (Baccianella et al, 2010), the Polarity Lexicon (Wilson et al., 2005), the General Inquirer (Stone et al., 1966), and the Stanford NLP tools (Toutanova et al., 2003) for preprocessing and feature selection. The preprocessing of Twitter messages is implemented in three steps namely, stop-word removal, stemming, and removal of words with occurrence frequency of one. Several extra features will be used: the number of positive words and negative words identified by three lexical resources mentioned above, the number of emoticons, the number of elongated words, and the number of punctuation tokens (single or repeated exclamation marks, etc.). As for SentiWordNet, for each word a score is calculated that shows the positive or negative weight of that word. No sense disambiguation is done (the first sense is used), but the scores are used for the right part-of-speech (in case a word has more than one possible part-of-speech). Part-of-Speech tagging was done with the Stanford NLP Tools. As for General Inquirer and Polarity Lexicon, we simply used the list positive and negative words from these resources in order to count how many positive and how many negative terms appear in a message.

3 Results

3.1 Task A

For classification, we first trained on our expanded training data using 10-fold cross-validation and using the SVM (libSVM) and Multinomial NaiveBayes classifiers from Weka, using their default settings. The training data was represented as a bag of words (BOW). These classifiers were chosen as they have given us good results in the past for text classification. The classifiers were run with 10-fold cross-validation. See Table 2 for the results. Without expanding the tweets, the accuracy of the SVM classifier was equal to the baseline of classifying everything into the most frequent class, which was “positive“ in the training data. For MNB, the results were lower than the baseline. After expanding the tweets, the accuracy increased to 73% for SVM and to 80.36% for MNB. We concluded that MNB works better for Task A. This is why the submitted runs used the MNB model that was created from the expanded training data. Then we used this to classify the Twitter and SMS test data. The average F-score for the positive and the negative class for our submitted runs can be seen in Table 3, compared to the other systems that participated in the task. We report this measure because it was the official evaluation measure used in the task.

System	SVM	MNB
Baseline	66.32%	66.32%
BOW features	66.32%	33.23%
BOW+ text expansion	73.00%	**80.36%**

Table 2: Accuracy results for task A by 10-fold cross-validation on the training data

System	Tweets	SMS
uOttawa system	0.6020	0.5589
Median system	0.7489	0.7283
Best system	0.8893	0.8837

Table 3: Results for Task A for the submitted runs (Average F-score for positive/negative class)

The precision, recall and F-score on the Twitter and SMS test data for our submitted runs can be seen in Tables 4 and 5, respectively. All our submitted runs were for the “constrained” task; no additional training data was used.

Class	Precision	Recall	F-Score
Positive	0.6934	0.7659	0.7278
Negative	0.5371	0.4276	0.4762
Neutral	0.0585	0.0688	0.0632

Table 4: Results for Tweet test data for Task A, for each class.

Class	Precision	Recall	F-Score
Positive	0.5606	0.5705	0.5655
Negative	0.5998	0.5118	0.5523
Neutral	0.1159	0.2201	0.1518

Table 5: Results for SMS test data for Task A, for each class.

3.2 Task B

First we present results on the training data (10-fold cross-validation), then we present the results for the submitted runs (also without any additional training data).

Table 6 shows the overall accuracy for BOW features for two classifiers, evaluated based on 10-fold cross validation on the training data, for two classifiers: SVM (SMO in Weka) and Multidimensional Naïve Bays (MNB in Weka). The BOW plus SentiWordNet features also include the number of positive and negative words identified from SentiWordNet. The BOW plus extra features representation includes the number of positive and negative words identified from SentiWordNet, General Inquirer, and Polarity Lexicon (six extra features). The last row of the table shows the overall accuracy for BOW features plus all the extra features mentioned in Section 2.2, including information extracted from SentiWordNet, Polarity Lexicon, and General Inquirer. We can see that the SentiWordNet features help, and that when including all the extra features, the results improve even more. We noticed that the features from the Polarity Lexicon contributed the most. When we removed GI, the accuracy did not change much; we believe this is because GI has too small coverage.

System	SVM	MNB
Baseline	48.50%	48.50%
BOW features	58.75%	59.56%
BOW+ SentiWordNet	69.43%	63.30%
BOW+ extra features	**82.42%**	73.09%

Table 6: Accuracy results for task B by 10-fold cross-validation on the training data

The baseline in Table 6 is the accuracy of a trivial classifier that puts everything in the most frequent class, which is neutral/objective for the training data (ZeroR classifier in Weka).

The results of the submitted runs are in Table 7 for the two data sets. The features representation was BOW plus SentiWordNet information. The official evaluation measure is reported (average F-score for the positive and negative class). The detailed results for each class are presented in Tables 8 and 9.

In Table 7, we added an extra row for a new uOttawa system (SVM with BOW plus extra features) that uses the best classifier that we designed (as chosen based on the experiments on the training data, see Table 6). This classifier uses SVM with BOW and all the extra features.

System	Tweets	SMS
uOttawa submitted system	0.4251	0.4051
uOttawa new system	**0.8684**	**0.9140**
Median system	0.5150	0.4523
Best system	0.6902	0.6846

Table 7: Results for Task B for the submitted runs (Average F-score for positive/negative).

Class	Precision	Recall	F-score
Positive	0.6206	0.5089	0.5592
Negative	0.4845	0.2080	0.2910
Neutral	0.5357	0.7402	0.6216

Table 8: Results for each class for task B, for the submitted system (SVM with BOW plus SentiWordNet features) for the Twitter test data.

Class	Precision	Recall	F-score
Positive	0.4822	0.5508	0.5142
Negative	0.5643	0.2005	0.2959
Neutral	0.6932	0.7988	0.7423

Table 9: Results for each class for task B, for the submitted system (SVM with BOW plus SentiWordNet features) for the SMS test data.

4 Conclusions and Future Work

In Task A, we expanded upon the Twitter messages from the training data using their keyword's definition and synonyms from SentiWordNet. We showed that the expansion helped improve the classification performance. In future work, we would like to try an SVM using asymmetric soft-boundaries to try and penalize the classifier for missing items in the neutral class, the class with the least items in the Task A training data.

The overall accuracy of the classifiers for Task B increased a lot when we introduced the extra features discussed in section 2.2. The overall accuracy of SVM increased from 58.75% to 82.42% (as measures by cross-validation on the training data). When applying this classifier on the two test data sets, the results were very surprisingly good (even higher that the best system submitted by the SemEval participants for Task B[1]).

References

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

Philip J. Stone, Dexter C. Dunphy, Marshall S. Smith, and Daniel M. Ogilvie. The General Inquirer: A computer approach to content analysis. MIT Press, 1966.

Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In Proceedings of HLT-NAACL 2003, pp. 252-259, 2003.

Theresa Wilson, Zornitsa Kozareva, Preslav Nakov, Sara Rosenthal, Veselin Stoyanov and Alan Ritter. SemEval-2013 Task 2: Sentiment Analysis in Twitter. In Proceedings of the International Workshop on Semantic Evaluation SemEval '13, Atlanta, Georgia, June 2013.

Theresa Wilson, Janyce Wiebe and Paul Hoffmann. Recognizing contextual polarity in phrase- level sentiment analysis. In Proceedings of HLT/ EMNLP 2005.

Ian H. Witten and Eibe Frank. Data Mining: Practical Machine Learning Tools and Techniques, 2nd edition, Morgan Kaufmann, San Francisco, 2005.

[1] Computed with the provided scoring script.