Applied Text Mining in Python

Basic NLP tasks with NLTK

An Introduction to NLTK

- NLTK: Natural Language Toolkit
- Open source library in Python
- Has support for most NLP tasks
- Also provides access to numerous text corpora

Let's set it up first!

```
>>> import nltk
```

- **Let's get some text corpora**

```
>>> nltk.download()
```

```
>>> from nltk.book import *
*** Introductory Examples for the NLTK Book ***
Loading text1, ..., text9 and sent1, ..., sent9
Type the name of the text or sentence to view it.
Type: 'texts()' or 'sents()' to list the materials.
text1: Moby Dick by Herman Melville 1851
text2: Sense and Sensibility by Jane Austen 1811
text3: The Book of Genesis
text4: Inaugural Address Corpus
text5: Chat Corpus
text6: Monty Python and the Holy Grail
text7: Wall Street Journal
text8: Personals Corpus
text9: The Man Who Was Thursday by G . K . Chesterton 1908
```

Let's set it up first! (2)

```
>>> text1
<Text: Moby Dick by Herman Melville 1851>


>>> sents()
sent1: Call me Ishmael .
sent2: The family of Dashwood had long been settled in Sussex .
sent3: In the beginning God created the heaven and the earth .
sent4: Fellow - Citizens of the Senate and of the House of Representatives :
sent5: I have a problem with people PMing me to lol JOIN
sent6: SCENE 1 : [ wind ] [ clop clop clop ] KING ARTHUR : Whoa there !
sent7: Pierre Vinken , 61 years old , will join the board as a nonexecutive director Nov.
29 .
sent8: 25 SEXY MALE , seeks attrac older single lady , for discreet encounters .
sent9: THE suburb of Saffron Park lay on the sunset side of London , as red and ragged as
a cloud of sunset .


>>> sent1
['Call', 'me', 'Ishmael', '.']
```

Simple NLP tasks (1)

- **Counting vocabulary of words**

```
>>> text7
<Text: Wall Street Journal>
>>> sent7
['Pierre', 'Vinken', ',', '61', 'years', 'old', ',', 'will', 'join', 'the',
'board', 'as', 'a', 'nonexecutive', 'director', 'Nov.', '29', '.']
>>> len(sent7)
18
>>> len(text7)
100676
>>> len(set(text7))
12408
```

Simple NLP tasks (2)

```
>>> list(set(text7))[:10]
[u'Mortimer', u'foul', u'Heights', u'four', u'spiders', u'railing', u'centimeter',
u'Until', u'payoff', u'Germany-based']
```

- **Frequency of words**

```
>>> dist = FreqDist(text7)
>>> len(dist)
12408
>>> vocab1 = dist.keys()
>>> list(vocab1)[:10]
[u'Mortimer', u'foul', u'Heights', u'four', u'spiders', u'railing', u'centimeter',
u'Until', u'payoff', u'Germany-based']
>>> dist[u'four']
20
>>> freqwords = [w for w in vocab1 if len(w) > 5 and dist[w] > 100]
>>> freqwords
[u'million', u'shares', u'market', u'president', u'trading', u'billion', u'company',
u'program', u'because']
```

Normalization and Stemming

- **Different forms of the same "word"**

```
>>> input1 = "List listed lists listing listings"
>>> words1 = input1.lower().split(' ')
>>> words1
['list', 'listed', 'lists', 'listing', 'listings']

>>> porter = nltk.PorterStemmer()
>>> [porter.stem(t) for t in words1]
[u'list', u'list', u'list', u'list', u'list']
```

Lemmatization

```
>>> udhr = nltk.corpus.udhr.words('English-Latin1')
>>> udhr[:20]
['Universal', 'Declaration', 'of', 'Human', 'Rights', 'Preamble', 'Whereas',
'recognition', 'of', 'the', 'inherent', 'dignity', 'and', 'of', 'the', 'equal', 'and',
'inalienable', 'rights', 'of']

>>> [porter.stem(t) for t in udhr[:20]]
[u'Univers', u'Declar', u'of', u'Human', u'Right', u'Preambl', u'Wherea', u'recognit',
u'of', u'the', u'inher', u'digniti', u'and', u'of', u'the', u'equal', u'and',
u'inalien', u'right', u'of']
```

- **Lemmatization: Stemming, but resulting stems are all valid words**

```
>>> WNlemma = nltk.WordNetLemmatizer()
>>> [WNlemma.lemmatize(t) for t in udhr[:20]]
['Universal', 'Declaration', 'of', 'Human', 'Rights', 'Preamble', 'Whereas',
'recognition', 'of', 'the', 'inherent', 'dignity', 'and', 'of', 'the', 'equal', 'and',
'inalienable', u'right', 'of']
```

Tokenization

- **Recall splitting a sentence into words / tokens**

```
>>> text11 = "Children shouldn't drink a sugary drink before bed."
>>> text11.split(' ')
['Children', "shouldn't", 'drink', 'a', 'sugary', 'drink', 'before',
'bed.’]
```

- **NLTK has an in-built tokenizer**

```
>>> nltk.word_tokenize(text11)
['Children', 'should', "n't", 'drink', 'a', 'sugary', 'drink', 'before',
'bed', '.']
```

Sentence Splitting

- **How would you split sentences from a long text string?**

```
>>> text12 = "This is the first sentence. A gallon of milk in the U.S. costs
$2.99. Is this the third sentence? Yes, it is!"
```

- **NLTK has an in-built sentence splitter too!**

```
>>> sentences = nltk.sent_tokenize(text12)
>>> len(sentences)
4
>>> sentences
['This is the first sentence.', 'A gallon of milk in the U.S. costs
$2.99.', 'Is this the third sentence?', 'Yes, it is!']
```

Take Home Concepts

- NLTK is a widely used toolkit for text and natural language processing
- NLTK gives access to many corpora and handy tools
- Sentence splitting, tokenization, and lemmatization are important, and non-trivial, pre-processing tasks