NLP Homework 2
Due November 3, 2011

For this homework, choose one of the following tasks. In either task, you may choose to work in a group of two and just do more rounds of development and description of results. If you want to work in a group with more than two people, you must propose additional work to me.

1. Chunk Parsing for Base Noun Phrases

What you will learn: More experience with regular expressions, syntactic knowledge of noun phrases, process of developing rules to improve accuracy of chunking.

For this task, you will do the analysis for a base noun phrase chunker for news style text. The noun chunker will use the NLTK functions to make a regular expression grammar based on POS tags, to make the chunker and to test it on text.

Recall that we had an simple example of a regular expression grammar:

```
NPchunkgrammar = r"""
  NP: {<DT|PP\$>?<JJ>*<NN>}       # determiner/possessive, adjectives, nouns
      {<NNP>+}                    # sequences of proper nouns
"""
```

To obtain this grammar, we may observer examples of noun phrases from the Wall Street Journal corpus:

another/DT sharp/JJ dive/NN
Pierre/NNP Vinken/NNP

For this assignment, we are going to extend this chunk grammar to do additional forms of noun phrases.

To explain the main idea of the assignment, first look at the first 50 sentences of the annotated Wall Street Journal and select any base noun phrases that will not match either of the above rules. Remember that base noun phrases are annotated with NP, but do not include any other noun phrases.

To look at the first sentence, we run: t = treebank.parsed_sents('wsj_0001.mrg')[0] and get:

```
(S
  (NP-SBJ
    (NP (NNP Pierre) (NNP Vinken))
    (, ,)
    (ADJP (NP (CD 61) (NNS years)) (JJ old))
    (, ,))
  (VP
    (MD will)
    (VP
      (VB join)
```

(NP (DT the) (NN board))
(PP-CLR (IN as) (NP (DT a) (JJ nonexecutive) (NN director)))
(NP-TMP (NNP Nov.) (CD 29))))
(. .))

To start with, we disregard the temporal noun phrase (NP-TMP) and observe that the other base NPs are:

(NP (NNP Pierre) (NNP Vinken))
(NP (CD 61) (NNS years))
(NP (DT the) (NN board))
(NP (DT a) (JJ nonexecutive) (NN director))

All these noun phrases match one of the two patterns except for the phrase "61 years". So we would collect that as a noun phrase that needs additional patterns.

Developing a Chunker

For our development, we are going to continue using the Penn Treebank corpus, but with a different format for chunk annotation. For this corpus, we can build regular expression chunk parsers with NLTK and then use its chunk scoring to evaluate how well our parser does so far. We can examine the errors and omissions of the chunking to determine additional regular expressions.

The use of the NLTK Regular Expression parser (RegexpParser) and the chunkscore function to develop a chunk parser are detailed in the python file called homework2examples.py. These examples will be discussed in lab.

For your homework assignment, continue the development of the chunk parser for 5-10 more rounds, improving the recall and precision, if possible. You may continue to use the first 5 files of the Penn Treebank corpus for your development and testing.

What to Hand In

Hand in a document that describes the final result of your chunk parser: the regular expression grammar and the results of chunkscore. Although you should NOT give every detail of your development process, you should write a brief discussion of any interesting aspects that occurred as you developed the chunk parser. These should include the following discussion questions:

Did you have difficulties with the ordering of the rules? For example, did an earlier rule capture an (incorrect) noun phrase that prevented a later rule from taking effect?

Did you introduce any rules that did not improve both precision and recall? For example, did you introduce a rule that gave greater recall, but reduced precision?

Were there any noun phrases that caused you particular problems in writing the rules? For example, you may observe chunked noun phrases where you believe that the human annotation for the gold standard is in error. Or you may write rules for phrases that are so rare that it hardly affects the score.

2. Content Extraction from a Twitter Corpus

What you will learn: More experience with regular expressions, process of developing rules to extract content of interest in a corpus (related to content analysis of text).

This task is based on a research project that worked on a collection of tweets collected about the topic of health care reform. From this collection of over 1 million tweets, a random sample of approximately 5,000 was taken to analyze for how people participate in the political process through the twitter medium. Of all the tweets, one of the types of tweets was what we termed "Calls to Action". In these tweets, the user would call on other tweeters or congressmen to do particular actions. For example, they may ask others to pass the word "pls share" or "RT this", they may call on congressman to "pass health care" or "pass #hcr", or call for other participatory actions, "watch this video" or "please call". Although this project also noted when people just tried to contact or give a message to a congressman, those tweets were removed as "calls to action" for this class project, so that we don't have to deal with a list of congressional names and tweet ids. (Although in the future, the research project will probably change to identifying the calls to action tweets as a classification task, the development of this idea and the labeling of the initial gold standard set was entirely done with regular expressions.)

For the research project, the entire tweet was labeled with the "call to action" , even when particular phrases were identified. For the class project, we will develop regular expression patterns that find particular call to action phrases, such as those mentioned above, and then evaluate according to whether the entire tweet was labeled as a call to action.

This type of content extraction task has a similar development cycle as the noun chunking. We look at examples of phrases that indicate calls to action, we develop one or more regular expressions to match those phrases, we apply the regular expression extractor to the tweet text and then evaluate by comparing the call to action tweets identified by the regular expressions with the gold standard tweets. We continue this cycle by examining the missing and incorrect examples to further develop the regular expressions.

To assist in this process, there are some Python functions written to read the data from a file, apply the regular expressions to the data and to evaluate the results. These functions will be detailed in the lab.

Unlike the chunk parser, this task is looking at patterns involving words, initially, and one of the goals of the extraction development will be to write as general a rules as possible.
As this is a new homework task, there will undoubtedly be changes as the task goes along, perhaps corrections to the data or functions. It may also be possible to use the POS tagged tweets to look at patterns that involve POS tags on preceding text.

What to Hand In

Hand in a document that describes the final result of your regular expression extractor: the regular expression rules and the results of the scoring. Although you should NOT give every detail of your development process, you should write a brief discussion of any interesting aspects that occurred as you developed the regular expressions. Further data and discussion questions may be forthcoming.

Submit your report to the iLMS system assignment dropbox at the end of the day on the due date.