homework2

September 8, 2022

1 Question 1

2 Question 2

This section imports the toolkit and initializes the necessary dependencies.

[2]:
```
import nltk
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('punkt')
nltk.download('omw-1.4')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\yucjo\AppData\Roaming\nltk_data…
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\yucjo\AppData\Roaming\nltk_data…
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\yucjo\AppData\Roaming\nltk_data…
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package omw-1.4 to
[nltk_data]     C:\Users\yucjo\AppData\Roaming\nltk_data…
[nltk_data]   Package omw-1.4 is already up-to-date!
```

[2]:
```
True
```

3 Question 3

- I learned that the API is outdated, as text1.tokens is a list, not a method.
- I learned that the tokens list is generated when the text object is created

[3]:
```
from nltk.book import text1
text1.tokens[0:20]
```

```
*** 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
```

[3]:
```
['[',
 'Moby',
 'Dick',
 'by',
 'Herman',
 'Melville',
 '1851',
 ']',
 'ETYMOLOGY',
 '.',
 '(',
 'Supplied',
 'by',
 'a',
 'Late',
 'Consumptive',
 'Usher',
 'to',
 'a',
 'Grammar']
```

4 Question 4

This section finds five instances of the word 'sea' and prints it in context

[10]:
```
text1.concordance('sea', lines=5)
```

```
Displaying 5 of 455 matches:
 shall slay the dragon that is in the sea ." -- ISAIAH " And what thing soever
 S PLUTARCH ' S MORALS . " The Indian Sea breedeth the most and the biggest fis
cely had we proceeded two days on the sea , when about sunrise a great many Wha
many Whales and other monsters of the sea , appeared . Among the former , one w
 waves on all sides , and beating the sea before him into a foam ." -- TOOKE '
```

5 Question 5

- This function is simply an abstraction on top of Python's existing count method.
- It uses the built in count method on object's "tokens" list.

[14]:
```
print(text1.count('sea'))
print('sea sea sea sea lol sea'.count('sea'))
```

```
433
5
```

6 Question 6

This section prints the first 10 elements of a list of tokens of an excerpt of the wikipedia page on NLP

[11]:
```
from nltk import word_tokenize
raw_text = '''Since the so-called "statistical revolution"[15][16] in the late␣
 ↪1980s and mid-1990s, much natural language processing research has relied␣
 ↪heavily on machine learning. The machine-learning paradigm calls instead for␣
 ↪using statistical inference to automatically learn such rules through the␣
 ↪analysis of large corpora (the plural form of corpus, is a set of documents,␣
 ↪possibly with human or computer annotations) of typical real-world examples.
Many different classes of machine-learning algorithms have been applied to␣
 ↪natural-language-processing tasks. These algorithms take as input a large␣
 ↪set of "features" that are generated from the input data. Increasingly,␣
 ↪however, research has focused on statistical models, which make soft,␣
 ↪probabilistic decisions based on attaching real-valued weights to each input␣
 ↪feature (complex-valued embeddings,[17] and neural networks in general have␣
 ↪also been proposed, for e.g. speech[18]). '''

tokens = word_tokenize(raw_text)
tokens[0:10]
```

[11]:
```
['Since',
 'the',
 'so-called',
 '``',
 'statistical',
 'revolution',
 "''",
 '[',
 '15',
 ']']
```

7 Question 7

This section prints the sentence tokenization of the raw_text variable

[12]:
```
from nltk import sent_tokenize
sent_tokenize(raw_text)
```

[12]:
```
['Since the so-called "statistical revolution"[15][16] in the late 1980s and
mid-1990s, much natural language processing research has relied heavily on
machine learning.',
 'The machine-learning paradigm calls instead for using statistical inference to
automatically learn such rules through the analysis of large corpora (the plural
form of corpus, is a set of documents, possibly with human or computer
annotations) of typical real-world examples.',
 'Many different classes of machine-learning algorithms have been applied to
natural-language-processing tasks.',
 'These algorithms take as input a large set of "features" that are generated
from the input data.',
 'Increasingly, however, research has focused on statistical models, which make
soft, probabilistic decisions based on attaching real-valued weights to each
input feature (complex-valued embeddings,[17] and neural networks in general
have also been proposed, for e.g.',
 'speech[18]).']
```

8 Question 8

This section stems all the words in the tokens list.

