Applying Contextual Polarity to Tweet Classification

Justin Drew
Department of Computer Science and Electrical Engineering
University of Maryland Baltimore County
Baltimore, MD, 21250
jdrew3@umbc.edu

Hope Miller
Department of Computer Science and Electrical Engineering
University of Maryland Baltimore County
Baltimore, MD, 21250
hmiller3@umbc.edu

Brett Smith
Department of Computer Science and Electrical Engineering
University of Maryland Baltimore County
Baltimore, MD, 21250
bsmith11@umbc.edu

Mohammad Umair
Department of Computer Science and Electrical Engineering
University of Maryland Baltimore County
Baltimore, MD, 21250
ip24496@umbc.edu

Abstract

This paper aims to take the work described in "Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis" (Wilson et al, 2005) and implement the features they discuss for use in the sentiment classification of tweets. Input tweets will be classified as positive, negative, or neutral. Our approach uses the scikit-learn toolkit's SGDClassifier as a MaxEnt model. In the experimentation phase, the model is evaluated on development split with different features on each phase in order to discover which produce the highest Macro F_1 score. Results are compared to a baseline model that only takes unigram features into account. Our final results show significant improvement within our domain, but performs similarly to the baseline when applied to a new dataset.

1 Motivation of Problem

Wilson et al (2005) describes a novel approach to phrase-level sentiment analysis to distinguish between prior polarity and contextual polarity of a phrase using two-step classification. Initially, phrases are classified as either neutral or polar, and in the second step, items labeled as "polar" are further disambiguated as having positive, negative, neutral, or both positive and negative polarity. This significantly improved both the positive and negative F_1 scores for each label over the baseline classifier which only had features defined on the unigrams in a given phrase. Given that this approach was so successful, we expect that we would observe similar success for such a classifier applied to tweets.

Raghuwanshi et al (2017) states that sentiment analysis on short-texts has become one of the major NLP concerns. The idea of being able to extract useful information such as intent, sentiment and opinion from a text has been presented by many researchers in the past, and more recently has been applied on social media sites like Twitter.

The classification of sentiment in tweets started with the work of Go et al (2009), which lead to the creation of the online Sentiment140 API for applications to be able to send any tweet to be classified as positive, negative, or neutral. Go et al (2009) points out that most of the past work on sentiment analysis at the time had been on classifying reviews, which was inherently different from tweet classification due to the more casual nature of "microblogs."

Go et al (2009) also cites Wilson et al (2005) as a recent research on phrase-level sentiment analysis, leading us to conclude that the idea of being able to classify the sentiment from a short

text (single sentence) is a sensible motivation to try a similar approach to tweet classification.

Kiritchenko S. (2014) presents a system that use short-text to extract sentiment from a large tweets dataset, whereas Wen et al (2015) explains how we can use certain features and measures to understand any general short-text classification task. For sentiment analysis datasets on twitter, Saif, Hassan, et al (2013) details 8 free datasets and how they can be used for such tasks. It helps to gain an understanding on how different approaches are used for similar tasks, since we are working on one such novel approach and applying it to a different dataset of tweets, which can be as short as a sentence or a phrase.

2 Description of Problem

We take a large corpus of previously annotated tweets directed towards different airlines scraped from Twitter in February 2015 that have been labeled by human annotators as having "negative," "neutral," or "positive" sentiment. Following the work laid out in Wilson et al (2005), we perform the task of a two-step sentiment classification. However, none of our data had a label of "both" for positive and negative sentiment, so that class is dropped in our implementation.

3 Description of Solution

To perform both steps of the classification, we use SGDClassifier from Scikit-learn toolkit and set the loss function to be logarithmic. This makes the SGDClassifier act as a MaxEnt model optimized via Stochastic Gradient Descent (Pedregosa, 2011). Features detailed in Table 3 are extracted during the data cleaning phase, followed by formatting the text and extracted features to the file structure used by the scikit-learn load_files function with a constant random seed to keep the file order constant across features.

CountVectorizer from scikit-learn is used to fit training data and transform it into a matrix of feature values. This results in 150,090 feature vectors being created off the training split in the final model. Scipy (Virtanen, 2019) sparse matrix representation keeps the memory needs low. Lastly, to combine all the feature vectors to a single matrix, we use Scipy Hstack function. After being fit, the CountVectorizer function also doubles as a set of feature functions defined on what was seen in training for a given feature class, transforming any held-out data to a matrix of the dimensions expected by the model.

Scikit-learn's GridSearchCV searches a provided parameter grid for the combination of values that maximizes the likelihood a classifier assigns to the training set. In the final evaluation, we use CalibratedClassifierCV to calibrate our probability distribution using the development split. According to the documentation, this produces more accurate class probabilities, but may not actually change what labels are assigned relative to the uncalibrated model. Still, the possibility of a change resulting from this calibration makes it worth doing.

4 Related Work

Nakov et al (2016) includes discussion of a subtask much like our problem: predicting whether a given tweet about a known topic expresses positive or negative sentiment toward that topic. This subtask performs its classification in one step as opposed to the two-step solution we're adapting from Wilson et al (2005). Furthermore, rather than MaxEnt models like ours, the model cited as most effective type is a Recurrent Neural Network. They found this model to be a "universally strong" approach which allowed them to get similar evaluation results even across different domains, i.e. systems trained on tweets performed well on SMS message.

MaxEnt was the approach used by Go et al (2009) for their tweet sentiment analysis and they specifically mention having high accuracy across separate topics to the way their training data was constructed. Rather than rely on manual annotations, they use "distant supervision" by trusting emoticons as labels to create a tweet sentiment corpus of over a million observations. With that in mind, it was clear that our model was likely to underperform if applied on tweets that weren't about airlines. Indeed, as part of our end evaluation, we attempt classification of

tweets on a novel topic, and scores on both steps of the classification are unimpressive.

The work described by Kouloumpis et al (2011) is closer to what we're attempting, but still dwarfs us in terms of scope and training data size. It functions as an investigation of the usefulness of lexical resources like prior-polarity lexicons for the informal language common to Twitter. They also leverage the presence of words in all-caps or having added character repetitions as informal intensifiers, a feature that we implemented for the same reason. They implemented a part-of-speech feature, which is common to Wilson et al (2005) but is beyond our present capabilities to use ourselves. However, they concluded that such tags weren't all that useful for sentiment analysis on Twitter, which bodes well for our model.

There are numerous other works focused on Twitter sentiment analysis that attempt similar and very different approaches to this task. For example, Saif et al (2012) introduces a feature where semantic concepts (e.g. "person", "place", or "city") that represent entities are extracted from the tweets and used in training. This is done because certain entities and concepts tend to have a more consistent correlation with positive or negative sentiment. We do not incorporate this into our model, but it definitely seems effective. This is contrary to Zhang et al (2011), who had a similar approach to us in which they classified tweets as "opinionated" or not. If opinionated, then the tweets were further classified as positive or negative. In our model, we classify as "polar" or "neutral," which is quite similar.

5 Pre-Processing and Featurization

For the purposes of data cleaning and feature extraction, we work primarily through the tidyverse package collection in R. The first consideration we make is the confidence score our corpus assigns to the gold-standard sentiment labels. To get the most reliable labels, we chose to only consider tweets that are labeled with 100% confidence. This reduced the corpus size from 14,640 to 10,354 observations.

Next, we replace all usernames in a tweet with "@user" in order to reduce the sparsity in the final feature matrix that could result from a given user only being mentioned in a single tweet. We also remove any URLs and non-ASCII characters. At each step of this process, the intermediate strings are saved for the purposes of feature extraction based on regex matching. One such feature we had removed in the final cleaned text were emoticons, which were the "noisy labels" Go et al (2009), and as such were believed to be valuable to the classification.

Two important features that could not be extracted from the text alone were the prior-polarity of typical unigrams and that of hashtags, both of which require lexicons to compare the tweets to. Finding a lexicon to suit the latter was simple. The National Research Council Canada (NRC) has several sentiment lexicons available, including one for common Twitter hashtags. We found all of the NRC lexicons to be rather small, but considering that hashtags have to be common by their very nature (Mohammad, 2012), we feel confident in using it for this purpose. For a general unigram polarity lexicon, we looked to the SentiWordNet 3.0 corpus (Baccianella, 2010). This corpus is organized by word sense, so we pre-process it to get the unique unigrams and encode their polarity as the union of all the polarities by any given sense of the word. This results in much of the lexicon having words with multiple polarities. We still separate them into positive, neutral, and negative categories, but further classify them as a "strong" polarity clue if they'd only been classed as either positive or negative, and a "weak" clue if they had multiple polarities in an attempt to mimic the strong and weak subjectivity clues used as features in Wilson et al (2005). Neutral words are not classified as "strong" in order to make this feature more useful in the step-1 classification of neutral versus polar. Finally, the data is divided into training, development, and testing splits for the purpose of learning, optimizing, and evaluation.

