Turkish Sentiment Analysis on Matlab using Naive Bayes Classification Algorithm

Oğuzhan Özavcı, Mustafa Ayyıldız
Department of Computer Engineering, Anadolu University, Eskişehir
oguzhanozavci@anadolu.edu.tr, mustafaayyildiz@anadolu.edu.tr

I. INTRODUCTION

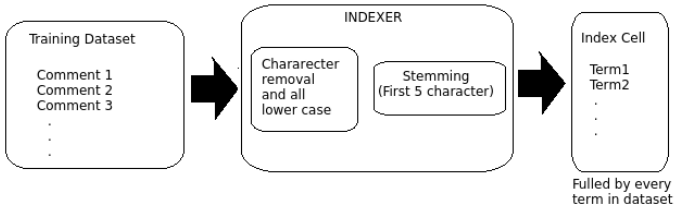
Sentiment analysis is a one of the head topics of Machine Learning and Classification problems, and it is a good example of Text Classification problem. In this project we design a simple sentiment analysis system which learns from a dataset full of class-labeled positive and negative movie comments and indexes them. We used Naive Bayes classification algorithm to decide a comment is positive or negative. And our system uses the %30 of dataset in order to test itself. It classifies the test dataset and controls them according to their labels and calculates an accuracy value and displays it.

II. SYSTEM AND DESIGN

Systems first aim is to creating an index file in order to operate on it. So we wrote a program named “indexer.m”. indexer needs 2 txt files in its directory named “positive.txt”, “negative.txt”.

A. *Preprocessing*

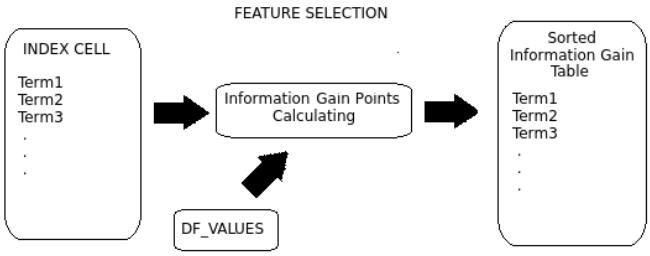
Program first reads every line on the this 2 file and removes all the non-characters and make the line lower-case. After that splits the line into term by spaces on the line. And stems all the terms by their first 5 character and after that preprocessing steps it puts every term to a Map object.(it ignores the duplications in terms.). And it provides an index cell which contains every single term in training dataset.



B. *Feature Selection*

In order to execute a feature selection system calculates the Information Gain points of every term and provides a sorted table ac according to these points.

We aiming to reduce the size of index file so we will get first N terms from this sorted table.

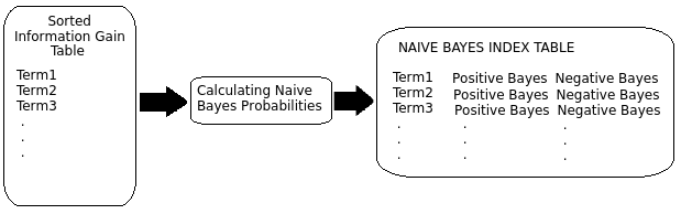


C. *Creation of Naive Bayes Index Table*

After we get the Sorted Information Gain Table our program Calculates the Naive Bayes Probabilities by using the equation below.

$$P(C \mid A) = \frac{P(A \mid C)P(C)}{P(A)}$$

And program adds a positive bayes value and a negative bayes value for every term and provides a Naive Bayes Index Table.



D. *Classification*

In order to classification we wrote a “classify.m” function and to execute this function an index.mat file must be exist in directory. So if you did not you need to run indexer first.

After indexer provided an index file fulled by bayes points of every term our classify program takes a comment (string type) as an argument applies the same preprocessing steps in indexer an gets terms of comments. Then gets the positive bayes probabilities and negative bayes probabilities of every term from index map and multiplies them in order to get a positive and negative probability of this comment. If a term does not exist in index map program gets its probability as 1, so this term does not effect the total probability.

Classify function returns 1 if comment is estimated as positive and returns 2 if negative.

E. *Testing*

In order to test our program “positive_test.txt” and “negative_test.txt” files must be exist in the same directory with test.m function.

Test function reads positive and negative labeled comments from test dataset files and tests the classify method using these test comments. After every test it compares the estimated value with known class label and calculates an accuracy value and displays it in console.

III. References

- https://tr.wikipedia.org/wiki/Naive_Bayes_sınıflandırıcı
- http://ybsansiklopedi.com/wp-content/uploads/2016/09/duygu_analizi.pdf