

LING 573 Emotion Classification Project Report

Xena Grant, Jesse R. Mena
Department of Linguistics
University of Washington
xg123@uw.edu, jesserme@uw.edu

Abstract

Stub

averages in a multi-class evaluation.

1 Introduction

Deciphering emotion from social media posts proves difficult with the absence of facial expressions or vocal cues. Research toward improving text-only emotion classification focuses mainly on English language social media posts, but less so in other languages. EmoEvalEs ([CodaLab Competition](#)) aims to classify emotion in a set of Spanish tweets. The goal is to discern the classification method that yields the highest test accuracy.

2 Task Description

Emotion classes in this task include anger, disgust, fear, joy, sadness, surprise, and other – a category containing neutral or emotionless sentiments. Our primary task is to compare different classification methods and choose the one that produces the highest test accuracy. Following that, we then employ sampling and augmentation techniques to further improve test accuracy, and then finally apply successful methods on a similar dataset to evaluate the generalizability of our process. The dataset contains a collection of tweets posted during the month of April 2019 that encompasses a variety of topics ranging from entertainment to environmental catastrophes. Any hashtags in the dataset were replaced with “HASHTAG” so as to not influence the classifier. The dataset is split into development, training, and test partitions. The evaluation process consists of ranking weighted- F1

3 System Overview

The following in Figure 1 is an overview of the architecture for our primary task:

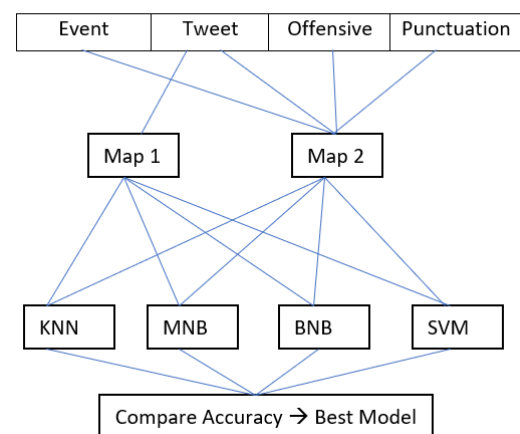


Figure 1

4 Approach

As shown in figure 1, we compared training accuracy across four different classifiers: Multinomial Naïve Bayes, Bernoulli Naïve Bayes, K Nearest Neighbors, and Support Vector Machine. The Naïve Bayes classifiers calculate conditional probability based on either term frequency or binary term presence within each tweet. The K Nearest Neighbors classifier calculates the proximity of a datapoint to others around it, then chooses the class more

prevalent among the neighboring datapoints. SVM classifiers find the maximum marginal hyperplane separating the datapoints and then classifies the resulting groupings. We created two models for comparison. The first is a bag-of-words model reflecting the counts of useful words in the dataset after filtering out stop words identified by TF-IDF score or consisting of non-word characters. The second contains extra features alongside the content of the tweet: the presence of exclamation marks, the event concerning which the tweet was posted, and whether or not the tweet was judged as offensive. For each trial, we created models using the corresponding sklearn implementations on the train data and test on the test dataset. When comparing classification models, we found consistently better performance using SVM when evaluating either model and when combined as an ensemble. We then implemented a SMOTE (Synthetic Minority Oversampling TEchnique) algorithm to create synthetic examples that would balance our dataset. Certain classes such as Fear or Disgust were vastly underrepresented in the dataset. After setting a minimum sampling rate at 500 samples for each class, we used a KNN comparison method to create synthetic data samples, thus further balancing the dataset.

5 Results

Performing classification with a bag-of-words model yielded higher results than when working with the extra features. However, when evaluating both models in an ensemble format, we produced a higher accuracy using SVM.

Method	Without SMOTE	With SMOTE
SVM 1 (BoW)	0.62915	0.62322
SVM 2 (Other Features)	0.59242	0.59242
SVM 1 and 2 Ensemble	0.65284	0.64810

Table 1

Table 1 also demonstrates the effect of adding in SMOTE synthetic data. While we see a similar pattern in performance where bag-of-words outperforms the extra features model and the ensemble outperforms them both, overall the synthetic data did not improve our test accuracy. This can be attributed to an overall imbalance in classes – the smallest class ‘Fear’ held as few as 65 tweets while ‘Other’ had over 2,500. Even with synthetic data added to smaller classes, we will need to employ other methods to further improve performance.

6 Discussion

Stub.

7 Conclusion

Stub

8 References

Stub