CSCI-8450

2018 Spring

Yuchen Wang

1. Wh-words in English are used in questions, relative clauses and exclamations. Consider the set of wh-words to consist exactly of the following members: *what, where, when, which, who, whom, whose, why.* 
   1. Report how many wh-words occurred in text2.
   2. Repeat the exercise for text7. Report how many wh-words occurred in text 7.

Code:

**def** exercise1():  
   
 size2 = len([w **for** w **in** text2 **if** re.findall(**r'what|where|when|which|who|whom|whose|why'**,w)])  
 print(**"text2: "**, size2)  
  
 size7 = len([w **for** w **in** text7 **if** re.findall(**r'what|where|when|which|who|whom|whose|why'**,w)])  
 print(**"text7: "**, size7)

Result:

Exercise 1

text2: 1854

text7: 609

1. ◑ Readability measures are used to score the reading difficulty of a text, for the purposes of selecting texts of appropriate difficulty for language learners. Let us define μw to be the average number of letters per word, and μs to be the average number of words per sentence, in a given text. The Automated Readability Index (ARI) of the text is defined to be: 4.71 μw + 0.5 μs - 21.43. Compute the ARI score for various sections of the Brown Corpus, including section f (lore) and j (learned). Make use of the fact that nltk.corpus.brown.words() produces a sequence of words, while nltk.corpus.brown.sents() produces a sequence of sentences

Code:

**def** average\_num\_words\_per\_sentence(text, category):  
  
 sent\_num = len(text.sents(categories=category))  
 word\_num = len(text.words(categories=category))  
 **return** word\_num / sent\_num  
  
**def** average\_num\_letters(text, category):  
  
 word\_num = len(text.words(categories=category))  
 smash\_text = **''**.join(text.words(categories=category))  
 letters\_len = len(smash\_text)  
  
 **return** letters\_len / word\_num  
  
**def** ari(text, category):  
  
 uw = average\_num\_letters(text, category)  
 us = average\_num\_words\_per\_sentence(text, category)  
 ari = (4.71 \* uw ) + ( 0.5 \* us ) - 21.43  
 **return** ari  
  
**def** exercise29():  
  
 **for** category **in** brown.categories():  
 print(category + **': '** + str(ari(brown, category)))

Result:

Exercise 29

adventure: 4.0841684990890705

belles\_lettres: 10.987652885621749

editorial: 9.471025332953673

fiction: 4.9104735321302115

government: 12.08430349501021

hobbies: 8.922356393630267

humor: 7.887805248319808

learned: 11.926007043317348

lore: 10.254756197101155

mystery: 3.8335518942055167

news: 10.176684595052684

religion: 10.203109907301261

reviews: 10.769699888473433

romance: 4.34922419804213

science\_fiction: 4.978058336905399

1. exercise 30. In this question, consider SimpleText for reporting your results.

◑ Use the Porter Stemmer to normalize some tokenized text, calling the stemmer on each word. Do the same thing with the Lancaster Stemmer and see if you observe any differences.

Code:

porter = nltk.PorterStemmer()  
lancaster = nltk.LancasterStemmer()  
port\_list = []  
lan\_list = []  
print(SimpleText)  
  
Simple = nltk.word\_tokenize(SimpleText)  
**for** word **in** Simple:  
 port = porter.stem(word)  
 port\_list.append(port)  
 lan = lancaster.stem(word)  
 lan\_list.append(lan)  
print(**"porter: "**, port\_list)  
print(**"lancaster: "**, lan\_list)

Result:

One day, his horse ran away. The neighbors came to express their concern: "Oh, that's too bad. How are you going to work the fields now?" The farmer replied: "Good thing, Bad thing, Who knows?" In a few days, his horse came back and brought another horse with her. Now, the neighbors were glad: "Oh, how lucky! Now you can do twice as much work as before!" The farmer replied: "Good thing, Bad thing, Who knows?"

