CS 6320: Natural Language Processing

Homework 2: Text Classification (25 points)

Due: March 2, 2020 2:30pm

This homework will expose you to scikit-learn: a Python API that is used for common NLP and Machine Learning tasks. Specifically, you will learn how to use scikit-learn to carry out feature engineering and supervised learning for sentiment classification of movie reviews. To download the package, use the following command:

pip install sklearn [-- user]

- Download and unzip the training and test corpora available on the class webpage. Datasets are simple plaintext files grouped into two folders: *pos* and *neg*. All files in the *pos* folder have a positive sentiment associated with them; and all files in the *neg* folder have a negative sentiment associated with them.
- Use the CountVectorizer and TfidfVectorizer classes provided by scikit-learn to obtain bag-of-words and tf-idf representations of the raw text respectively.
- With the feature representation as input; train the Naive Bayes and Logistic Regression classifier(s) to carry out text classification.
- Test the performance of your classifier(s) on the test set by reporting accuracy, precision, recall and F-score values for the test set.

Additionally, carry out these experiments:

- Observe the effect of using bag-of-words and tf-idf representations on the model's performance.
- Look into how stop words can be removed. Observe the effect of removing stop words on model performance.
- Observe the effect of L1 and L2 regularization v/s no regularization with Logistic Regression on model performance.

Your program should take as input the following six arguments:

```
python program.py <training-set> <test-set> <representation>
<classifier> <stop-words> <regularization>
```

where `<training-set>` represents the path to the training folder, `<test-set>` represents the path to the test folder, `representation` $\in$ {`bow, tfidf`} is a string indicating what representation to use, `classifier` $\in$ {`nbayes, regression`} is a string indicating what classifier to use, `stop-words` $\in$ {`0, 1`} indicates whether or not to use stop words, `regularization` $\in$ {`no, l1, l2`} indicates whether to use L1 or L2 regularization or neither (note that this argument is applicable only if you choose logistic regression classifier)

For example, the call `python program.py train test tfidf regression 0 l1` requires the program to train logistic regression with L1 regularization on the files present in train folder and test them on the file present in test folder. The tf-idf representation must be used without removing any stop words.

Submit the following bundled into a single zip file via eLearning:

1. Your code file(s)
2. A readme giving clear and precise instructions on how to run the code
3. A plaintext file outlining the results you obtained.

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).
2. https://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html