Spring 2018, Text Mining

Homework assignment 1.

Due Feb. 22, 12:00 pm.

In this assignment you experiment with a set of Python and NLTK functionalities, including, regular expressions, stemming, POS tagging, etc. You will also be dealing with the text data stored in different files. In addition, use this assignment to review Python Pandas data structures such as dictionaries, data frames, etc. (https://pandas-docs.github.io/pandas-docs-travis/dsintro.html)

In this assignment you will create and use vector space models. As the input corpus you will use the movie_reviews corpus from the set of NLTK corpora (https://github.com/sloria/textfeel-web/tree/master/nltk_data/corpora/movie_reviews). Each of the documents is tagged with the 'pos' and 'neg' labels. See the screenshot below:

```
In [1]: from nltk.corpus import movie_reviews

        categories = movie_reviews.categories()
        categories

Out[1]: ['neg', 'pos']
```

1. Create three document-term matrices, where each term is a word AND its POS tag:

    i. With tf scores for all the documents
    ii. With tf scores for all the documents labeled as positive
    iii. With tf scores for all the documents labeled as negative

    Try to apply a length normalization function. If you apply length normalization function, describe this function.

    For each of the three document-term matrices output the first 10 adjectives (words with the ADJ / JJ POS tags) and the first 10 adverbs (words with ADV / RB POS tags) with the highest scores.

    What can you conclude about the three text classes given the output?

    Output the size of your feature vector.

2. Create three document-term matrices using document words. For this experiment, remove capitalization by converting al words to lowercase, remove from consideration stop words.

    i. With tf*idf values for all the documents
    ii. With tf*idf values for all the documents labeled as positive
    iii. With tf*idf values for all the documents labeled as negative

Additional experiments:

- Try different length normalization functions
- Try different term forms: words, stemming

Use Pandas dataFrames to store the document-term matrix. Pick 10 documents (5 positive, 5 negative). For each document output the 10 highest tf*idf values and the corresponding words.

Describe the normalization function, if you used any. Describe what term form you use and why.

Output the size of your feature vector.

Depending on the computing power you are using for this assignment you might not have enough memory to store the information about all the terms in the input document collection. If you encounter such a situation, reduce the number of terms. For example, set up the minimal number of documents in which the terms should occur. If you use this option, specify in your answers what was the minimal number of documents in which the terms should occur to be used as features.

Minimal submission: an .ipynb notebook showing your work. I will run your notebook cell by cell.

- All the cells should run and produce results;
- All the outputs marked in red in the assignment description should be present in your notebook. Use plain text cell and comments for proper documentation
- All the questions marked in blue in the assignment description should be answered.

Grade distribution:

15 points – **Execution:** Each notebook cell must run without error or warning.

15 points – **Answers** to every questions highlighted in red in the assignment.

10 points – **Correctness** of the scoring implementation.

5 points – **Instructor's Discretion:** Is the code well written and documented? Are the responses particularly thoughtful or insightful?

If you have time and are interested in sentiment analysis, here is a link to a collection of papers that use the movie_reviews corpus. You can try to replicate any of those experiments, or, perhaps, these papers might give you ideas for your class project.

Also, you can try to run the above experiment for a different document collection that you can. For this you can either download a collection that is of interest to you, or create your own document collection.