NLP Lab 3

Dixitha Kasturi
dkasturi@syr.edu

Topic: Regular expressions.

I did both cases that were asked. I included the regular expression in the predefined pattern and twitter patterns. I separately showed them in the end for easy understanding. Firstly,I processed both patterns on the sample text that was given :

text = “Mr. Black and Mrs. Brown attended the lecture by Dr. Gray, but Gov. White wasn’t there.”

tweet1 = "@natalieohayre I agree #hc09 needs reform- but not by crooked politicians who r clueless about healthcare! #tcot #fishy NO GOV'T TAKEOVER!"

tweet2 = "To Sen. Roland Burris: Affordable, quality health insurance can't wait http://bit.ly/j63je #hc09 #IL #60660"

tweet3 = "RT @karoli: RT @Seriou: .@whitehouse I will stand w/ Obama on #healthcare, I trust him. #p2 #tlot"

a)

```
x = r'''(?x)
[A-Z]+\.*\w+\.
| [A-Za-z]*\'[t]'''
print(nltk.regexp_tokenize(text, x))
```

Answer: ['Mr.', 'Mrs.', 'Dr.', 'Gov.', "wasn't"]

b)

```
z= r''' (?x)
w/+
|[A-Z a-z]*\'[t]
|[A-Z]+\.*\w+\. '''

print(nltk.regexp_tokenize(tweet1,z))
print(nltk.regexp_tokenize(tweet2,z))
print(nltk.regexp_tokenize(tweet3,z))
```

```
[]
['Sen.', "can't"]
['w/']
```

c) on own sample:

```
sample = '''MS. Dixitha Kasturi is an aspiring datascientist,She's a dog lover.
w/ it wasn't unlikely that she doesn't like cakes. Mr. Hayd is one of her favourite'''

print(nltk.regexp_tokenize(sample,x))
print(nltk.regexp_tokenize(sample,z))
```

```
['MS.', "wasn't", "doesn't", 'Mr.']
['MS.', 'w/', "wasn't", "doesn't", 'Mr.']
```

The same patterns are added as the first lines in the pattern and tweetpattern variable in the python notebook.

```
pattern = r''' (?x) # set flag to allow verbose regexps
 (?:[A-Z]\.)+ # abbreviations, e.g. U.S.A.
 |[A-Za-z]*\'[t] #" to take ' separated words as singke token'"
 |[A-Z]+\.*\w+\. # for words ending with .
 | \$?\d+(?:\.\d+)?%? # currency and percentages, $12.40, 50%
 | \w+(?:-\w+)* # words with internal hyphens
 | \.\.\. # ellipsis
 | [][.,;"'?():-_%#'] # separate tokens
 | [A-Z]+\.*\w+\. #for titles ending with . like Mr
 | [\w\.-]+'[\w\.-]+ '''
```

```
['That', 'book', 'is', 'interesting', '.']
['That', 'U.S.A.', 'poster-print', 'costs', '$12.40', ',', 'but', 'with', '10%', 'off', '.']
['That', 'U.S.A.', 'poster-print', 'costs', '$', '12.40', ',', 'but', 'with', '10', '%', 'off', '.']
```

```
tweetPattern = r''' (?x) # set flag to allow verbose regexps
 (?:https?://|www)\S+ # simple URLs
 | w/+ #for token 'w/'
 |[A-Za-z]*\'[t] #" to take ' separated words as singke token'"
 |[A-Z]+\.*\w+\. # for words ending with .
 | (?::-\)|;-\)) # small list of emoticons
 | &(?:amp|lt|gt|quot); # XML or HTML entity
 | \#\w+ # hashtags
 | @\w+ # mentions
 | \d+:\d+ # timelike pattern
 | \d+\.\d+ # number with a decimal
 | (?:\d+,)+?\d{3}(?=(?:[^,]|$)) # number with a comma
 | (?:[A-Z]\.)+ # simple abbreviations
 | (?:--+) # multiple dashes
 | \w+(?:-\w+)* # words with internal hyphens or apostrophes
 | ['\".?!,:;/]+ # special characters
 '''
```

```
Tweetpattern tweet1----- ['@natalieohayre', 'I', 'agree', '#hc09', 'needs', 'reform', 'but', 'not', 'by', 'crooked', 'politicians', 'who', 'r', 'clueless', 'about', 'healthcare', '!', '#tcot', '#fishy', 'NO', 'GOV', "'", 'T', 'TAKEOVER', '!']
```

```
Tweetpattern tweet2----- ['To', 'Sen.', 'Roland', 'Burris', ':', 'Affordable', ',', 'quality', 'health', 'insurance', "can't", 'wait', 'http://bit.ly/j63je', '#hc09', '#IL', '#60660']
```

```
Tweetpattern tweet3----- ['RT', '@karoli', ':', 'RT', '@Seriou', ':', '.', '@whitehouse', 'I', 'will', 'stand', 'w/', 'Obama', 'on', '#healthcare', ',', 'I', 'trust', 'him', '.', '#p2', '#tlot']
```

```
Tweettokenizer  ['@natalieohayre', 'I', 'agree', '#hc09', 'needs', 'reform', '-', 'but', 'not', 'by', 'crooked', 'politicians', 'who', 'r', 'clueless', 'about', 'healthcare', '!', '#tcot', '#fishy', 'NO', "GOV'T", 'TAKEOVER', '!']
```

Report:

There are different ways/ regular expressions for finding the same pattern, I only mentioned 1 type here. I would like to explore other patterns for the same questions.