Sentiment Analysis on Text

Jae-Eun Lim

May 15, 2017

Abstract

Every day, people go through different emotions and they express their emotions on social media like Twitter. This paper uses machine learning techniques to classify Twitter posts into one of twelve categories of emotion: empty, worry, neutral, sadness, boredom, love, hate, enthusiasm, happiness, surprise, relief, and fun. Emotion detection allows for more realistic interactions between human and computer, especially in fields like e-learning environment or online customer services.

1 Introduction

The advent of internet and social media has reduced the total amount of time people spend in activities that involve human to human interactions. In human to human interaction, people respond to each other's concerns, wants, and needs. What human-computer interaction lacks is the immediate and appropriate response to the user's input.

The ability of a computer to detect emotions is becoming increasingly important as many activities are going online. In particular, online courses essentially replace teachers with computers. Hence, it is important that the online course provides similar environment to that of a real classroom. For the computer to be able to respond to the student's feeling towards the course and his or her progress, it needs to be able to detect emotions [9]. Similar principle applies to customer services of online shopping.

This paper uses supervised machine learning techniques to analyze sentiments of text, which is the most common form of online communication. It seeks to find the most effective way to accurately classify Twitter posts into one of twelve categories of emotion: empty, worry, neutral, sadness, boredom, love, hate, enthusiasm, happiness, surprise, relief, and fun. It does so by extracting bigram and regular expression feature space from texts in addition to unigram feature space in order to take into account the broader context of keywords. Moreover, this research ignores function words, words without affective meanings, to further increase the accuracy of emotion detection.

This paper is organized as follows. Section 2 explores previous work on sentiment analysis and their limitations. Section 3 outlines the procedure of the analysis. Section 4 outlines how the data was divided into development, cross-validation, and test sets. Section 5 explores error analysis. Section 6 explains the baseline performance. Section 7 explores parameter tuning. Finally, Section 8 presents the final results.

2 Related Work

This section explores previous work on sentiment classification of texts and their limitations.

Some of the previous research on sentiment analysis involved direct input of a list of affective words for each category of

emotion. For instance, Pang and Lee asked two graduate students in computer science to make a list of indicator words for positive and negative sentiments in a movie review [6]. Talmy used force dynamics theory to identify words in text that counteract each other with opposite forces, or opposite meaning, with different magnitudes, or different intensity of meaning [3, 10]. The disadvantages of these approaches, however, lies in the fact that the process is manual. They require humans to read over the texts and identify words. Not only is this process time consuming, but it is prone to prejudice of an individual and inconsistency due to variable condition of the individual.

Osgood's work on understanding emotions in texts used keyword-based detection [7, 8]. For each keyword, he identified three dimensions: evaluation, activity, and potency. Evaluation measured how much a word expressed pleasant or unpleasant feeling. Activity measured whether the nature of a word was active or passive. Potency measured the intensity of a word's emotion. However, there are limitations to using only keywords to classify sentiments. First, many words have multiple meanings and this cannot be distinguished with keyword-based detection. Second, some emotion-filled sentences or phrases do not contain keywords [1].

In their sentiment analysis, Hsu, See, and Wu placed weights on function words such as 'and', 'the', and prepositions [5]. These words, however, are usually insignificant in terms of meaning. Therefore, the presence of these words may cause distraction regardless of their weights.

This paper's sentiment analysis accounts for the limitations of previous work. First, it does not manually generate a list of affective words, which is labor intensive and time consuming. Second, instead of using keyword-based detection, this research uses bigrams and regular expressions to account for surrounding words which reveal the context of the keywords. Third, this research ignores determiners, prepositions, and pronouns that don't have affective meanings and cause distractions. Lastly, while most of the previous work mentioned used binary classification, i.e. positive or negative sentiments, this research uses multiple categories of emotions—the benefit of this is that it will have broader range of applications.

3 Procedure Outline

The purpose of this paper is to train a classifier that can predict the sentiment of Twitter content. This section outlines the procedure, which is performed in LightSide.

First, the dataset is prepared for analysis by removing unnecessary instance attributes and dividing into development, cross-validation, and holdout sets (Section 4). Then, error analysis is performed on the development set to determine which algorithm performs best and then to improve feature selection and increase accuracy (Section 5). The best algorithm selected in Section 5 with default setting is used on cross-validation set to obtain baseline performance (Section 6). Next, parametric optimization is performed in which the optimized performance is compared with the baseline (Section 7). Finally, the optimized model is trained and built on cross-validation set and is used to test on holdout set (Section 8).

4 Data Preparation

This paper uses a dataset with 8000 instances of Tweeter posts. Each instance has been labeled based on emoticons and hashtags used. Only the posts with emotion

indicators have been collected for the purpose of this research. Each instance has been given one of the twelve sentiments: empty, worry, neutral, sadness, boredom, love, hate, enthusiasm, happiness, surprise, relief, and fun.