Table 1. Number of tweets in each split by Label

	Negative	**Neutral**	**Positive**
Training	5000	1000	1000
Dev.	1089	274	241
Test	1250	250	250

6 Experimentation

To evaluate the performance of our models, we considered two common metrics: accuracy and the F-measure statistic. Given the class imbalance evidenced in Table 1, we view the F-measure as more valuable.

$$(1)\ F_\beta = \frac{(1+\beta^2)\bullet precision\bullet recall}{(\beta^2\bullet precision)+recall}$$

$$(2)\ F_1 = \frac{2\bullet precision\bullet recall}{precision+recall}$$

Equation (1) is a general form of the F-measure, an evaluation metric based off the precision and recall of a classification, and the β parameter determines which is weighted as more important. Equation (2) is the balanced F-measure, and as we have no reason to believe either statistic is more or less important than the other in our classification, we chose that as our evaluation metric.

The SGDClassifier offers several parameters that can be changed to produce different results. We chose "alpha", the constant for the regularization and learning rate; and "l1_ratio," which determines whether the model uses L1 regularization (1), L2 regularization (0), or a mix of these two. GridSearchCV performs an exhaustive search over this parameter grid using 5-fold cross-validation to determine which combination produced the highest score for the holdout data. This search is performed as part of the model training.

We began our experimentation with three relatively simple models, and the results of evaluating them on the development split are summarized in Table 2.

Table 2. Possible Baselines

Model	**Step 1 Macro F_1**	**Step 2 Macro F_1**
Unigram	0.78	0.69
(1,3)-gram	0.79	0.68
(1,3)-gram + Prior-Polarity	0.77	0.7

This gave us an idea of where the N-grams and prior-polarity would be most useful, which turned out to be in Step 1 and Step 2, respectively. For the full model, we considered the features in Table 3.

Table 3. Feature Descriptions

Word Features	1. Word token 2. Word bigram 3. Word trigram 4. Word prior polarity 5. Hashtag prior polarity
Sentence Features	1. Topic (Airline) 2. Number of strong polarity clues 3. Number of weak polarity clues 4. Presence of cardinal number 5. Presence of ALL CAPS 6. Presence of character repetition 7. Presence of "!" 8. Presence of "?" 9. Presence of ":)" 10. Presence of ":(" 11. Number of users mentioned 12. Number of Hashtags 13. Presence of "Negative Reason"*

*Tweets like "4 flights Cancelled Flightled in one week" [sic] contain special phrases that make no sense in context, so we detect them explicitly.

Word Features (WF) 1-4 and Sentence Features (SF) 1-4 were found in Wilson et al (2005). After we came up with the rest, we did more research and noticed some commonalities.

Table 4. Features in Related Work

Paper	**Features**
Wilson et al (2005)	WF1-4, SF1-4
Mohammad (2012)	WF5
Kouloumpis et al (2011)	SF5-6
Zhang et al (2011)	SF7-8
Go et al (2009)	SF9-10
Original	SF11-13

Based on the features, we performed four experiments to determine which features would be most useful for each step of the classification:

Wilson: Include all features mentioned or inspired by Wilson et al (2005), with the exception being that, based off the results in Table 2, we exclude WF4 from Step 1.

Step 1: WF1-3, SF1-4

Step 2: WF1, WF4

PastWork: Include all the features that were also seen in related work, in the appropriate

steps (i.e. if a given work performed positive-negative classification in one step, those features could only be used in Step 2).

Step 1: WF1-3, SF1-8
Step 2: WF1, WF4-5, SF9-10

ObsBased: Building off the features from PastWork, we add in the features we extracted that we didn't find parallels to in our research, and adjust which step each feature is used on based on our beliefs about that data.

Step 1: WF1-3, SF2-8, SF11-12
Step 2: WF1, WF4-5, SF1, SF9-10, SF13

Synthesis: Based on the results of all previous experiments, we attempt to build a system that outperforms our previous ones by changing feature placement and inclusion.