porter: ['one', 'day', ',', 'hi', 'hors', 'ran', 'away', '.', 'the', 'neighbor', 'came', 'to', 'express', 'their', 'concern', ':', '``', 'Oh', ',', 'that', "'s", 'too', 'bad', '.', 'how', 'are', 'you', 'go', 'to', 'work', 'the', 'field', 'now', '?', "''", 'the', 'farmer', 'repli', ':', '``', 'good', 'thing', ',', 'bad', 'thing', ',', 'who', 'know', '?', "''", 'In', 'a', 'few', 'day', ',', 'hi', 'hors', 'came', 'back', 'and', 'brought', 'anoth', 'hors', 'with', 'her', '.', 'now', ',', 'the', 'neighbor', 'were', 'glad', ':', '``', 'Oh', ',', 'how', 'lucki', '!', 'now', 'you', 'can', 'do', 'twice', 'as', 'much', 'work', 'as', 'befor', '!', "''", 'the', 'farmer', 'repli', ':', '``', 'good', 'thing', ',', 'bad', 'thing', ',', 'who', 'know', '?', "''"]

lancaster: ['on', 'day', ',', 'his', 'hors', 'ran', 'away', '.', 'the', 'neighb', 'cam', 'to', 'express', 'their', 'concern', ':', '``', 'oh', ',', 'that', "'s", 'too', 'bad', '.', 'how', 'ar', 'you', 'going', 'to', 'work', 'the', 'field', 'now', '?', "''", 'the', 'farm', 'reply', ':', '``', 'good', 'thing', ',', 'bad', 'thing', ',', 'who', 'know', '?', "''", 'in', 'a', 'few', 'day', ',', 'his', 'hors', 'cam', 'back', 'and', 'brought', 'anoth', 'hors', 'with', 'her', '.', 'now', ',', 'the', 'neighb', 'wer', 'glad', ':', '``', 'oh', ',', 'how', 'lucky', '!', 'now', 'you', 'can', 'do', 'twic', 'as', 'much', 'work', 'as', 'bef', '!', "''", 'the', 'farm', 'reply', ':', '``', 'good', 'thing', ',', 'bad', 'thing', ',', 'who', 'know', '?', "''"]

When we are using Porter and Lancaster to output, the lower-case words and punctuations are the same. However, the main differences between each other is the Upper-case word. If we have the original Upper-case word, Porter will output the original one, but Lancaster will output the lower-case one. In addition, the porter one will have more correctness than Lancaster one. For example, “neighbor” in the porter is correct, but in Lancaster it becomes “neighb”.

1. exercise 40. Section 3.8 in “Sentence Segmentaion” lists an example of using Punkt. Use nltk.word\_tokenize() function to tokenize given text into words.
   1. Test your code on “ABC rural News”. Command nltk.corpus.abc.raw('rural.txt') allows one to access “ABC rural News” as a string. Report your results on this text.
   2. Report what your code computes for “ABC Science news” from ntlk.corpus.abc .

★ Obtain raw texts from two or more genres and compute their respective reading difficulty scores as in the earlier exercise on reading difficulty. E.g. compare ABC Rural News and ABC Science News (nltk.corpus.abc). Use Punkt to perform sentence segmentation.

Code:

**def** ARI(raw):  
 sent\_tokenizer = nltk.data.load(**'tokenizers/punkt/english.pickle'**)  
 sents = [nltk.word\_tokenize(s) **for** s **in** sent\_tokenizer.tokenize(raw)]  
  
 words = nltk.word\_tokenize(raw)  
 summation\_w = 0.0  
 **for** w **in** words:  
 summation\_w += len(w)  
 summation\_s = 0.0  
 **for** s **in** sents:  
 summation\_s += len(s)  
  
 av\_wordlength = summation\_w / len(words)  
 av\_sentlength = summation\_s / len(sents)  
 **return** (4.71 \* av\_wordlength) + (0.5 \* av\_sentlength) - 21.43  
  
**def** exercise40():  
 print (**"rural: "**, ARI(nltk.corpus.abc.raw(**"rural.txt"**)))  
 print (**"Science: "**, ARI(nltk.corpus.abc.raw(**"science.txt"**)))

Result:

Exercise 40

rural: 12.61676279331764

Science: 12.773526577547678