Manasa.R
20BCE1055

Natural Language processing DA-1

```
import nltk
from nltk.corpus import brown
nltk.download('brown')
```

```
[nltk_data] Downloading package brown to /root/nltk_data...
[nltk_data]   Package brown is already up-to-date!
True
```

Utilize Python NLTK (Natural Language Tool Kit) Platform and do the following. Install relevant Packages and Libraries

Explore Brown Corpus and find the size, tokens, categories

```
len(brown.raw())
```

```
9964284
```

```
brown.categories()
```

```
['adventure',
 'belles_lettres',
 'editorial',
 'fiction',
 'government',
 'hobbies',
 'humor',
 'learned',
 'lore',
```



```
 'reviews',
 'romance',
 'science_fiction']
```

```
brown.words()
```

```
['The', 'Fulton', 'County', 'Grand', 'Jury', 'said', ...]
```

Find the size of word tokens?

```
len(brown.words())
```

```
1161192
```

Find the size of word types?

```
len(set(brown.words()))
```

```
56057
```

Find the size of the category “government”

```
len(brown.words(categories="government"))
```

```
70117
```

List the most frequent tokens

```
freq = nltk.FreqDist(brown.words())
print("Common Words:", freq.most_common(10))
```

```
Common Words: [('the', 62713), (',', 58334), ('.', 49346), ('of', 36080), ('and', 27915), ('to', 25732), ('a', 21881), ('in', 1953
```

Count the number of sentences

```
len(brown.sents())
```

```
57340
```

Explore the corpora available in NLTK

```
from nltk.corpus import gutenberg
nltk.download('gutenberg')
from nltk.corpus import reuters
nltk.download('reuters')
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')
nltk.download('conll2000')
nltk.download('treebank')
nltk.download('conll2007')
nltk.download('indian')
nltk.download('wordnet')
```

```
[nltk_data] Downloading package gutenberg to /root/nltk_data...
[nltk_data]   Package gutenberg is already up-to-date!
[nltk_data] Downloading package reuters to /root/nltk_data...
[nltk_data]   Package reuters is already up-to-date!
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /root/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
[nltk_data] Downloading package conll2000 to /root/nltk_data...
[nltk_data]   Package conll2000 is already up-to-date!
[nltk_data] Downloading package treebank to /root/nltk_data...
[nltk_data]   Package treebank is already up-to-date!
[nltk_data] Downloading package conll2007 to /root/nltk_data...
[nltk_data]   Package conll2007 is already up-to-date!
[nltk_data] Downloading package indian to /root/nltk_data...
[nltk_data]   Package indian is already up-to-date!
```


Raw corpus

```
gutenberg.raw()
```

```
'[Emma by Jane Austen 1816]\n\nVOLUME I\n\nCHAPTER I\n\n\nEmma Woodhouse, handsome, clever, and rich,
with a comfortable home\nand happy disposition, seemed to unite some of the best blessings\nof existen
ce; and had lived nearly twenty-one years in the world\nwith very little to distress or vex her.\n\nSh
e was the youngest of the two daughters of a most affectionate,\nindulgent father; and had, in consequ
ence of her sister\'s marriage,\nbeen mistress of his house from a very early period.  Her mother\nhad
died too long ago for her to have more than an indistinct\nremembrance of her caresses; and her place
had been supplied\nby an excellent woman as governess, who had fallen little short\nof a mother in aff
ection.\n\nSixteen years had Miss Taylor been in Mr. Woodhouse\'s family,\nless as a governess than a
```

```
reuters.raw()
```

```
'ASIAN EXPORTERS FEAR DAMAGE FROM U.S.-JAPAN RIFT\n  Mounting trade friction between the\n  U.S. And J
apan has raised fears among many of Asia\'s exporting\n  nations that the row could inflict far-reachi
ng economic\n  damage, businessmen and officials said.\n      They told Reuter correspondents in Asian
capitals a U.S.\n  Move against Japan might boost protectionist sentiment in the\n  U.S. And lead to c
urbs on American imports of their products.\n      But some exporters said that while the conflict wou
ld hurt\n  them in the long-run, in the short-term Tokyo\'s loss might be\n  their gain.\n      The U.
S. Has said it will impose 300 mln dlrs of tariffs on\n  imports of Japanese electronics goods on Apri
l 17, in\n  retaliation for Japan\'s alleged failure to stick to a pact not\n  to sell semiconductors
```

