Week7_5_NLP_text_cleaning

May 21, 2021

Natural Language Processing (NLP) Text Preprocessing

[1]:
```
import nltk
import re
import string
import pandas as pd
from nltk.stem import PorterStemmer
```

Lowercasing

[2]:
```
words=["America","AMERICA","HTTP://gil","AmeriCA"]
lwrwrds = [word.lower() for word in words]
lwrwrds
```

[2]:
```
['america', 'america', 'http://gil', 'america']
```

[3]:
```
sentence = "Alabama Republican Gov. Kay Ivey signed into law on Monday a bill !␣
↪legalizing medical marijuana // > in the state?."
# Split my_string on sentence endings and print the result
sentence_endings = r"[.,!,?,>, //]"
print(re.split(sentence_endings, sentence))
```

```
['Alabama', 'Republican', 'Gov', '', 'Kay', 'Ivey', 'signed', 'into', 'law',
'on', 'Monday', 'a', 'bill', '', '', 'legalizing', 'medical', 'marijuana', '',
'', '', '', '', 'in', 'the', 'state', '', '']
```

Removing HTTP links, URL address

[4]:
```
words=["America","AMERICA","http://gil","AmeriCA"]
#Removal of HTTP links/URLs mixed up in any text:
for word in words:
    cleanword = [re.sub('http://\S+|https://\S+','', word)]
    print(cleanword)
```

```
['America']
['AMERICA']
['']
['AmeriCA']
```

Lemmatization

[5]:
```
from nltk.stem import WordNetLemmatizer
nltk.download('wordnet')

# init lemmatizer
lemmatizer = WordNetLemmatizer()

words=["connect","connected","connection","connections","connects"]
lemmatized_words=[lemmatizer.lemmatize(word=word,pos='v') for word in words]

#Prepare into a data table
lemmatizeddf= pd.DataFrame({'original_word': words,'lemmatized_word':␣
 ↪lemmatized_words})
lemmatizeddf=lemmatizeddf[['original_word','lemmatized_word']]
lemmatizeddf
```

```
[nltk_data] Downloading package wordnet to
[nltk_data]     /home/jayanthikishore/nltk_data…
[nltk_data]   Package wordnet is already up-to-date!
```

[5]:

	original_word	lemmatized_word
0	connect	connect
1	connected	connect
2	connection	connection
3	connections	connections
4	connects	connect

Stemming

[6]:
```
import nltk
import pandas as pd
from nltk.stem import PorterStemmer

# init stemmer
porter_stemmer=PorterStemmer()

words=["connect","connected","connection","connections","connects"]

stemmed_words=[porter_stemmer.stem(word=word) for word in words]

#prepare dataframe
stemdf= pd.DataFrame({'original_word': words,'stemmed_word': stemmed_words})
stemdf
```

[6]:

	original_word	stemmed_word
0	connect	connect
1	connected	connect
2	connection	connect

```
3   connections      connect
4      connects      connect
```

Stop words

[7]:
```
stopwords=['this','that','and','a','we','it','to','is','of','up','need','the','there']
text="this is a text full of content and we need to clean it up"

words=text.split(" ")
shortlisted_words=[]

#remove stop words
for w in words:
    if w not in stopwords:
        shortlisted_words.append(w)
    else:
        shortlisted_words.append("R")

print("original sentence = ",text)
print("sentence with stop words removed= "," ".join(shortlisted_words))
```

```
original sentence =  this is a text full of content and we need to clean it up
sentence with stop words removed=  R R R text full R content R R R R clean R R
```

Noise Removal

[8]:
```
import nltk
import pandas as pd
import re
from nltk.stem import PorterStemmer

porter_stemmer=PorterStemmer()

raw_words=["..trouble..","trouble<","trouble!","<a>trouble</a>",'1.trouble']
stemmed_words=[porter_stemmer.stem(word=word) for word in raw_words]

#concating nating original and output into a table
stemdf= pd.DataFrame({'raw_word': raw_words,'stemmed_word': stemmed_words})
stemdf
```

[8]:

	raw_word	stemmed_word
0	..trouble..	..trouble..
1	trouble<	trouble<
2	trouble!	trouble!
3	<a>trouble	<a>trouble
4	1.trouble	1.troubl

Split into words

[9]:
```
sentence= "This is certainly a sensitive ? and emotional issue and something␣
 ↪that is continually > being studied, Ivey said in a statement. On the state␣
 ↪level, we have had a study group that has looked closely at this issue, and␣
 ↪I am interested in the potential good medical cannabis can have for those␣
 ↪with chronic illnesses or what it can do to improve the quality of life of␣
 ↪those in their final days! !!."

from nltk.tokenize import word_tokenize
tokens = word_tokenize(sentence)
print(tokens[:100])
```

```
['This', 'is', 'certainly', 'a', 'sensitive', '?', 'and', 'emotional', 'issue',
'and', 'something', 'that', 'is', 'continually', '>', 'being', 'studied', ',',
'Ivey', 'said', 'in', 'a', 'statement', '.', 'On', 'the', 'state', 'level', ',',
'we', 'have', 'had', 'a', 'study', 'group', 'that', 'has', 'looked', 'closely',
'at', 'this', 'issue', ',', 'and', 'I', 'am', 'interested', 'in', 'the',
'potential', 'good', 'medical', 'cannabis', 'can', 'have', 'for', 'those',
'with', 'chronic', 'illnesses', 'or', 'what', 'it', 'can', 'do', 'to',
'improve', 'the', 'quality', 'of', 'life', 'of', 'those', 'in', 'their',
'final', 'days', '!', '!', '!', '.']
```

Filterout punctuation marks

[10]:
```
# remove all tokens that are not alphabetic
words = [word for word in tokens if word.isalpha()]
print(words[:100])
```

```
['This', 'is', 'certainly', 'a', 'sensitive', 'and', 'emotional', 'issue',
'and', 'something', 'that', 'is', 'continually', 'being', 'studied', 'Ivey',
'said', 'in', 'a', 'statement', 'On', 'the', 'state', 'level', 'we', 'have',
'had', 'a', 'study', 'group', 'that', 'has', 'looked', 'closely', 'at', 'this',
'issue', 'and', 'I', 'am', 'interested', 'in', 'the', 'potential', 'good',
'medical', 'cannabis', 'can', 'have', 'for', 'those', 'with', 'chronic',
'illnesses', 'or', 'what', 'it', 'can', 'do', 'to', 'improve', 'the', 'quality',
'of', 'life', 'of', 'those', 'in', 'their', 'final', 'days']
```

Filterout stop words

[11]:
```
from nltk.corpus import stopwords
stop_words = stopwords.words('english')
# print(stop_words)

words = [w for w in words if not w in stop_words]
print(words[:100])
```

```
['This', 'certainly', 'sensitive', 'emotional', 'issue', 'something',
'continually', 'studied', 'Ivey', 'said', 'statement', 'On', 'state', 'level',
'study', 'group', 'looked', 'closely', 'issue', 'I', 'interested', 'potential',
'good', 'medical', 'cannabis', 'chronic', 'illnesses', 'improve', 'quality',
```

```
'life', 'final', 'days']
```

[ ]: