Alexandria University
Faculty of Engineering
Computer and Systems Engineering Dept.

Fourth Year
Prof. Dr. Nagwa El-Makky
Prof. Dr. Marwan Torki

Assignment 3
Using Classical Machine Learning for an NLP Task

Objectives

1) Become familiar with text preprocessing techniques

2) Become familiar with libraries such as NLTK, Gensim and scikit-learn

3) Applying, tuning and analyzing the performance of different classical classifiers on a large dataset.

Problem Statement

You are given the IMDB movie review dataset, which is a dataset for binary sentiment classification. The IMDB dataset was first proposed by Maas et al. [1] as a benchmark for sentiment analysis. The core dataset contains 50,000 reviews split evenly into 25k training and 25k testing sets. The overall distribution of labels is balanced in both the training and testing sets (25k positive and 25k negative). There are additional 50,000 unlabeled reviews that may be used for unsupervised learning.
The dataset can be found at: https://ai.stanford.edu/~amaas/data/sentiment/
You are required to apply any required text preprocessing techniques on the dataset. Then, you are required to construct different classification models using different approaches, tune the hyper-parameters of these models and compare the performance of the models under multiple factors.

1- Download Data and Apply Text Pre-processing

Text pre-processing is essential for NLP tasks. Check NLTK for available text-preprocessing operations such as: tokenization, stop words removal, stemming, lemmatization, etc.

2- Create a Data Matrix

You will need to convert the text of each review (after pre-processing) into a vector form to construct the data matrix. It is required to consider different alternatives for text representation such as traditional methods (BOW, TF-IDF, etc.) and modern methods such as word embedding.
For traditional methods, use sklearn feature vectorizer (Count, TF-IDF) to create the required feature vector.
For embedding methods, use Gensim library for obtaining fasttext word embedding. You may use pre-trained fasttext models or use the fasttext wrapper in Gensim to generate word embedding. **Do not use the test set in the generation of word embedding.**

3- Classification

In this step, you will apply multiple of classification models (using scikit-learn). Every group is required to apply at least **5** models of the following. It is also required to **tune** the hyper-parameters of these models on a validation set (hold 10% of the training set for validation).
Model choices are

a. KNN
b. Naive Bayes
c. Adaboost
d. Random Forests
e. Linear SVM
f. Non-linear SVM
g. Logistic Regression

4- Evaluation

Compare the performance of the learned models (in terms of accuracy) with respect to the following factors.

a. Pre-processing effect
b. Features choice
c. Classifier choice

5- Report Requirements

Your report should contain the following:

- Plots of the performance results obtained in the evaluation part.
- Comparison and analysis of the performance results.
- Success and failure cases should be presented as well

6-Bonus

Students with best 3 accuracy values will get a bonus

Notes

1. You should deliver well documented code as well as a report illustrating every step in the assignment.
2. Copied assignments will be penalized. So, not delivering the assignment would be much better.
3- You should work in groups of 2

References

[1] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. (2011). Learning Word Vectors for Sentiment Analysis. The 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011).

GOOD LUCK