A rank coverage analysis on #emotions in Tweets

Naveen Kumar
IMS,University of Stuttgart,
Germany
kumarnn@ims.uni-stuttgart.de

Aswathy Velutharambath
IMS,University of Stuttgart,
Germany
veluthay@ims.uni-stuttgart.de

Abstract

Automatic detection of emotions from short texts published in micro-blogging services like twitter provide powerful insights to the pattern of human communication over internet.In this paper we are investigating the pattern of emotions in twitter messages.We use a supervised classification method to classify emotions into 6 classes - *Happy*, *Sad*, *Anger*, *Disgust*, *Fear* and *Surprise*. To find the best features, we compare the performance of the system using different combinations of features. Further analysis is done on the obtained result using a rank coverage method, in which we predict the top two emotions. The improved results suggest a possibility of close relation or co-occurrence of two emotions.

1 Introduction

In the recent years, social media is being used extensively by individuals for expressing their emotions. It is gaining lot of attention within NLP community as it provides large amount of data for opinion mining and emotion analysis. Twitter is one such source of interest which allows only short text messages.

Emotion analysis on tweets comes with lot of challenges. Tweets are short text messages which are limited to a maximum length of 140 characters which necessitates the expression of emotions in short and precise manner. The range of information conveyed through such messages cover wide range of domains from personal thoughts and opinions to public affairs and social issues. The task is made even more difficult by the informal nature of the text containing slangs, URLs, spelling errors, emoticons, abbreviations such as 'RT' for re-tweets and hashtags.

In this project we are focusing on predicting the emotions expressed in individual tweets.We are approaching this emotion analysis task as a classification problem with different classes as *Happy* , *Sad*, *Anger*, *Disgust*, *Fear* and *Surprise*. The data is preprocessed for normalizing and detecting duplicates.A perceptron learning method with a multi-class perceptron is implemented for classification. We propose different features and as part of feature selection choose the features which contribute towards improving the system performance.

2 Method

2.1 Preprocessing

We use the data crawled from twitter as our training data. This labeled dataset is preprocessed to remove junk information and to obtain a better structured text. Following are the main rules applied as part of the pre-processing steps :

- Urls present as part of the tweets are replaced by the string '[URL]'.
- Usernames in tweets usually starts with @ symbol (e.g. @MyCoolSelf). All strings starting with @ are replaced by [USRNAME].
- Re-tweeted messages contain RT tags. We removed such tags so that identification and removal of duplicate tweets will be easier.
- All hashtags used for crawling are removed from the tweet in-order to avoid biasing for the hashtag class.
- After the above steps are completed, we removed the duplicate tweets by using cosine similarity.

  Original Tweet: *RT @aubordducanal: When #PrayForNigeria is in the top trends,*

no news on what happened on the western tv #sad https://t.co/BN3eA

Preprocessed Tweet Eg. [USRNAME]*: When #PrayForNigeria is in the top trends, no news on what happened on the western tv* [URL]

2.2 Perceptron

In this work, we use a supervised learning method for the classification, given that the number of classes are constant. A Perceptron learning algorithm is used for learning weights for different features which accounts for the dimensions on hyperplane and helps in dividing the feature space. We implement a multi-class perceptron with an idea to find one winning perceptron per class.

For each class label, we implement a perceptron which decides if the tweet belongs to that class or not. During training, we start with random initialization of weights for each feature.The training process involves 'K' number of iterations,where the weights improve with each iteration.During each iteration, we get a winning perceptron with highest value *'y'*, where

$$y = \sum_{i=1}^{m} (x_i * w_i)$$

and *m* is the number of features, *x* is the feature vector and *w* is the weight vector. Considering the winning perceptron as predicted label for the instance in the iteration. We further adjust the weights after each iteration, as explained in Algorithm 1.

Algorithm 1 Adjusting weights in Perceptron Algorithm

```
w(p) ← Weight Vector for Perceptron of Predicted Label
w(g) ← Weight Vector for Perceptron of Gold Label
x ← Feature Vector for the instance
if PredictedLabel = GoldLabel then
    continue
else
    i=0
    while i < m do        ▷ where m is number of features
        w(p)_i = w(p)_i - x_i
        w(g)_i = w(g)_i + x_i
        i++
    end while
end if
```

In testing phase, we get 'y' value for each perceptron and the highest 'y' value perceptron is referred as the predicted label.In the rank coverage setup, we sort the 'y' values to get the sorted order of predicted labels in terms of ranked results.

2.3 Feature sets

We implement different features for emotion classification which includes basic features like ngrams, advanced features like discourse connectives and also external lexical resources.

Ngrams. As a feature, we use unigrams and bigrams extracted from the tweet.

Emotion dictionary. The NRC Emotion Lexicon[1] consists of a list of English words and their associations with eight basic emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust) and two sentiments (negative and positive). The annotations were manually done by means of crowdsourcing. For example, below listed are the entries from the NRC dictionary for the word *abuse*

abuse	anger	1	abuse	anticipation	0
abuse	disgust	1	abuse	fear	1
abuse	joy	0	abuse	sadness	1
abuse	surprise	0	abuse	trust	0
abuse	negative	1	abuse	positive	0

We use the emotions as well as the two sentiments as features. So if the token *abuse* is present in the tweet, we add the features NRC-anger, NRC-fear, NRC-disgust, NRC-negative, NRC-sadness

Discourse feature. We consider few discourse features like conditionals and connectives mentioned in Mukherjee and Bhattacharyya (2012). Presence of a conditional like *if* , *might* etc in a sentence expresses a degree of uncertainty and hence cannot be ignored. So we add conditional tokens as a feature.

In the case of connectives, we take a set of conjunction words which gives more importance to the following part of the sentence.Like in following sentence, the part of sentence after *but* has more relevance than the previous.We add every word after the conjunction as a feature to give more importance to them.

'He wasn't prepared for the exam *but he did pretty well*'.

[1] http://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm

Features Used	Anger	Sad	Happy	Surprise	Fear	Disgust	Micro F-Score	Macro F-Score
Ngram	0.45	0.57	0.79	0.30	0.59	0.23	0.66	0.49
Ngram + NRC	0.44	0.57	0.78	0.30	0.58	0.14	0.65	0.47
Ngram + NRC + Negation	0.44	0.57	0.78	0.32	0.59	0.18	0.65	0.48
Ngram + Negation + Discourse	0.52	0.64	0.82	0.36	0.62	0.26	0.71	0.54
Ngram + Negation + Discourse + Smiley	0.50	0.64	0.82	0.36	0.61	0.32	0.71	0.54

Table 1: Contributions of different features to the f-score of different classes, the micro and the macro F-scores

Negation feature. We use negation feature to identify the tweets which contain negated phrases like '*hardly surprised*' or '*not really happy*'. Such phrases can result in wrong classification if only unigrams and bigrams are considered. We add as negation feature, the three unigram following the negation word , until and unless another negation or connectives occurs which reverse the semantics of negation. For example in the following sentence, we add negation feature only for tokens 'like' and 'Nokia'.'*I do not like Nokia but like Samsung*'.

Smiley feature. People generally resort to smileys to express emotions especially in short text messages. Here we use an external lexicon[2] of smileys in which which smileys are categorized as *Extremely Positive* , *Positive*, *Neutral*, *Negative* and *Extremely Negative*.Below listed are some examples from the lexicon.

:D C:	Extremely-Positive
: −) :) : *o*) :] : 3 : *c*) :>=]8) =) :}	Positive
: |	Neutral
:-(:(:c :[:{	Negative
D: D8 D; D= DX v.v	Extremely-Negative

If the tweet contains smileys listed in the lexicon, we add the smiley appended with its category as one feature. For example, if a tweet goes as "*How lucky was that?* :D", we add the feature as ':D-*Extremely-positive*'

3 Experiment

3.1 Experimental setup

Data. We use the data crawled from twitter using different emotion classes as hashtags and also hashtags of related emotions (eg, #happy and # happiness). This data was used as the training data as they are already annotated with the hashtags.

Feature Selection. In order to select the best features, we add the features incrementally to the model and evaluate using micro and macro F-score. Table 1 shows the result of this feature selection experiment.

From the table it can be seen that after adding the NRC feature, the micro and macro F-score reduced and in the next step, even on adding Negation feature there was no considerable improvement in the F-scores. So we removed the NRC feature and added Discourse features, which improved the micro and macro F-scores significantly.