POS tagged

```
print(brown.tagged_words())
```

```
[('The', 'AT'), ('Fulton', 'NP-TL'), ...]
```

```
from nltk.corpus import conll2000, switchboard
print(conll2000.tagged_words())
```

```
[('Confidence', 'NN'), ('in', 'IN'), ('the', 'DT'), ...]
```

Parsed

```
from nltk.corpus import treebank
print(treebank.parsed_sents('wsj_0003.mrg')[0])
```

```
      (S
        (S-TPC-1
          (NP-SBJ
            (NP (NP (DT A) (NN form)) (PP (IN of) (NP (NN asbestos))))
            (RRC
              (ADVP-TMP (RB once))
              (VP
                (VBN used)
                (NP (-NONE- *))
                (S-CLR
                  (NP-SBJ (-NONE- *))
                  (VP
                    (TO to)
                    (VP
                      (VB make)
                      (NP (NNP Kent) (NN cigarette) (NNS filters))))))))))
          (VP
            (VBZ has)
            (VP
              (VBN caused)
              (NP
                (NP (DT a) (JJ high) (NN percentage))
                (PP (IN of) (NP (NN cancer) (NNS deaths)))
                (PP-LOC
                  (IN among)
                  (NP
                    (NP (DT a) (NN group))
                    (PP
                      (IN of)
                      (NP
                        (NP (NNS workers))
                        (RRC
                          (VP
                            (VBN exposed)
                            (NP (-NONE- *))
                            (PP-CLR (TO to) (NP (PRP it)))
                            (ADVP-TMP
                              (NP
                                (QP (RBR more) (IN than) (CD 30))
```



```
          (NP-SBJ (NNS researchers))
          (VP (VBD reported) (SBAR (-NONE- 0) (S (-NONE- *T*-1))))
          (. .))
```

```
from nltk.corpus import conll2007
print(conll2007.parsed_sents('esp.train')[0].tree())
```

```
      (fortaleció
        (aumento El (del (índice (de (desempleo estadounidense)))))
        hoy
        considerablemente
        (al
          (euro
            (cotizaba
              ,
              que
              (a (15.35 las GMT))
              se
              (en (mercado el (de divisas) (de Fráncfort)))
              (a 0,9452_dólares)
              (frente_a , (0,9349_dólares los (de (mañana esta)))))))
        .)
```

Multilingual aligned

```
from nltk.corpus import wordnet as wn
nltk.download('omw-1.4')
```

```
      [nltk_data] Downloading package omw-1.4 to /root/nltk_data...
      [nltk_data]   Package omw-1.4 is already up-to-date!
      True
```

```
wn.langs()
```

```
      dict_keys(['eng', 'als', 'arb', 'bul', 'cmn', 'dan', 'ell', 'fin', 'fra', 'heb', 'hrv', 'isl', 'ita', 'ita_iwn', 'jpn', 'cat',
      'eus', 'glg', 'spa', 'ind', 'zsm', 'nld', 'nno', 'nob', 'pol', 'por', 'ron', 'lit', 'slk', 'slv', 'swe', 'tha'])
```

Spoken language

```
from nltk.corpus import indian
indian.raw()
```

```
'<Corpora type="Monolingual-POS-TAGGED" Language="Bangla">\n<Sentence id=1>\nমহিষের_NN সন্তান_NN :_SYM
তোড়া_NNP উপজাতি_NN ।_SYM \n</Sentence>\n<Sentence id=2>\nবাসস্থান-ঘরগৃহস্থালি_NN তোড়া_NNP ভাষায়_NN গ্রামকেও
_NN বলে_VM `_SYM মোদ_NN \'_SYM ।_SYM \n</Sentence>\n<Sentence id=3>\nমোদের_NN আয়তন_NN খুব_INTF বড়ো_
JJ নয়_VM ।_SYM \n</Sentence>\n<Sentence id=4>\nপ্রতি_QF মোদে_NN আছে_VM কিছু_QF কুঁড়েঘর_NN ,_SYM সাধারণ_J
J মহিষশালা_NN ।_SYM \n</Sentence>\n<Sentence id=5>\nআর_CC গ্রামের_NN বাইরে_NST থাকে_VM ডেয়ারি-মন্দির_NN ।_
SYM \n</Sentence>\n<Sentence id=6>\nআয়তনের_NN তারতম্য_NN অনুসারে_PSP গ্রামগুলি_NN দু_QC রকমের_NN :_SYM
এতূডমোদ_NNP (_SYM বড়ো_JJ গ্রাম_NN )_SYM ওকিনমোদ_NNP (_SYM ছোট_JJ গ্রাম_NN )_SYM ।_SYM \n</Sentence>\n<
Sentence id=7>\nকোন_DEM কোন_RDP গ্রামের_NN আবার_CC ধর্মীয়_JJ বা_C\ufeffমহিষের_NN সন্তান_NN :_SYM তোড়া_NN
```

