Week 4 Lab Exercise

September 28, 2017

Pan Chen Week 4 Lab

0.1 Lab Exercise

- Run the regexp tokenizer with the regular pattern on the sentence "Mr. Black and Mrs. Brown attended the lecture by Dr. Gray, but Gov. White wasn't there."

a. Design and add a line to the pattern of this tokenizer so that titles like "Mr." are tokenized as having the dot inside the token. Test and add some other titles to your list of titles.

b. Design and add the pattern of this tokenizer so that words with a single apostrophe, such as "wasn't" are taken as a single token.

In [10]:
```
#Run the regexp tokenizer with the regular pattern on
#the sentence Mr. Black and Mrs. Brown attended the lecture by Dr. Gray, but Gov. Whi

#import the package
import nltk



tmp = "Mr. Black and Mrs. Brown attended the lecture by Dr. Gray, but Gov. White wasn

'''1. Design and add a line to the pattern of this tokenizer so that titles like Mr.
are tokenized as having the dot inside the token. Test and add some other titles to y
'''
pattern = r''' (?x)          # set flag to allow verbose regexps
         \w+.\ #to allow dot
        | (?:[A-Z]\.)+    # abbreviations, e.g. U.S.A.
        | \$?\d+(?:\.\d+)?%?    # currency and percentages, $12.40, 50%
        | \w+(?:-\w+)*  # words with internal hyphens
        | \.\.\.        # ellipsis
        | [][.,;?():-_%#]    # separate tokens
        '''
print(nltk.regexp_tokenize(tmp, pattern),"\n")

'''b. Design and add the pattern of this tokenizer so that words with a single
apostrophe, such as wasn't are taken as a single token.
```

```
        '''

        pattern2 = r''' (?x)         # set flag to allow verbose regexps
                (?:[A-Z]\.)+    # abbreviations, e.g. U.S.A.
                | \$?\d+(?:\.\d+)?%?    # currency and percentages, $12.40, 50%
                | \w+.\ #to allow dot
                | \w+(?:\w+)*  # words with internal apostrophe
                | \w+(?:-\w+)*  # words with internal hyphens
                | \.\.\.        # ellipsis
                | [][.,;?():-_%#]    # separate tokens

                '''

        print(nltk.regexp_tokenize(tmp, pattern2))
```

```
['Mr. ', 'Black ', 'and ', 'Mrs. ', 'Brown ', 'attended ', 'the ', 'lecture ', 'by ', 'Dr. ',
```

```
['Mr. ', 'Black ', 'and ', 'Mrs. ', 'Brown ', 'attended ', 'the ', 'lecture ', 'by ', 'Dr. ',
```

In [ ]: