Lab#4

September 20, 2022

[9]:
```
from nltk.tokenize import word_tokenize, sent_tokenize

input_sentence = "Brocolli is good to eat. My brother likes to eat good␣
 ↪brocolli, but not my mother."

number_of_words = word_tokenize(input_sentence)
print("number_of_words :", len(number_of_words))

number_of_sentences = sent_tokenize(input_sentence)
print("number_of_sentences : ", len(number_of_sentences))

print("number_of_paragraphs : ", input_sentence.count('\n\n') + 1)
```

```
number_of_words : 19
number_of_sentences :  2
number_of_paragraphs :  1
```

[4]:
```
pip install gensim
```

```
Requirement already satisfied: gensim in c:\users\legion\anaconda3\lib\site-
packages (3.8.3)
Requirement already satisfied: scipy>=0.18.1 in
c:\users\legion\anaconda3\lib\site-packages (from gensim) (1.4.1)
Requirement already satisfied: smart-open>=1.8.1 in
c:\users\legion\anaconda3\lib\site-packages (from gensim) (4.1.2)
Requirement already satisfied: six>=1.5.0 in c:\users\legion\anaconda3\lib\site-
packages (from gensim) (1.14.0)
Requirement already satisfied: Cython==0.29.14 in
c:\users\legion\anaconda3\lib\site-packages (from gensim) (0.29.14)
Requirement already satisfied: numpy>=1.11.3 in
c:\users\legion\anaconda3\lib\site-packages (from gensim) (1.18.1)
Note: you may need to restart the kernel to use updated packages.
```

[5]:
```
pip install stop-words
```

```
Requirement already satisfied: stop-words in c:\users\legion\anaconda3\lib\site-
packages (2018.7.23)
Note: you may need to restart the kernel to use updated packages.
```

[7]:
```
from nltk.tokenize import RegexpTokenizer
from stop_words import get_stop_words
from nltk.stem.porter import PorterStemmer
from gensim import corpora, models
import gensim

tokenizer = RegexpTokenizer(r'\w+')

# create English stop words list
en_stop = get_stop_words('en')

# Create p_stemmer of class PorterStemmer
p_stemmer = PorterStemmer()

# create sample documents
doc_a = "Some health experts suggest that driving may cause increased tension␣
 ↪and blood pressure."
doc_b = "I often feel pressure to perform well at school, but my mother never␣
 ↪seems to drive my brother to do better."
doc_c = "Health professionals say that brocolli is good for your health."
doc_d = "Brocolli is good to eat. My brother likes to eat good brocolli, but␣
 ↪not my mother."
doc_e = "My mother spends a lot of time driving my brother around to baseball␣
 ↪practice."


# compile sample documents into a list
doc_set = [doc_a, doc_b, doc_c, doc_d, doc_e]

# list for tokenized documents in loop
texts = []

# loop through document list
for i in doc_set:

    # clean and tokenize document string
    raw = i.lower()
    tokens = tokenizer.tokenize(raw)

    # remove stop words from tokens
    stopped_tokens = [i for i in tokens if not i in en_stop]

    # stem tokens
    stemmed_tokens = [p_stemmer.stem(i) for i in stopped_tokens]

    # add tokens to list
    texts.append(stemmed_tokens)
```

```
# turn our tokenized documents into a id <-> term dictionary
dictionary = corpora.Dictionary(texts)

# convert tokenized documents into a document-term matrix
corpus = [dictionary.doc2bow(text) for text in texts]

# generate LDA model
ldamodel = gensim.models.ldamodel.LdaModel(corpus, num_topics=2, id2word =␣
 ↪dictionary, passes=20)
print(ldamodel.print_topics(num_words=3))
```

```
[(0, '0.072*"drive" + 0.043*"health" + 0.043*"pressur"'), (1, '0.081*"good" +
0.081*"brocolli" + 0.059*"mother"')]
```

[ ]: