# Text Processing Assignment 2 Report

Zhicong Jinag

## Introduction

Users can choose whether a movie is worth their time by reading reviews. The goal of this study is to use the Rotten Tomatoes movie review data set to implement a corpora-based naive Bayes model for emotion analysis tasks. The model is trained by preprocessing and feature extraction of the sentences in the data set, and then the emotion of each sentence is detected and the prediction is formed. The standard results were compared with the expected results, and the validity of the model was evaluated using macroscopic f1 values.

## Progress

Pre-Process: Use stop_word to exclude irrelevant words, as well as lowercase sentences, tokenize sentences, and remove punctuation.

Feature Extraction: Part of speech tagging is used to mark the parts of speech of words, and select adjectives, adverbs, nouns, and verbs as feature words. What's more, I use stemming to reduce words to their basic form and focus on the words that may have a greater impact on sentiment analysis.

Training: Calculate the values needed for various naive Bayes classifier

Predict: According to the values obtained by the training, the features in the phrases are modeled and calculated. It's worth noticing that zero probability can cause the algorithm to crash, resulting in incorrect predictions. when calculating likelihood, Laplace smoothing is applied in order to Prevent zero probability when calculating the probability of a given event:

$$p(t_j|s_i) = \frac{count(t_j, s_i) + 1}{\sum count(t_f, s_i)) + |V|}$$

Finally, the formula given by the Bayesian classifier is used to get the prediction results for the reviews:

$$s^* =_{s_i} p(T|s_i)p(s_i)$$

Evaluation:By comparing the results obtained from the prediction development set with the standard results given by the prediction development set, the F1 value of each class was calculated, and the sum of all the F1 values was averaged to get the macro-F1 value. The higher the value, the better the model performance and the more accurate the prediction.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN}$$

## Result

| Class | Feature | macro-F1 | Accuracy | Run Time |
|---|---|---|---|---|
| 3 | All_words | 0.496701 | 0.6430 | 1s |
| 3 | Features | 0.513008 | 0.6450 | 8s |
| 5 | All_words | 0.302750 | 0.3920 | 1s |
| 5 | Features | 0.333861 | 0.3780 | 8s |

Table 1: Summery Table

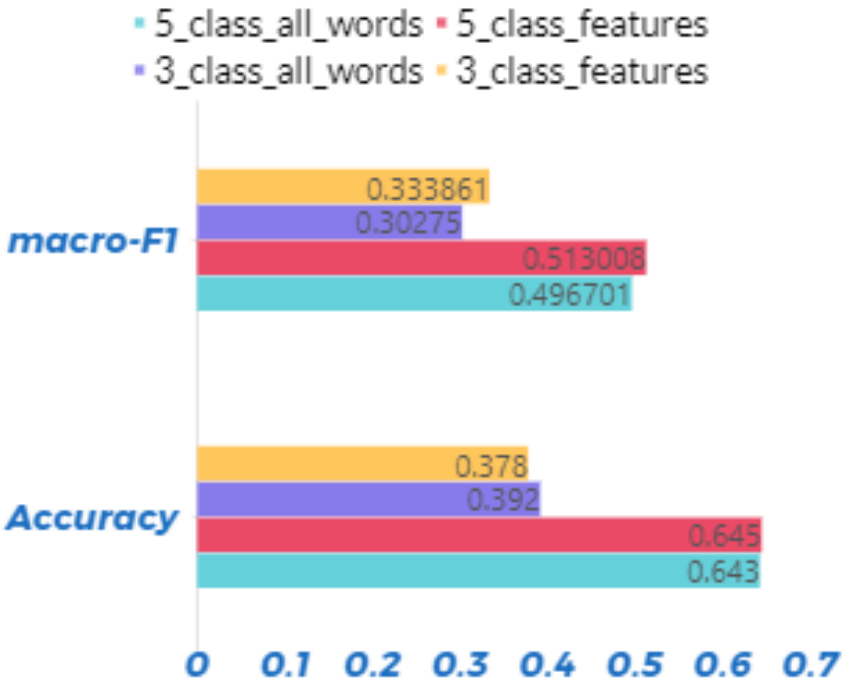


Figure 1: Figure comparison

For the **3-class** with all words considered,the value of macro-F1 is **0.496701** and accuracy is **0.6430**. In terms of features extraction applied,macro-F1 is **0.513008** and accuracy is **0.6450**. The accuracy for the **5-class** with all words taken into account is **0.3920**, and the value of **macro-F1** is **0.302750**. For features extraction, **macro-F1** is **0.333861** and **accuracy** is **0.3780**.

## Analysis

The observed macroscopic f1 value is not high, suggesting that the classifier is not good enough, by the projected outcomes of the training model on the expansion set when compared with the standard response. The fact that the naive Bayes classifier thinks that all characteristics are independent and does not account for covariance, in my opinion, is one of the main causes of the low final accuracy and the low value of macro-F1. The text context will not be taken into consideration to alter the assessment of the emotional polarity of the token, affecting the accuracy of the forecast. However, each feature tends to have distinct emotions in different text context information.

Moreover, it is not difficult to see from the above results that no matter whether 3-class or 5-class, there is little difference between All_words and the results obtained by using feature extraction, but model with features extraction slightly increases the performance neither for 3-class nor 5-class. In the part of feature extraction, I extracted the pre-processed comments, tagged the words by nltk's pos_tag library, and then extracted the adjectives, adverbs, and nouns in the comments as features and stemming the features for model training. After feature extraction, the number of words is reduced to 57,789. The number of features is reduced and the performance of the model is even slightly improved. Therefore, if the data set is large enough, it is necessary to take feature extraction to reduce the memory size of the model and the subsequent training. And emotion prediction time, but feature extraction can also take some time to perform.