SWAT-CMW: Classification of Twitter Emotional Polarity using a Multiple-Classifier Decision Schema and Enhanced Emotion Tagging

Riley Collins, Daniel May, Noah Weinthal, and Richard Wicentowski
Swarthmore College
Swarthmore, PA 19081
{rcollin4, dmay1, nweinth1, richardw}@cs.swarthmore.edu

Abstract

In this paper, we describe our approach to Semeval 2015 task 10 subtask B, message level sentiment detection. Our system implements a variety of classifiers and data preparation techniques from previous work. The set of features and classifiers used in the final system produced consistently strong results using cross-validation on the provided training data. Our final system achieved an F-score of 57.60 on the provided test data. The overall best performing system had an F-score of 64.84.

1 Introduction

With the unprecedented growth of social media in the past decade, more individuals than ever before have a means to share their opinions and broadcast their voice. As the number of readily available opinions grows, a challenge of academic and commercial importance emerges. Namely, if the sentiment of social media communications can be reliably determined algorithmically, a deeply informative dataset can be developed. Such data can be used in a variety of applications, from predicting election results to seeing how well a new product is received. However, this task is greatly complicated by inconsistencies in spelling, grammar, lexicon, and other linguistic phenomena found in online communications. The SemEval 2015 Task 10 Subtask B (Rosenthal et al., 2015) challenges participants to determine the sentiment polarity of posts on the social media site Twitter. Specifically, the task is to decide whether the sentiment of a given tweet is positive, negative, or neutral. In this paper we present an approach to this task which synthesizes a number of different preprocessing techniques and classification methods, which we use to classify the tweets.

Our approach was inspired by several approaches to previous iterations of this task. The winning team in 2014, TeamX, used several preprocessors including text normalization, lexical sense mapping, clustering, and word sense disambiguation to train a machine learner to determine emotion (Miura et al., 2014). Ultimately, we hypothesized that a successful approach relies not only on the choice of a good classifier, but also in large part upon the preparation of data for that classifier. This hypothesis led us to place high value on our prepocessing, and as such we focused our energy on implementing strategies that would lead to improvements within existing classifiers, a decision which ultimately led to the creation of our decision schema.

2 System Description

2.1 Preprocessing

Our system makes use of various preprocessing steps in order to reduce the dimensionality of the data set and improve overall performance. These steps included:

- Tokenization using Twokenizer (Gimpel et al., 2011), a tokenizer designed specifically for Tweets.
- Case-folding so that all text is lower-cased.
- All unique URLs were conflated to a single token in both the training and test data.

Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 669–672,
Denver, Colorado, June 4-5, 2015. ©2015 Association for Computational Linguistics

Each of these preprocessing steps improved performance regardless of which features we later extracted and which classifier we tried.

2.2 External Lexicons

As part of feature extraction, our system makes use of two external lexicons. We used a manually created list of definitively positive and negative words (Hu and Liu, 2004) and an automatically generated list of words and their associated sentiment polarities in the Sentiment140 lexicon (Mohammad et al., 2013). The polarities associated with the words in the Sentiment140 lexicon are determined based on how often the word appears in automatically labeled positive or negative Tweets.

Our system searches through each token in the Tweet for matches against the two sentiment lexicons. When a match was found in the Sentiment140 lexicon, a special positive (or negative) feature was added to the feature set with a magnitude correlated to the polarity listed in the lexicon. When a match was found only in the Hu and Liu lexicon, a special positive (or negative) feature was added to the feature set but with a fixed magnitude because this lexicon did not provide strength of the sentiment along with each word.

2.3 Features

Our system finds tokens indicating negation, such as “no”, “never”, and “not” plus any contractions containing “not”. Unlike many other implementations, which prefixes negation words with a single identifying term, our implementation prefixes each negation token with either “NO”, “NEVER”, or “NOT” until the next punctuation mark, similar to (Zhu et al., 2014). This strategy performed better than one which used a single negation prefix.

Features in our system included unigrams and bigrams of tokens in the Tweet (modified as necessary by negation as described above) and the positive and negative features added by finding matches in the external lexicons.

2.4 Classifiers

The system uses an SVM and Naive Bayes classifier from SciKit Learn (Pedregosa et al., 2011), and a simple classifier that counts occurrences of tokens in the Tweet that match words in the sentiment lexicon (Hu and Liu, 2004). Our experience with the SVM was that while it was our best preforming classifier overall, it had a tendency to mislabel both positive and negative Tweets as neutral. Therefore, once the SVM has performed its classification, our system uses a secondary classifier before providing its final sentiment labeling. Figure 1 gives an overview of the classification system.

SVM + Neutralizer The initial classifier involves using the default SVM classifier found in SciKit. This produces a three-way labeling of either positive, negative or neutral. After the initial SVM classification, we use a rule-based classifier to reduce the number of tweets that are incorrectly labeled as negative. This classifier counts the number of positive and negative words in the tweet according to the sentiment lexicon. If the number of positive words is greater than the number of negative words and the tweet was labeled negative, we change the label to neutral; otherwise the label is unchanged.

Naive Bayes We used the default implementation of the Naive Bayes classifier from SciKit.

Sentiment Lexicon We use the sentiment lexicon to count the number of tokens in the tweet that have positive or negative sentiment. If there are more negative words in the tweet than positive words, we label the tweet negative; otherwise, positive.

2.5 The Decision Schema

The SVM classifier had two large sources of error. First, it incorrectly labeled many neutral tweets as positive. Second, it labeled many positive and negative tweets as neutral. In order to address this, we implemented a decision schema to correct for these errors in the SVM, as shown in Figure 1.

To correct for errors where the SVM incorrectly labeled neutral tweets as positive, we used a secondary Naive Bayes classifier. This secondary classifier was trained only on positive and neutral tweets, and provides a final classification as either positive or neutral.

To correct for errors where the SVM incorrectly over-labeled tweets as neutral, we also used a secondary Naive Bayes classifier. However, this classifier was trained on all tweets in a binary fashion, where the tweets were labeled as either neutral or non-neutral. If this Naive Bayes classifier provided

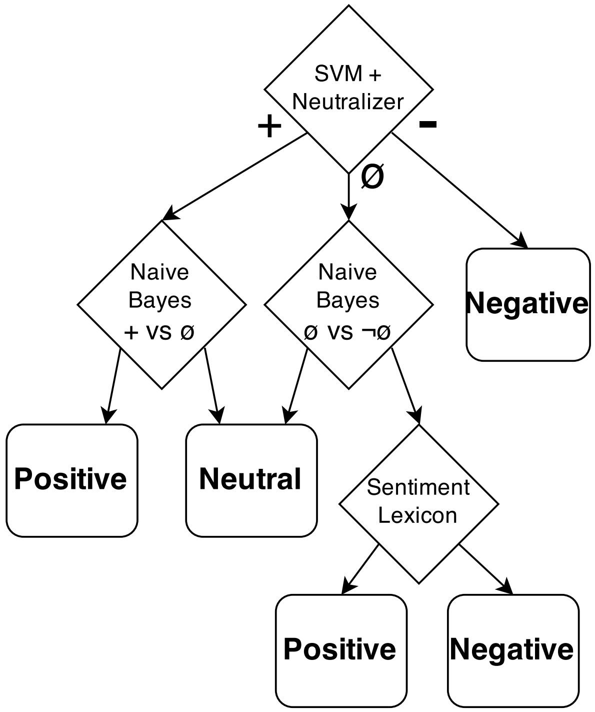

Figure 1: The System's Decision Schema.

a neutral labeling, that became the final label. In the case where a non-neutral label was predicted, we used the sentiment lexicon classifier to provide the final labeling.

3 Results

3.1 Decision Schema Performance

During development, the performance of our Decision Schema was evaluated in two ways. First, we performed a cross-evaluation, where we split the training test into a number of 'chunks', reserved one chunk for testing and trained on the others, then swapped which chunk was reserved and repeated until all 'folds' had been tested and reported an average of the results. Then, we tested against a small development Tweet corpus. Results for each can be seen in Tables 1 and 2, respectively. Against the 2015 test data, we achieved an overall score of 57.60.

3.2 Conclusions

As can be seen from Table 1 and Table 2, our classifier performs quite well in assigning tags to positive and neutral Tweets. Our system tends not to perform as well in our tests at tagging negative Tweets. This potentially implies that we may not be providing enough weight to negative-polarity Tweet features throughout our preprocessing and feature extraction processes, that our decision schema logic unfairly discourages negative tags, or simply that more training data is needed due to the comparatively small number of negative Tweets in the corpus.

Sentiment	Prec	Recall	F1
Negative	51.52	57.48	54.34
Neutral	77.16	71.98	74.78
Positive	69.66	73.51	71.53
Overall Score	62.93		

Table 1: Performance of the system cross-validated on the 2015 training set.

Sentiment	Prec	Recall	F1
Negative	54.77	60.55	57.51
Neutral	72.96	68.06	70.42
Positive	68.26	70.91	69.56
Overall Score	63.53		

Table 2: Performance of the system trained on the 2015 training set and evaluated on the 2015 development set.

4 Future Work

With every new preprocessing and classification system that we added, numerous potential improvements presented themselves. While time constraints prohibited implementing these improvements, we briefly mention them here.

4.1 Preprocessing

We experimented with using case (e.g. HAPPY vs happy) as a feature and expected that all-caps would serve as an indicator of stronger emotional content. In evaluation, this was not the case, but we would like to explore this further.

We would like to incorporate a dependency parser, such as (Kong et al., 2014), which might enable more accurate negation by better revealing where the negating word stops modifying the words in the Tweet. We would also like to include the part-of-speech tagger in Twokenizer (Gimpel et al., 2011) and incorporate word-sense disambiguation, both of which might allow us to better determine emotional polarities for homographs.

4.2 Classification

We would like to experiment with more classifiers. In particular, we would like to investigate SciKit's AdaBoost and Decision Tree modules, both of which promise better performance but are computationally expensive. We would also like to further develop our approach of dividing the task into a series of binary classifications rather than a ternary classification. Additionally, we would like to explore dimensionality-reduction methods like Spectral Clustering on the feature matrices, in order to address some of the failings we observed in our decision schema.

References

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. 2011. Part-of-speech tagging for Twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, pages 42–47.

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.

Lingpeng Kong, Nathan Schneider, Swabha Swayamdipta, Archna Bhatia, Chris Dyer, and Noah A. Smith. 2014. A dependency parser for tweets. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1001–1012.

Yasuhide Miura, Shigeyuki Sakaki, Keigo Hattori, and Tomoko Ohkuma. 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)*, pages 628–632.

Saif Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets. In *Proceedings of SemEval-2013*, pages 321–327.

Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Sara Rosenthal, Preslav Nakov, Svetlana Kiritchenko, Saif M Mohammad, Alan Ritter, and Veselin Stoyanov. 2015. Semeval-2015 task 10: Sentiment analysis in Twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, SemEval '2015.

Xiaodan Zhu, Svetlana Kiritchenko, and Saif Mohammad. 2014. NRC-Canada-2014: Recent improvements in the sentiment analysis of tweets. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 443–447.