HW1: Sentiment Analysis

March 16, 2019; due March 22, 2019 (11:59 pm)

In this homework you will create different models to generate the positive/negative sentiment classification of movie reviews.

This homework should be done **individually** without cooperation with others.

Download the homework:
git clone https://github.com/qishicpc/NLP_Homework.git

You are given the following files:

- *hw01.ipynb*: Notebook file with starter code
- *tools.py:* utility code
- train.txt: Training set to train your model
- test.txt: Test set to report your model's performance
- sample_prediction.csv: Sample file for your prediction result

Remember to leverage code in the tools.py, so you don't have to build everything from Scratch.

- *load_data(filename)*: load the input data and return sklearn.Bunch object. For basic usage of Bunch object, please refer to sklearn documentation.
- *save_prediction(arr, filename)*: save your prediction into the format required by the course

0. Install Anaconda

If you do not yet have Python and Jupyter Notebook on your laptop, use this link (https://www.anaconda.com/) to install anaconda. Anaconda is a suite that provides one stop solution for all you need for Python development environment. This site contains installation document for Windows, Mac, and Linux, choose the one that suits your operating system.

1. Feature Dictionary Vectorization

A quite unique step for NLP is to engineer the raw text input into numerical features. You will eventually implement several featurizers, just like dummy_featurize, that distinguish your models with others. However, we are not there yet. For convenience we allow you to

represent the features using a dictionary, look at the dummy_featurize code. So each data point can be translated into a dictionary. However later we have to translate this list of dictionaries into a homogeneous data structure. Therefore, you need to first implement the pipeline method in the SentimetnClassifier class. Look into the code comment for more details. To ensure your implementation works, we also provided some test code in the cell below.

2. Better featurizers

Have you finished the first step, you can run the model using the provided featurizer. See the performance is nearly as good as a donkey? No surprise! The dummy_featurize should have been named really_dummy_featurize. Now it is your turn to implement better featurizers. For this homework, you need to implement at least 3 distinguishable featurizer. Describe your features briefly in the write-up and include the accuracy of the model. No idea at all? Look at your lecture notes for inspiration. Still no clue? Why not start with Bag of words?

Note: Model performance is important but it's not the only thing we care about, your work will also be rewarded by your creativity.

3. Optional: Try different learning methods

All the work you have done so far are related to feature engineering, and the featurized data is trained using Logistic Regression. Try to use different learning methods to train the model and see if you achieve any difference in the performance. Discuss your findings in the write-up.

4. Deliverables (zip them all)
    - pdf version of your final notebook
    - Use the best model you trained, generate the prediction for test.txt, name the output file prediction.csv (Be careful: the best model in your training set might not be the best model for the test set).
    - HW1_writeup.pdf: summarize the method you used and report their performance. If you worked on the optional task, add the discussion. Add a short essay discussing the biggest challenges you encounter during this assignment and what you have learnt.