Introduction

In the program I designed, the sentiment analysis program for Rotten Tomatoes movie is mainly implemented. Based on the Naive Bayesian model, it mainly implements the training model, prediction and prediction evaluation. The following is the Naive Bayesian model formula mainly involved The N is the total class number :

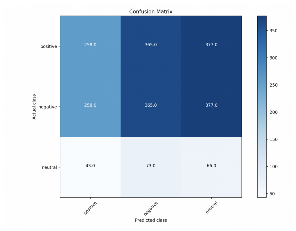
$$macro - F1 = \frac{1}{N}\sum_{i=0}^{N} F1 - score_i \tag{1}$$

Preprocessing in the training model includes: preprocessing, move_comma_s. Used to remove various punctuation marks (except !), remove stoplist, lowercase, remove spaces, and perform the feature selection I designed here, using stemming.The naive Bayesian model used includes: prio_probability: used to calculate the prior probability, likelihood_molecular is used to calculate the molecular part of the likelihood, (the frequency of occurrence of feature words is in a class of a certain type), likelihood_denominator: used to calculate what is required in the likelihood All feature types and the total number of features appearing in a class. In addition, *Laplacian smoothing* is used in these two methods in order to solve the problem of zero probability and prevent the final result from being zero due to the occurrence of zero probability.

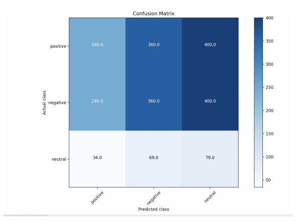
Likelihood_claculate and posteriors calculate the final posterior probability, return the predicted emotional value of each file, and finally f1 required for the calculation of f1 score, at the same time, I also calculated the accuracy and wrote it into a table for more intuitive data representation. The final plot_confusion_matrix method is used to draw the matrix and calculate the value x of each of my models.

Evaluation

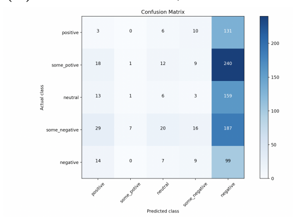
First, the analysis set from movie sentiment is divided into training set, development set and test set. The feature words are: all words, and feature words that have been specially preprocessed and screened, and are trained and developed in the form of 3class and 5class respectively. , and finally select a model with a higher accuracy rate for testing the test set. The confusion matrix of the four methods is shown in the figure:Figure 1



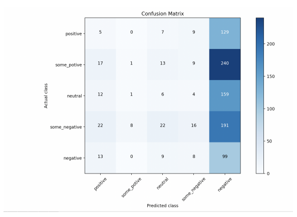
(a) 3-value scale, feature :all-word



(b) 3-value scale, feature:features



(c) 5-value scale, feature :all-word



(d) 5-value scale, feature:features

Figure 1: Confusion matrix

The accuracy profile is shown as Figure 2. Two sentiment scales produced 200 models in total. The best accuracy was obtained when the feature token ratio was 1.0, where accuracy is 0.311 for the 3-value sentiment scale and 0.301 for the 5-value sentiment scale.

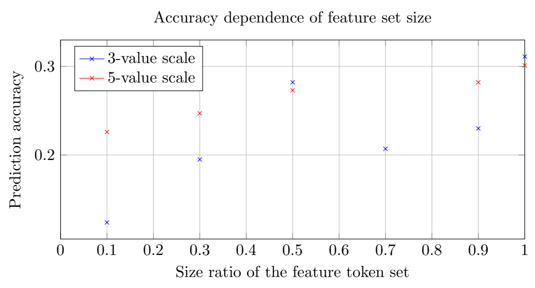


Figure 2: Model accuracy

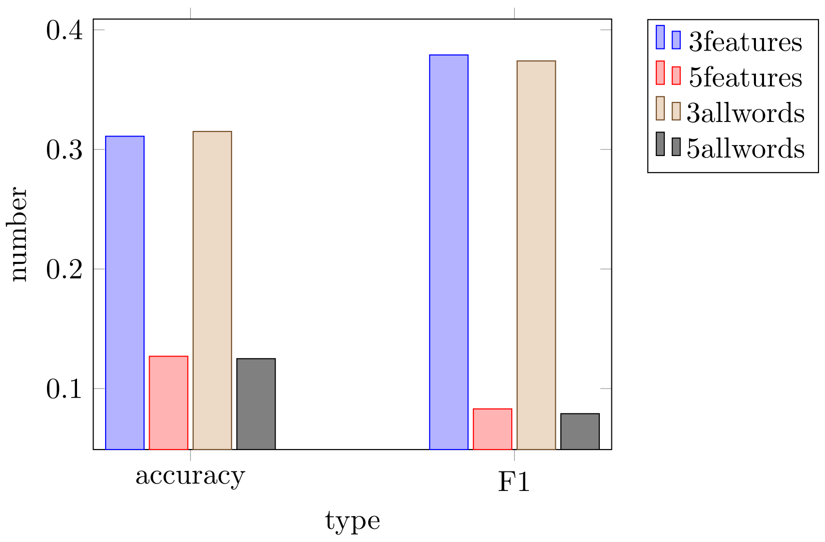


Figure 3: class

Analysis

The accuracy of both classifiers (3-class and 5-class) is not very high, most likely because of the limitations of the Naive Bayes classifier itself, probably because it assumes that all features are independent but text There is a contextual relationship in , so this assumption cannot be fully applied to the classification of text features. Among them, I have tried to use part-of-speech classification to improve accuracy, but unfortunately for the last step: I did not make it very good until the last submission time Embedded into my pandas sequence, I think that although the features of the Naive Bayesian classifier are relatively opposite, if the part of speech analysis and extraction can be realized, the accuracy rate can be effectively improved. In addition, according to the above table, I found that: due to the influence of Laplace smoothing. In the absence of large feature token counts, Laplacian smoothing will tend to take the lower likelihood of feature sentiment with higher prior probability. This leads to the fact that the prior probability of sentiment being high will have a lower posterior likelihood, which goes against the concept of Naive. From the figure 2, it can be seen that the decline in accuracy is approximately proportional to the number of feature cards, which shows that with the emergence of more detailed classification, the misclassification of the Bayesian classifier will be clearly presented