[16]:
```
from nltk.stem.porter import PorterStemmer
stemmer = PorterStemmer()
[stemmer.stem(word) for word in tokens]
```

[16]:
```
['sinc',
 'the',
 'so-cal',
 '``',
 'statist',
 'revolut',
 "''",
 '[',
 '15',
 ']',
 '[',
 '16',
 ']',
 'in',
 'the',
 'late',
 '1980',
 'and',
 'mid-1990',
 ',',
 'much',
 'natur',
```

```
'languag',
'process',
'research',
'ha',
'reli',
'heavili',
'on',
'machin',
'learn',
'.',
'the',
'machine-learn',
'paradigm',
'call',
'instead',
'for',
'use',
'statist',
'infer',
'to',
'automat',
'learn',
'such',
'rule',
'through',
'the',
'analysi',
'of',
'larg',
'corpora',
'(',
'the',
'plural',
'form',
'of',
'corpu',
',',
'is',
'a',
'set',
'of',
'document',
',',
'possibl',
'with',
'human',
'or',
```

```
'comput',
'annot',
')',
'of',
'typic',
'real-world',
'exampl',
'.',
'mani',
'differ',
'class',
'of',
'machine-learn',
'algorithm',
'have',
'been',
'appli',
'to',
'natural-language-process',
'task',
'.',
'these',
'algorithm',
'take',
'as',
'input',
'a',
'larg',
'set',
'of',
'``',
'featur',
"''",
'that',
'are',
'gener',
'from',
'the',
'input',
'data',
'.',
'increasingli',
',',
'howev',
',',
'research',
'ha',
```

```
'focus',
'on',
'statist',
'model',
',',
'which',
'make',
'soft',
',',
'probabilist',
'decis',
'base',
'on',
'attach',
'real-valu',
'weight',
'to',
'each',
'input',
'featur',
'(',
'complex-valu',
'embed',
',',
'[',
'17',
']',
'and',
'neural',
'network',
'in',
'gener',
'have',
'also',
'been',
'propos',
',',
'for',
'e.g',
'.',
'speech',
'[',
'18',
']',
')',
'.']
```

9 Question 9

- create-created
- enemi-enemy
- ani-any
- mani-many
- realiz-realize

[17]:
```
from nltk.stem import WordNetLemmatizer
wnl = WordNetLemmatizer()
[wnl.lemmatize(t) for t in tokens]
```

[17]:
```
['Since',
 'the',
 'so-called',
 '``',
 'statistical',
 'revolution',
 "''",
 '[',
 '15',
 ']',
 '[',
 '16',
 ']',
 'in',
 'the',
 'late',
 '1980s',
 'and',
 'mid-1990s',
 ',',
 'much',
 'natural',
 'language',
 'processing',
 'research',
 'ha',
 'relied',
 'heavily',
 'on',
 'machine',
 'learning',
 '.',
 'The',
 'machine-learning',
 'paradigm',
 'call',
```

```
'instead',
'for',
'using',
'statistical',
'inference',
'to',
'automatically',
'learn',
'such',
'rule',
'through',
'the',
'analysis',
'of',
'large',
'corpus',
'(',
'the',
'plural',
'form',
'of',
'corpus',
',',
'is',
'a',
'set',
'of',
'document',
',',
'possibly',
'with',
'human',
'or',
'computer',
'annotation',
')',
'of',
'typical',
'real-world',
'example',
'.',
'Many',
'different',
'class',
'of',
'machine-learning',
'algorithm',
```

```
'have',
'been',
'applied',
'to',
'natural-language-processing',
'task',
'.',
'These',
'algorithm',
'take',
'a',
'input',
'a',
'large',
'set',
'of',
'``',
'feature',
"''",
'that',
'are',
'generated',
'from',
'the',
'input',
'data',
'.',
'Increasingly',
',',
'however',
',',
'research',
'ha',
'focused',
'on',
'statistical',
'model',
',',
'which',
'make',
'soft',
',',
'probabilistic',
'decision',
'based',
'on',
'attaching',
```

```
'real-valued',
'weight',
'to',
'each',
'input',
'feature',
'(',
'complex-valued',
'embeddings',
',',
'[',
'17',
']',
'and',
'neural',
'network',
'in',
'general',
'have',
'also',
'been',
'proposed',
',',
'for',
'e.g',
'.',
'speech',
'[',
'18',
']',
')',
'.']
```

10 Question 10

The NLTK library has an incredibly robust array of functions for operating on and parsing langu