In the case of Smiley feature, even though it did not contribute much in improving the micro and macro F-scores of all classes, the F-score of the class *disgust* was increased significantly which was otherwise very poor.

4 Result

Best performance for the system was attained when the combination of features *Ngram + Negation + Discourse + Smiley* were used. However poor performance was observed for some classes compared to other, which is why we conducted an error analysis study on the obtained result.

4.1 Error Analysis

The results shown in Table 1 clearly indicate poor performance for the classes *Disgust* and *Surprise* which could be due to the sparsity of the class in training and test data.

On manually analyzing the results, we found that multiple emotions are expressed in the same tweet. For example, from the confusion matrix given in Table 3, we can see that the class *Dis-*

[2]https://github.com/mayank93/Twitter-Sentiment-Analysis/blob/master/phrase-level-analysis/code/emoticonsWithPolarity.txt

Rank Coverage / F-score	Anger	Sad	Happy	Surprise	Fear	Disgust	Micro F-Score	Macro F-Score
1	0.49	0.63	0.81	0.36	0.61	0.32	0.71	0.54
2	0.78	0.84	0.88	0.81	0.81	0.87	0.84	0.83
3	0.88	0.92	0.93	0.9	0.9	0.94	0.91	0.91
4	0.94	0.96	0.96	0.95	0.95	0.96	0.96	0.96
5	0.97	0.98	0.98	0.98	0.98	0.98	0.98	0.98
6	1	1	1	1	1	1	1	1

Table 2: Rank coverage and score per category

gust is predicted as *Sad* 9 times. Also the number of times *Surprise* is predicted as *Happy* is more than *Surprise* itself.This could be because both the emotions are present in the same tweet. In the next section,we investigate further with a second experiment on the data by considering more than one top ranked emotions.

Gold / Predicted	Disgust	Sad	Anger	Happy	Fear	Surprise
Disgust	4	9	0	3	1	1
Sad	2	821	55	184	84	13
Anger	0	133	222	96	60	4
Happy	1	282	48	2321	142	36
Fear	0	137	44	155	519	8
Surprise	0	43	8	78	22	61

Table 3: Confusion matrix result for test data

4.2 Rank Coverage

If there are multiple closely related emotions expressed in the same tweet, the tweet might be classified as the related class and not as the gold class. To investigate on this aspect, we modified the algorithm to consider the top n predicted classes and update the confusion matrix.

As explained in Algorithm 2, if the gold class is present in the top 2 predicted classes, we increment the true positive count for both predictions by 0.5 . In the case where gold label is not present in the top predictions, we increment the false negative for the gold class by 1 and increment the false positive for the predictions by the fraction.

Algorithm 2 Evaluation in Ranking Scenario

$PL \leftarrow$ List of top N predicted labels
$G \leftarrow$ Gold label for the data
if $PL \quad contains \quad G$ **then**
    Increment TP for all PL by (1/(Size of PL))
else
    Increment FN for G by 1
    Increment FP for all PL by (1/(Size of PL))
end if

From the results in Table 2, we can see that the scores improve drastically when the top 2 ranked classes as predicted, which is very evident for the classes disgust and surprise for which data is relatively sparse. The co-occurrence or close relation between emotions predicted together, can be assumed to be the reason for this improved score.

5 Summary

In this paper, we did emotion analysis on Tweets using a supervised classification method. On the results obtained after feature engineering we did an error analysis to investigate further on false positives. We used a rank coverage method, in which we considered top *N* classes, checked for the gold class and updated confusion matrix as per the new algorithm. It was observed that the result improved drastically when the first two ranks where considered which suggested a possibility of co-occurrence of emotions in tweets.

6 Future scope

For further improving the result, we would like to concentrate on the feature improvements by experimenting with features including filtering out the features based on mutual information. Another possibility is to use standard polarity dictionaries to segregate emotions in the feature space.

The rank coverage method and its implications towards the co-occurrence of emotions could be a focus for further research.

References

Subhabrata Mukherjee and Pushpak Bhattacharyya. 2012. *Sentiment Analysis in Twitter with Lightweight Discourse Analysis*. Proceedings of COLING 2012: Technical Papers, pages 1847–1864

Perceptron for Imbalanced Classes and Multiclass Classification 2011 URL : https://www.cs.utah.edu/ piyush/teaching/ imbalanced_multiclass_perceptron.pdf