Semantic tagged

```
brown.categories()
```

```
['adventure',
 'belles_lettres',
 'editorial',
 'fiction',
 'government',
 'hobbies',
 'humor',
 'learned',
 'lore',
 'mystery',
 'news',
 'religion',
 'reviews',
 'romance',
 'science_fiction']
```

```
reuters.categories()
```

```
'income',
'instal-debt',
'interest',
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff

```
'jobs',
'l-cattle',
'lead',
'lei',
'lin-oil',
'livestock',
'lumber',
'meal-feed',
'money-fx',
'money-supply',
'naphtha',
'nat-gas',
'nickel',
'nkr',
'nzdlr',
```

```
 'veg-oil',
 'wheat',
 'wpi',
 'yen',
 'zinc']
```

Create a text corpus with a minimum of 200 words (unique content). Implement the following text processing

Nearly 161 million persons live with a disabling visual impairment, of whom 37 million are blind. About 90% of them live in developing countries of Africa, Asia, Latin America and the Pacific Regions. 9 out of 10 blind children in developing countries have no access to education. The system of embossed writing invented by Louis Braille gradually came to be accepted throughout the world as the fundamental form of written communication for blind individuals. This paper is concerned about the transliteration of English and Hindi text to Braille. Braille is a dotted pattern used by the blind people for reading and writing. In this paper we have discussed the possible ways to teach Blind people. Audio, Braille. The problems with the particular pattern and which approach is better is also discussed .In this transliteration we have used a chart as database, from where we have done mapping for the corresponding Braille representation ,then implementation ,testing of the system and future scope are discussed. English to Braille conversion responds to increased demands on the Braille code that is integrated education of blind children; great diversity of presentation techniques used in printed textbooks; computer aided translation from print to Braille; globalization underpinning resource sharing and the widespread use of English as a second or further language. The change inherent in English to Braille transliteration is mostly minor for literary Braille and most evident for mathematics and science notation – making Braille easier to learn by all stakeholders and easier to read and write for blind people. English to Braille Transliteration is optimized for use by students integrated into regular schools; ideal for students using English as their primary and secondary language; and well –suited to the needs of blind students in developing countries.

```
import os
```

```
PATH = os.getcwd()
FILE_NAME = "sample.txt"
```

```
from nltk.corpus.reader.plaintext import PlaintextCorpusReader
```



```
corpus.raw()
```

```
'Nearly 161 million persons live with a disabling visual impairment, of whom 37 million are blind. Abo
ut 90% of them live in developing countries of Africa, Asia, Latin America and the Pacific Regions. 9
out of 10 blind children in developing countries have no access to education. The system of embossed w
riting invented by Louis Braille gradually came to be accepted throughout the world as the fundamental
form of written communication for blind individuals. This paper is concerned about the transliteration
of English and Hindi text to Braille. Braille is a dotted pattern used by the blind people for reading
and writing. In this paper we have discussed the possible ways to teach Blind people. Audio, Braille.
The problems with the particular pattern and which approach is better is also discussed .In this trans
```

Word segmentation

```
corpus.words()
```

```
['Nearly', '161', 'million', 'persons', 'live', 'with', ...]
```

Sentence segmentation

```
corpus.sents()
```

```
[['Nearly', '161', 'million', 'persons', 'live', 'with', 'a', 'disabling', 'visual', 'impairment', ',', 'of', 'whom', '37',
'million', 'are', 'blind', '.'], ['About', '90', '%', 'of', 'them', 'live', 'in', 'developing', 'countries', 'of', 'Africa', ',',
'Asia', ',', 'Latin', 'America', 'and', 'the', 'Pacific', 'Regions', '.'], ...]
```

Convert to Lowercase

```
text = 'Nearly 161 million persons live with a disabling visual impairment, of whom 37 million are blind. About 90% of them live in deve
```

