CSC 5930/9010: Text Mining
Lab 3

Dr. Paula Matuszek
Paula.Matuszek@villanova.edu
Paula.Matuszek@gmail.com
(610) 647-9789
Goals

• Goals for this lab are:
  • Use NLTK Naive Bayes Classifier to classify movie reviews based on word frequency and evaluate the results
    • Document Classification section of Ch 6.
  • Use Decision Tree Classifier to classify POS
    • Part of Speech Tagging section of Ch 6.
  • Do exercise 6.2 from Ch 6.
Classifying Documents

• Same set of steps
• Create a feature set.
  • Get a frequency distribution of words in the corpus
  • Pick the 2500 most common (more or less)
  • Create a feature set of “word is present”, true or false.
• Classify into positive and negative reviews
• Evaluate results
Movie Reviews

- The NLTK corpus includes a set of 2000 movie reviews, classified into directories of positive and negative. (From Cornell, released in 2004).

- NLTK.corpus includes methods to get the categories of reviews, the fileids in each category and the words in each fileid.
Creating the feature set

• Too many terms for us! (almost 40K)

• Get a frequency count and take the most frequent.

• For each of the words in that list, for each document, create a feature:
  • 'contains(like)': True,

• Each document is a two-item list: dictionary of features, category

• The featureset is a list of these documents
Doing this for your documents

- Decide your features and your categories!
- Input your documents and their categories.
- Categories could be:
  - the file they are in (like names)
  - the directory they are in (like movie reviews)
  - a tag in the document itself (first token, for instance)
- Build feature list for each document: a dictionary of label-value pairs
  - BOW, length, diversity, number of words, etc, etc.
Part of Speech Classifier

- By yourself or in small teams work through the Part of Speech section of Chapter 6 to build a decision tree and create the confusion matrix for it.
Exercise 6.2

- ☉ Using any of the three classifiers described in this chapter, and any features you can think of, build the best name gender classifier you can. Begin by splitting the Names Corpus into three subsets: 500 words for the test set, 500 words for the dev-test set, and the remaining 6900 words for the training set. Then, starting with the example name gender classifier, make incremental improvements. Use the dev-test set to check your progress. Once you are satisfied with your classifier, check its final performance on the test set. How does the performance on the test set compare to the performance on the dev-test set? Is this what you'd expect?