The dataset has been divided into 1600 instances for development set, 4800 instances for cross-validation set, and 1600 instances for holdout set.

Table 1 shows the instance attributes of the original dataset. It contains four attributes: tweet_id, sentiment, author, and content. sentiment is the class attribute. tweet_id and author are unique to each instance, so they are not useful in the analysis and have been removed. content attribute is the text from which features are extracted.

To prepare the dataset for analysis, the spellings in the content attribute of each instance were corrected in order to prevent overfitting due to rare appearances of each of the misspelled words.

Table 1: Original instance attributes

Attribute	Type	Explanation
tweet_id	numerical	User ID number
sentiment	nominal	Emotion class
author	nominal	Username
content	nominal	Posted texts

5 Error Analysis

Error analysis was performed on the development set with 10-fold cross-validation. First, unigram feature space was extracted from the dataset. Then, models were built using four different algorithms: Naïve Bayes, J48, SVM, and Logistic Regression.

5.1 Selecting Best Algorithm

All of the models were built using default settings. Naïve Bayes model was built without any special configurations. J48 model was built configured to pruning with two minimum objects in leaves. SMO model was built with normalized nominal class values and LibLINEAR setting. Logistic regression model was built using L2 Regularization setting. The results are summarized in Table 2.

Among the four algorithms, logistic regression model had the highest accuracy and Kappa statistics. Therefore, logistic regression will be used for the rest of the process.

Table 2: Results of multiple algorithms

Algorithm	Accuracy	Kappa
Naïve Bayes	0.2487	0.0746
J48	0.2025	0.0548
SVM	0.2131	0.0729
Log. Reg.	0.2494	0.0866

5.2. Analyzing Errors

Exploring the results from the logistic regression model, the most problematic case identified was *worry* being misclassified as *neutral*, with 112 instances of misclassification.

For this specific case, frequency, feature weight, and horizontal absolute difference of each present feature was observed. Function words have been ignored in this analysis. The most problematic feature with overall highest frequency, weight, and horizontal absolute difference was ‘just’.

Neutral instances that were correctly predicted as *neutral* were compared to *worry* instances that were misclassified as *neutral*. It was noticed that for all except one correctly predicted *neutral* instances, ‘just’ was used to mean ‘simply’, which is logically a neutral term that does not significantly affect the meaning of the phrase when omitted. For instance, “just driving”, “it was just the handle”, or “things

like these just take time." On the other hand, all misclassified instances used 'just' to mean 'very recently', which introduces a sense of urgency. For instance, "just left" or "just saw something." When 'just' is omitted in this case, the sense of urgency is also removed and thus the overall meaning of the phrase is affected.

These distinct meanings of 'just' used for *worry* and *neutral* instances can be very useful in correctly predicting the instances.

In addition to the horizontal absolute difference, the vertical absolute difference was also observed. The most problematic feature with overall highest frequency, weight, and vertical absolute difference was 'go'.

Neutral instances that were correctly predicted as *neutral* were compared to *worry* instances that were misclassified as *neutral*. It was noticed that for most of *worry* instances, 'go' follows a negative term such as "doesn't go there anymore" or "I can't go to Spain". Some instances are followed by 'but', such as "I went to go visit you but they wouldn't let us in the school!" On the other hand, most of *neutral* instances did not contain negative terms or 'but' in the text.

These distinct characteristics of 'go' used for *worry* and *neutral* instances can be very useful in correctly predicting the instances.

To account for the distinctions between *worry* and *neutral* instances based on 'just' and 'go' features, two extra feature spaces were added. Bigram feature space was added so that the neighboring words were taken into account to better understand the context of 'just' and 'go'. This would help the model learn to distinguish the two different meanings of 'just' and recognize negative terms accompanying 'go'. The second feature space added was regular expression, 'no|not|n't|but'. These four words or structures all connote negative meaning. Tying them together as one would not only help the model learn that all of these words have negative meanings but also reduce overfitting due to too many unnecessary distinct instances.

Using these extra feature spaces, logistic regression model was built on the development set. This model had a higher accuracy of 25.37% and Kappa statistics of 0.0946.

Table 3: Logistic regression model results

Feature	Frequency	Absolute Difference	Feature Weight
just	6	0.0194	0.0704
go	5	0.0249	0.0629

Table 4: 'just' feature characteristics

Actual-Predicted	Meaning	# of Instances
Neutral-neutral	simply	7
	very recently	1
Worry-neutral	simply	0
	very recently	6

Table 5: 'go' feature characteristics

Actual-Predicted	Used with	# of Instances
Neutral-neutral	negative terms	1
	'but'	0
Worry-neutral	negative terms	1
	'but'	1
Worry-worry	negative terms	3
	'but'	1

6 Baseline Performance

Using the default setting in LightSide, the logistic regression model was built on the cross-validation set with 10-fold cross-validation.

