NLP Final Project
Fall 2014, Due Friday, December 12

For the final project, everyone is required to do some sentiment classification and then choose one of the other three types of projects: annotation, sentiment classification experiments and implementation. You may also propose your own project and you may work in groups.

Required Part 0. Sentiment Classification

For this part, you are to do at least one experiment in the NLTK on the movie review data in which you compare the baseline performance of using just word features with some modified or additional features that you implement.

You will need to write a Python feature function to generate the features that you choose. One idea is to vary the representation of the subjectivity lexicon features or to use positive and negative emotion words from LIWC. Another idea would be to extend the representation of the negation features up to the next punctuation. Another idea would be to run the POS tagger on the text and include as features the counts of the POS tags of type verb (tag starts with V), noun (starts with N), adjective (starts with J) and adverb (starts with R).

Note that you will first want to develop your feature function until it works. Then to do the experiment, define a random training and test set; define the regular word features, run the classifier and get accuracy; and then define your new features, run the classifier and get accuracy. The important thing is the two classifier runs are made on the same training and test sets so that you can compare the accuracy. Note that two runs are the absolute minimum, and it is preferred that you make a series of related feature comparison runs.

Include a description of your experiments in the final project report, together with the feature function code that you wrote and the accuracies of the two classifications.

Now choose one of the following 3 options.

Option I. Annotation and Analysis of Data

For this task, you will annotate data in a corpus and compare your annotations with other annotators on the emotion data from Jasy Liew.

To complete this project, write a reflection report that describes what you did. First, write a reflection on the annotation process. Include in your report examples of tweets and emotion labels that were straightforward and easy to decide, and also examples of those that were either hard for you to decide or hard for the group to come to agreement. Include the inter-annotator agreements for your group over the process of the semester. For the last part of the reflection, look at some of the emotion clue words or phrases and speculate on what types of features would be good to characterize the tweets for a ML classifier to label the tweets with the emotions. Do you see differences in clues for emotion than we saw in lab for sentiment?

Option 2. Processing and Classification of Sentiment or other Data

For this task, you should choose to work on classification of a data set. If you do enough experiments on this task, then you do not also have to do the required part 0 of the Final Project, as that will be replaced by the extra experiments that you do in this option.

Based on the data and the task, decide what level of NLP processing is desirable and carry out any NLP processing needed, e.g. you may want to run special purpose Tweet POS tagging. Read the data into the NLTK (either by reading the file, or using a PlainCorpusReader) and write Python/NLTK that defines features.

Produce the features in the notation of the NLTK and use one of their classifiers to train and test a classifier on the data. If you use an NLTK classifier, you should also implement cross-validation. Or you can choose to produce the features as a csv file and use Weka or Sci-Kit Learn to train and test a classifier, using cross-validation scores.

Available Data:

- Twitter data annotated with general sentiment from the SemEval shared task in 2014.
- Twitter data annotated with general sentiment from Sentiment 140 circa 2010.
- Email data separated into Spam and Ham directories for spam detection
    - (See http://blog.nerdery.com/2013/03/playing-in-the-sandbox-building-a-spam-detector-with-python/ for a description of this data and using it to make a spam detection classifier.)
- Rotten Tomatoes movie review phrase data from Kaggle.

Available Resources to Augment text processing in the NLTK:

- Twitter processing from the ARK (include Tweet Motif tokenization).
- Lexical resources: the Subjectivity lexicon from Wiebe, the LIWC dictionary from Pennebaker and the ANEW dictionaries from Florida, the NRC twitter sentiment lexicon.
- Stanford POS tagger, named entity recognizer and parser(s).

Both text and tweet processing will be discussed further in lab for week 13.

To complete this project, carry out at least several experiments where you use two different sets of features and compare the results. For example, you may take the unigram word features as a baseline and see if the features you designed improve the accuracy of the classification. Write a report that describes the data processing, the features and the classification experiment(s). As one of your experiments, you may instead compare results from different classifier algorithms in Weka or Sci-Kit Learn.

Option 3. Programming Projects

Write a program to process text and discover lexical chains. We will work from the paper:

Barzilay and Elhadad, “Using Lexical Chains for Text Summarization”, 1999. We will identify a subset of their algorithm that is reasonable to implement in NLTK using WordNet. More details will be forthcoming.

Write a Python program with a window interface that allows a user to specify a file or directory of files to process. The program should use the Stanford Named Entity Recognizer to process the text. There is a Python interface to the Stanford NER by Dat Hoang at https://github.com/dat/pyner. After the NER processes the text, the python program should make and display most frequent words, pruned by a stop word list, and most frequent named entities for the categories Person, Organization and Location. More details are found in the Programming Projects document.

The programs should be well-documented in a report that is handed in with the code and with test results.

What to Hand In

If you are working in a group, you should choose a task for each person. If you do annotation, hand in the annotation data. Every group should hand in a report with the description of all that you did and the discussion of the results. As usual, submit these documents to the Blackboard system by the end of the day on the due date.