Step 1: WF1-3, SF1-8, SF11-12
Step 2: WF1, WF4-5, SF1, SF7-10, SF13

In each experiment, we evaluated the models on the development split with the appropriate features incorporated.

Table 5. Wilson Experiment Summary

	Precision	**Recall**	**F_1**
Step 1	0.77	0.82	0.79
Step 2	0.68	0.6	0.62

Compared to our results with only the N-grams and prior-polarity as features, the Wilson experiment improves the step 1 classification, but leads to a step 2 performance that is worse than even the unigram baseline. Based on what occurred in later experiments, we believe that this occurs due to the fact that higher F_1 in step 1 is due to the neutral class having high recall and moderate precision rather than vice versa.

Table 6. PastWork Experiment Summary

	Precision	**Recall**	**F_1**
Step 1	0.84	0.78	0.80
Step 2	0.71	0.64	0.67

In PastWork, the recall and precision values of the step 1 neutral class are swapped, leading to a similar first F_1 score, but notably improving the step 2 F_1 score. Because features were added to both steps, it is difficult to conclude at this point whether the step 2 performance was driven more by the step 1 difference or the new step 2 features.

Table 7. ObsBased Experiment Summary

	Precision	**Recall**	**F_1**
Step 1	0.84	0.78	0.80
Step 2	0.78	0.68	0.72

Table 8. Synthesis Experiment Summary

	Precision	**Recall**	**F_1**
Step 1	0.86	0.76	0.80
Step 2	0.79	0.71	0.74

ObsBased served as a baseline for the iterative Synthesis experiment. Most surprising from this phase was the realization that Topic was informative to both steps in terms of increasing the F_1 score. Another major finding made here was that adding Hashtag Prior-Polarity to step 1 improved the precision of its neutral class. This corresponded with an increase in performance for the neutral class during step 2 across all metrics. We believe that this correlation is a significant factor in explaining why the final Synthesis model outperforms all others. A possible explanation for this is that the neutral class having high precision and middling recall during step 1 reflects a model that doesn't initially capture a sizable portion of the true neutral class. This leaves more true neutral tweets for the to be captured in step 2, which is in contrast to earlier experiments with higher recall in step 1 that may have left only the neutral tweets that were most difficult to classify as such to step 2, causing that neutral class's performance to suffer.

7 Results

To stay comparable with the work done in Wilson et al (2005), we used the unigram-only MaxEnt model as our baseline and found that the

features used in the Synthesis experiment led to notably superior step 1 classification and a slight improvement for step 2 that again appears correlated with the Synthesis features leading to much higher precision in the step 1 neutral class.

Table 9. Final Evaluation Summary

Model	Step 1 Macro F_1	Step 2 Macro F_1
Unigram only	0.81	0.69
Synthesis	0.84	0.71

8 Limitations

Wilson et al (2005) used structure features that relied on syntactic clues that we couldn't exploit due to our lack of a parser. Moreover, based on the findings of Kouloumpis et al (2011) about Tweet sentiment and standard lexical resources, it is possible that we would have needed a parser built for Twitter specifically to hope to benefit from such syntactic features. Furthermore, due to the lack of explicit syntax clues, we could not reap the full benefits the SentiWordNet 3.0 corpus is capable of giving, leading to a weaker prior-polarity lexicon.

Our research led us to believe that our model would not do well when applied to a new topic. We tested this on a dataset about self-driving cars after subjecting to the same processing as our original data and found that we were correct.

Table 10. Cross-Domain Evaluation Summary

Model	Step 1 Macro F_1	Step 2 Macro F_1
Unigram-only	0.59	0.32
Synthesis	0.50	0.36

Based on the full model being beaten on step 1, we believe that even the phrases expressing sentiment are unique across different topics, to say nothing of features like SF13 that were highly specific to our data.

9 Potential Follow-Up Work

Sentiment Analysis tasks have a wide scope and comes with new ideas for potential follow-up. Generally, the tweets model we used can be tested with various other tweet datasets as well as datasets of text messages, Facebook posts and other short-text problems. E.g. Expanding our training data to larger, existing corpora like the Sentiment140 dataset (Go et al, 2009) or the other corpora used by Kouloumpis et al (2011) would eliminate our constraint to airline-related sentiment classification. There are a number of rich features that can further be implemented and tested with our approach as mentioned in earlier sections. i.e. POS-tagger feature by Wilson et al (2005) and Kouloumpis et al (2011), extraction of semantic concepts from tweets to represent entities in Saif et al (2012). Moreover, this approach can be tested with other languages like Dutch, Spanish etc, and computation models like Advanced Neural Networks.

