### Assignment on Chapter 6 in “*Natural Language Processing with Python --- Analyzing Text with the Natural Language Toolkit” by* Steven Bird, Ewan Klein, and Edward Loper

You will find Chapter 6 posted at:

<http://www.nltk.org/book/ch06.html>

On the Canvas, a file named

nltkChapter6\_template.py

is posted. You are requested to use this template in formulating your code solutions to the following exercises.

**In all your implementations for questions below: DO NOT use random.shuffle on your data.**

1. exercise 2.  Design at least 5 features and report what these features capture. Additionally, use three classifiers, namely, nltk.NaiveBayesClassifier, nltk.DecisionTreeClassifier, nltk.MaxentClassifier. Compare the performance of the three classifiers by analyzing the accuracy. Report the accuracy of each classifier built using all of the features that you designed.
2. exercise 4. To report, pick any 5 features out of the computed 30 and describe their relevance.
3. exercise 7. Design at least 5 features and report what these features capture. Report the accuracy of your classifier. Place your classifier code into the report.
4. exercise 0 (0 is a dummy number in this case). Word features can be very useful for performing document classification, since the words that appear in a document give a strong indication about what its semantic content is. However, many words occur very infrequently, and some of the most informative words in a document may never have occurred in our training data. One solution is to make use of a lexicon, which describes how different words relate to one another. Using WordNet lexicon, augment the movie review document classifier presented in Chapter 6 to use the following two features on the intersection of words appearing in a document to classify and words appearing in “word\_features”:
   1. Make a binary feature which reports “KNOWN” if the word is found in WordNet (i.e. wn.synsets is non-empty) and “UNK” if it is not found.
   2. Make a lemma name feature. Select the first synset from wn.synsets and choose the first lemma name from synset.lemma\_names as the appropriate lemma. Report “UNK” if it is not found.

Report the accuracy of your classifier: use nltk.NaiveBayesClassifier, your **test** set should contain the first 100 instances in documents defined as follows:

from nltk.corpus import movie\_reviews

documents = [(list(movie\_reviews.words(fileid)), category)

for category in movie\_reviews.categories()

for fileid in movie\_reviews.fileids(category)]

The remaining instances in documents should be part of your **training** set.

How does this accuracy compare to the accuracy of the classifier trained on the original feature set from the book? (Note that accuracy may not improve.) Why do you think you observe the behavior you observe?

1. (Extra Credit 20 points) exercise 9. Design at least 5 features and explain them. Use nltk.NaiveBayesClassifier. Report the accuracy of your classifier built using all of the features that you designed. Use show\_most\_inforamtive\_feautures(5) functionality from the classifier to inspect the individual feature performance. Which of your features seem to be most influential? Note: http://www.nltk.org/howto/corpus.html#other-corpora provides a little more information on ppattach corpora than the textbook. Section 4 of the publication posted at <https://works.bepress.com/yuliya_lierler/55/> starts by describing the dataset by Ratnaparkhi et al. (1994). This is exactly the dataset included in ppattach in NLTK.

Submit

* printed solutions to the assignment on a due date in the beginning of the class.
* file nltkChapter6\_template.py (populated with your coding solutions) via Canvas 30 minutes before the class on a due date.