Homework2: **Topic Modeling**

You can work in pairs or individually on this homework. If working in pairs, only one of you should submit the code and result files. Also, name your partner in the README and in the text file containing your analysis. The other person should submit a text file containing the information of both partners (full names and ucsc usernames).  
  
In this homework, we will experiment with performing text normalization specific to twitter and experiment with topic models.

**Dataset**: We will run topic modeling on tweets. The dataset comes from twitter scrapes executed during the months of September-November 2016 on topics related to election, women, etc.

There is a CSV file in the location Files/Homework/Homework2/Homework2\_data.csv

**Your tasks:**

**Task1**: **Preprocessing and lexical normalisation of tweets (5 points)**

The goal here is to normalise non-standard words in English Twitter messages to their canonical forms. In this, we aim to correct things like non-standard spellings (e.g., toook for took), expand informal abbreviations (e.g., tmrw for tomorrow), and normalise phonetic substitutions (e.g., 4eva for forever). Proper nouns shall be left untouched, even if they are not in the given lexicon (e.g., Twitter), and remove stopwords. Keep in mind some of the preprocessing tasks you did in the first assignment, and see how you can build on what you have already learned. **Call it preprocess.py**

This is an open-ended task and you are expected to be creative and provide your own implementation of the task.

You may use resources such as

<http://pythonhosted.org/pyenchant/tutorial.html>

Normalisation Lexicon (Han et al., 2012)

The following paper may be a good start.

Lexical Normalisation of Short Text Messages: Makn Sens a #twitter

**Task 2**: **Compare topic models (3 points)**

Fit a topic model with number of topics varying between 3-5 (one for each algorithm given below. Have two functions in the code. **Call it topicmodel.py**

1. [LDA from Gensim](https://radimrehurek.com/gensim/models/ldamodel.html)

2. [Biterm Topic Model](https://github.com/xiaohuiyan/OnlineBTM)

**Task3. Create a main.py** that calls all the above tasks in sequence and writes top n words for a given topic model. User should be able to give the model name, number of topics and the count of top n words as an input to the main.py. The output should give top n words for each topic as per model name, K (topics) and n (top words).

**Task4**: **Qualitative Evaluation (2 points)**

Do a qualitative analysis of the topics you get by observing top n words for each model you created. Based on the analysis, select the number of topics that gives more coherent topics, say K. Based on these words, try to assign meaningful labels to the topics. Compare and contrast the LDA model to Biterm Model for your best value k.

**Submission**

1. A text file containing the top 10 words in each topic for all the 6 models you created.
2. A text file containing qualitative evaluation and your choice of K, labels that you assigned to the topics. Compare and contrast LDA to BTM for your choice of K. In your opinion, how effectively have the topics been identified?
3. The source code to finish this task (you should have the following files at least:

* preprocess.py
* topicmodel.py
* main.py

4. A README file that briefly explains the main idea and implementation of your algorithm and also the instructions to run and test your code. If there is anything that the grader should know about your code, please include it in the README.

**We should be able to run your code for all the types of algorithms and generate all the result files.**

To get full marks, you must implement the algorithms in a modular way allowing for easy addition of new features with minimal modification.

**Please submit a zipped file containing all of the above files.**

**Grading Rubric**

* 5/5: code is readable and runs, all scripts and output files are included, and explanation of results is meaningful
* 4/5: same as above, but does not easily run
* 3/5: code is provided but results analysis is superficial
* 2/5: missing components from code, still some effort shown
* 1/5: messy/incomplete code and output files, very limited analysis
* 0/5: no effort demonstrated