The resulting baseline performance had an accuracy of 32.67% and Kappa statistics of 0.1004.

Table 6: Baseline performance

Accuracy	Kappa
0.2974	0.0046

7 Optimization

This section explores parameter tuning to find the optimal parameter setting for logistic regression algorithm. CVParameterSelection from Weka with five folds cross-validation was used.

The parameter tuned for logistic regression was M, the maximum number of iterations to perform. M=1, 5, 10, 50, 80, and 100 were used. M value was not tuned beyond 100 because somewhere between 50 and 80 the performance value started to converge. At M=100, it was clear that the performance wouldn't change even if M is increased any further. The accuracy ranged from approximately 29% to 30% which was a negligible difference, so Kappa statistics were used to compare different settings. The results of the performance in Kappa statistics are summarized in Table 7. M=5 had the best performance with Kappa statistics of 0.0072.

Table 7: CVParameterSelection results in Kappa

M=1	M=5	M=10	M=50	M=80	M=100
0.0067	0.0072	0.0046	0.0061	0.006	0.006

To further tune the parameter, M=4 and 6 were used since they are neighbors of M=5. M=4 had Kappa statistics of 0.0045 and M=6 had Kappa statistics of 0.0067. Both were lower than that of M=5. Therefore, M=5 was the most optimal setting. Five maximum number of iterations is relatively a low value, so it would have low computational cost.

Lastly, the baseline and M=5 performances were compared. The accuracy and Kappa statistics of baseline were 29.74% and 0.0046 and those of M=5 setting were 29.64% and 0.0072. The difference in their performances was statistically insignificant (p=0.564, t=0.577). In conclusion, the optimization was not worth doing and the baseline model will be used to perform the final test on the holdout set.

8 Final Results

Finally, the baseline was used to build model on cross-validation set and test on holdout set. The accuracy and Kappa statistics came out to be 21.75% and 0.0021.

Table 7: Final results

Accuracy	Kappa
0.2175	0.0021

9 Discussion

The final results performed more poorly than the model. This is most likely due to overfitting on the cross-validation set. The overfitting may have occurred because, since the cross-validation set is larger than the holdout set, it happened to contain many slangs or abbreviations which are relatively infrequent. These occurrences of rare words could have produced unique features spaces that caused overfitting.

The final results predicted about a fifth of the instances correctly. There is a large gap for improvement, which implies that there are some limitations to the work of this paper. One of the limitations is that some of the categories of sentiments had very few

instances. For example, there was only one instance of *boredom* in the cross-validation set. This uneven occurrence of classes would have made the model-building process more unreliable.

Another limitation of this paper is that it doesn't take into account the fact that some of the sentences or phrases express multiple emotions. Furthermore, the method of initially labeling the instances is not very accurate because by visual inspection it was apparent that some texts and their classes were not the right fit.

For improvements, future work will collect more balanced datasets with approximately equal number of instances of each class of sentiments. Also, future work will use more precise way of labeling the instances and account for slangs and abbreviations via regular expressions. Further down the road, future models will be trained to make predictions for instances with multiple sentiments.

References

[1] Banerjee, S. and Dutta, U. (2015). Detection of Emotions in Text: A Survey. In *International Journal of Advanced Engineering and Global Technology,* volume 3.

[2] Buechel, S. and Hahn, U. (2016). Emotion Analysis as a Regression Problem—Dimensional Models and Their Implications on Emotion Representation and Metrical Evaluation. IOS Press.

[3] Hearst, M. (1992). Direction-based text interpretation as an information access refinement. In Paul Jacobs, editor, *Text-Based Intelligent Systems*. Lawrence Erlbaum Associates.

[4] Hemalatha, I., Varma, S., and Govardhan, A. (2013). Sentiment Analysis Tool using Machine Learning Algorithms. In *International Journal of Emerging Trends & Technology in Computer Science,* volume 2.

[5] Hsu, R., See, B., and Wu, A. (2010). Machine Learning for Sentiment Analysis on the Experience Project. In *Enterprise Risk Management in Finance*.

[6] Lee, L., Pang, B., and Vaithyanathan, S. (2016). Sentiment Classification Using Machine Learning Techniques. In *International Journal of Science and Research (IJSR)*, 5.4: 819-21.

[7] Osgood, C. (1990). Cross-cultural universals of affective meaning. University of Illinois Press.

[8] Osgood, C. and Tzeng, O. Language, meaning, and culture: The selected papers of CE Osgood. Praeger Publishers.

[9] Rodriguez, P., Ortigosa, A., and Carro, R. (2012). Extracting Emotions from Texts in ELearning Environments. In *2012 Sixth International Conference on Complex, Intelligent, and Software Intensive Systems,* pages 887-892. Ieee, July.

[10] Talmy, L. (1985). Force dynamics in language and thought. In *Parasession on Causatives and Agentivity*, University of Chicago. Chicago Linguistic Society (21st Regional Meeting).