Classifier for Sentiment Analysis on Movie Reviews

Hao-Hsin Shih
hs2762@columbia.edu
Kaggle id: Bryant Shih

Abstract

The purpose for this assignment is to use the training data given to train a model, and use this model to classify the sentiment in the review of a movie. For this assignment, some basic conceptions should be included and implemented. First, Preprocessing: for example, we could normalize the data before we use it to train the model; second, Feature Extraction: we have choose which feature given from training data we want to use and use it to train the model; third, Feature Selection: this greatly influence the model we get, by selecting relevant and good features data for train probably improve the accuracy for the classification well; and most important one, Classification: to decide which classifier algorithm to use is very important, which is the main point for the accuracy to this assignment.

I. Preprocessing

For this project, I spent lots of time thin, king about what we should do in preprocessing, because we could filter some useless (or more precisely, irrelevant) token (word) before training and it could improve the efficiency and accuracy.

A. Converting all letters to lower case:

Changing all letter to lower case so that it would be convenient for us to design the program. But the disadvantage is that some meaningful abbreviation may be ignore this way.

B. Removing punctuation:

We just want to tokenize words into token to count the number and probability of occurrence, so all punctuation should be remove. Let consider some situation for this method, for instance, if we remove "'" from "isn't", then after tokenization we will have 'isn' token as well as 't' token . In this case, we could just see 't' token as 'not' token and remove all 'isn' tokens. We could get lots of 't' tokens from the training data, by Chi-Squared test in feature selection we would discuss late, 't' token get quite big a value which means it play a role in classification, so we should keep it for classifier. There are other example in the same situation we would deal them in the same way as “isn’t” like “didn’t”, ”won’t” and “wouldn’t”…etc.

C. Removing numbers:

For classify the sentiment in the data of movie review, numbers seem not useful for training, most number in data referred to age, year and so on, which is not obviously relevant for the classification.

D. Removing stop words or "too common" words:

For example, we would build a list like [‘a’, ‘the, ‘of’] to filter all ineffective words without tendency of sentiment and remove them. These tokens wouldn’t be helpful to our classifier at all.

E. Stemming:

The reason is obvious for stemming the works. If we don’t do the, tokens which should belong to the same one would be counted as different and have a bad effect to accuracy especially the corpus is large.

II. Feature Extraction

In this assignment, words (tokens) are features we use to do some analysis for classifier to classify. As mentioned above in preprocessing, we just need meaningful words (tokens) to train the model and remove other useful thing like punctuation, stop words and "too common" words.

III. Feature Selection

A. Chi-Squared Test:

I thinks this is most difficult part. People having same classifier may have greatly different accuracy for test data due to feature selection. First, we need one tool to evaluate a weight, which means the importance of the word in this classification. The tool I chose is Chi-Squared test, which is mentioned in the class and be used to test the in dependence for each word. The advantage for it is easy to implement although it’s a rough estimate of confidence. But it accept weaker, less accurate data as input than other parametric test, so it can be used in a wide variety of research context, like the data used in this assignment.

B. Cross Validation:

For selecting good feature to get good model used in classifier, we should have some result for evaluation. Except the train.py and test.py file for requirement for this assignment, train_vc.py is the file I used to do the cross validation to evaluate the accuracy by the feature I select.

The number of document is 6531, so I separate the data into 3 different part. When part I is the test data, part II and part III are training data; part II is the test data, part I and part III are training data; part III is the test data, part I and part II are training data. By doing this, we could avoid the overfeeding problem that we use a file as both training and test file. And the accuracy from cross validation is quite closed to the accuracy I get from Kaggle.

C. Feature Selection:

I do the feature selection by analyzing the influence of words with their length and independent value given by Chi-Squared test.

1. This table is the accuracy in cross validation corresponded to model removing word with length less than L:

L	Accuracy
0	75.3267%
1	75.3109%
2	75.3739%
3	74.8386%
4	73.2483%
5	71.1226%

2. Table by analyzing the influence of removing independent value given by Chi-Squared less than I with removing word with length less than 2:

I	Accuracy
0.000022	75.3739%
0.000487	75.3897%
0.001462	75.3739%
0.5	74.8858%
11	72.0516%

They are just some simple examples I do the feature selection, It usually get a better accuracy on Kaggle.

Finally, I use the result that removing the word with length less than 2 or with independent value less than 0.0005.

IV. Classification

The classifier I decide to implement is Naïve Bayes Classifier, the algorithm for this part is exactly the same with the way in IR textbook. In this classifier, it assume that each feature is independent to the other, and that's the reason I decide to implement Naïve Bayes Classifier, and Chi-Squared test.

Conclusion

By this assignment, I know that even 0.001% vibration of the accuracy is also important in the Machine Learning field and IR field, and so does the rank on Kaggle. So learning how to use efficient algorithm and how to select the feature would be the critical point in this assignment.