TITLE: SENTIMENTAL ANALYSIS - MOVIE REVIEWS

ABSTRACT

AIM

To develop a sentimental analysis program which can be used to find out whether given movies review is positive or negative. Using this, we can further scrap the reviews from internet and find the attached sentiments. Distance Measure along with T_f matrix is used for this purpose.

Link of the most related research paper is:
https://arxiv.org/ftp/arxiv/papers/1305/1305.6143.pdf

Although in this paper Naïve Bayes Classifier is used, T_f and distance measure is used for the classification in this project.

PROCEDURE

Sentiment analysis or opinion mining is the computational study of people's opinions, sentiments, attitudes, and emotions expressed in written language. Using the sentimental analysis of the movie reviews, we can find out whether the watcher like it or not and is movie worth watching or not, in general.

The Dataset used is IMdb Users Reviews which contains about 30000 reviews and programming language is Python3.

In this project, *Unigram* model is used because of the fact that the results generated by other N-Gram model where less accurate than those of Unigram model for the given dataset.

Since it written in python, extensive use of python library *NLTK* is done. First of all word tokenization is performed. Then after removing the stop-words, these words along with some of the synonyms are stored in the T_f feature matrix.

The above process is repeated for both positive training samples as well as the negative training samples. The training is saved in a JSON file so that future computation is reduced and efficiency is improved.

Since the training samples have been earlier split into 80-20 ratio of training samples vs test sample, thus the testing is performed on these reviews again after performing similar king of preprocessing. T_f matrix is computed and *Manhattan* distance is used, as the distance measure, to find the distance from each of the positive and negative test samples after normalization.

In this way, the test samples are checked and final accuracy is printed along with the time require for the computation.

CONCLUSION

Thus, the classifier gives an accuracy of about 82.2%. Some of the possible reason of the misclassification is mainly due use of quotes from movie which might make it polarized in a way that is not aligned with the context of the review. The time required for the complete run of the code is around 600 seconds with multiprocessing. Since the training data is preprocessed hence the time required is drastically reduced to around 54 seconds.

Project link (with code and dataset): https://github.com/apoorv698/nlp-project

Submitted by –

Name - Apoorv Sharma

Roll No. - 2K15/IT/015