```
text.lower()
```

```
'nearly 161 million persons live with a disabling visual impairment, of whom 37 million are blind. abo
ut 90% of them live in developing countries of africa, asia, latin america and the pacific regions. 9
out of 10 blind children in developing countries have no access to education. the system of embossed w
riting invented by louis braille gradually came to be accepted throughout the world as the fundamental
form of written communication for blind individuals. this paper is concerned about the transliteration
of english and hindi text to braille. braille is a dotted pattern used by the blind people for reading
and writing. in this paper we have discussed the possible ways to teach blind people. audio, braille.
the problems with the particular pattern and which approach is better is also discussed .in this trans
```

Stop words removal

```
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
True
```

```
from nltk.corpus import stopwords
en_stops = set(stopwords.words('english'))
words = []
for x in corpus.words():
    if x not in en_stops:
        words.append(x)
print(words)
```

```
['Nearly', '161', 'million', 'persons', 'live', 'disabling', 'visual', 'impairment', ',', '37', 'million', 'blind', '.', 'About',
```

Stemming

```
# importing modules
from nltk.stem import PorterStemmer
from nltk.tokenize import word_tokenize

ps = PorterStemmer()

words = word_tokenize(text)
s = ""
l = []

for w in words:
    s = w + " : " + ps.stem(w)
    l.append(s)
print(l)
```



```
['Nearly : nearli', '161 : 161', 'million : million', 'persons : person', 'live : live', 'with : with', 'a : a', 'disabling : disat
```

Lemmatization

```
import nltk
from nltk.stem import WordNetLemmatizer
from nltk.corpus import wordnet

lemmatizer = WordNetLemmatizer()

# function to convert nltk tag to wordnet tag
def nltk_tag_to_wordnet_tag(nltk_tag):
    if nltk_tag.startswith('J'):
        return wordnet.ADJ
    elif nltk_tag.startswith('V'):
        return wordnet.VERB
    elif nltk_tag.startswith('N'):
        return wordnet.NOUN
    elif nltk_tag.startswith('R'):
        return wordnet.ADV
    else:
        return None

def lemmatize_sentence(sentence):
    #tokenize the sentence and find the POS tag for each token
    nltk_tagged = nltk.pos_tag(nltk.word_tokenize(sentence))
    #tuple of (token, wordnet_tag)
    wordnet_tagged = map(lambda x: (x[0], nltk_tag_to_wordnet_tag(x[1])), nltk_tagged)
    lemmatized_sentence = []
    for word, tag in wordnet_tagged:
        if tag is None:
            #if there is no available tag, append the token as is
            lemmatized_sentence.append(word + " : " + word )
        else:
            #else use the tag to lemmatize the token
            lemmatized_sentence.append(word + " : " + lemmatizer.lemmatize(word, tag))
    return lemmatized_sentence

print(lemmatize_sentence(text))
```

```
['Nearly : Nearly', '161 : 161', 'million : million', 'persons : person', 'live : live', 'with : with', 'a : a', 'disabling : disat
```

Part of speech tagger

```
nltk_tagged = nltk.pos_tag(nltk.word_tokenize(text))
nltk_tagged
```

```
('making', 'VBG'),
('Braille', 'NNP'),
('easier', 'JJR'),
('to', 'TO'),
('learn', 'VB'),
('by', 'IN'),
('all', 'DT'),
('stakeholders', 'NNS'),
('and', 'CC'),
('easier', 'JJR'),
('to', 'TO'),
('read', 'VB'),
('and', 'CC'),
('write', 'VB'),
('for', 'IN'),
('blind', 'JJ'),
('people', 'NNS'),
('.', '.'),
('English', 'JJ'),
('to', 'TO'),
('Braille', 'NNP'),
('Transliteration', 'NNP'),
('is', 'VBZ'),
('optimized', 'VBN'),
('for', 'IN'),
('use', 'NN'),
('by', 'IN'),
('students', 'NNS'),
('integrated', 'VBN'),
('into', 'IN'),
('regular', 'JJ'),
('schools', 'NNS'),
```



```
('students', 'NNS'),
('using', 'VBG'),
('English', 'NNP'),
('as', 'IN'),
('their', 'PRP$'),
('primary', 'NN'),
('and', 'CC'),
('secondary', 'JJ'),
('language', 'NN'),
(';', ':'),
('and', 'CC'),
('well', 'RB'),
('-suited', 'VBN'),
('to', 'TO'),
('the', 'DT'),
('needs', 'NNS'),
('of', 'IN'),
('blind', 'NN'),
('students', 'NNS'),
('in', 'IN'),
('developing', 'VBG'),
('countries', 'NNS'),
('.', '.')]
```