Acknowledgments

The datasets we're using have been provided by *Data for Everyone* hosted at https://www.figure-eight.com/data-for-everyone/ under the CC BY-NC-SA 4.0 license.

The documentation for the Sentiment140 classifier referenced and hosted at http://help.sentiment140.com/home was extremely helpful in clarifying both motivation for this sort of problem and also the validity of our approach to finding a solution.

The vast documentation at scikit-learn. org was instrumental in the design of our models, featurization, and experiments.

SentiWordNet is distributed under the CC BY-SA 4.0 license.

This research makes use of the NRC Hashtag Sentiment Lexicon, created by Svetlana Kiritchenko and Saif M. Mohammad at the National Research Council Canada. The data is hosted at http://saifmohammad.com/WebPages/AccessResource.htm. Papers associated with this resource are Kiritchenko et al (2014), Mohammad et al (2013), and Zhu et al (2014).

References

Baccianella, S., Esuli, A., & Sebastiani, F. (2010). Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In *Lrec* (Vol. 10, No. 2010, pp. 2200-2204).

Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, *1*(12), 2009.

Kiritchenko, S., Zhu, X., Mohammad, S. (2014). Sentiment Analysis of Short Informal Texts. *Journal of Artificial Intelligence Research*, 50:723-762, 2014.

Kouloumpis, E., Wilson, T., & Moore, J. (2011). Twitter sentiment analysis: The good the bad and the omg!. In *Fifth International AAAI conference on weblogs and social media*.

Mohammad, S. M. (2012). # Emotional tweets. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation* (pp. 246-255). Association for Computational Linguistics.

Mohammad, S., Kiritchenko, S., Zhu, X. (2013). NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets. In *Proceedings of the seventh International Workshop on Semantic Evaluation Exercises (SemEval-2013)*, June 2013, Atlanta, USA.

Nakov, P., Ritter, A., Rosenthal, S., Sebastiani, F., & Stoyanov, V. (2016). SemEval-2016 Task 4: Sentiment Analysis in Twitter. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, D., Brucher, M., Perrot,M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

Raghuwanshi, A. S., & Pawar, S. K. (2017). Polarity Classification of Twitter Data using Sentiment Analysis. *International Journal on Recent and Innovation Trends in Computing and Communication*, *5*(6), 434-439.

Saif H., He Y., Alani H. (2012) Semantic Sentiment Analysis of Twitter. In: Cudré-Mauroux P. et al. (eds) The Semantic Web – ISWC 2012. ISWC 2012. Lecture Notes in Computer Science, vol 7649. Springer, Berlin, Heidelberg

Saif, H., Fernandez, M., He, Y., & Alani, H. (2013). *Evaluation datasets for Twitter sentiment analysis: a survey and a new dataset*, the STS-Gold.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., ... & van der Walt, S. J. (2019). SciPy 1.0--Fundamental Algorithms for Scientific Computing in Python. *arXiv preprint arXiv:1907.10121*.

Wilson, T., Wiebe, J., & Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*.

Zhang, L., Ghosh, R., Dekhil, M., Hsu, M., & Liu, B. (2011). Combining lexicon-based and learning-based methods for Twitter sentiment analysis. *HP Laboratories, Technical Report HPL-2011*, *89*.

Zhu, X., Kiritchenko, S., Mohammad, S. (2014). NRC-Canada-2014: Recent Improvements in Sentiment Analysis of Tweets. In *Proceedings of the eighth international workshop on Semantic Evaluation Exercises (SemEval-2014)*, August 2014, Dublin, Ireland.

Saif, Hassan, et al. "Evaluation datasets for Twitter sentiment analysis: a survey and a new dataset, the STS-Gold." (2013).

Hua, W., Wang, Z., Wang, H., Zheng, K., & Zhou, X. (2015, April). *Short text understanding through lexical-semantic analysis*. In 2015 IEEE 31st International Conference on Data Engineering (pp. 495-506). IEEE.