Aim: Implement problems on natural language processing - Part of Speech tagging, N-gram & smoothening and Chunking using NLTK

Short notes:

The **Natural Language Toolkit (NLTK)** is a platform used for building programs for text analysis. One of the more powerful aspects of the NLTK module is the Part of Speech tagging.

Part-of-speech (POS) tagging is a process of converting a sentence to forms – list of words, list of tuples (where each tuple is having a form (word, tag)). The tag in case of is a part-of-speech tag, and signifies whether the word is a noun, adjective, verb, and so on.

keywords:

Corpus : Body of text, singular. Corpora is the plural of this.

Lexicon : Words and their meanings.

Token : Each “entity” that is a part of whatever was split up based on rules.

Tags and their meanings

CD cardinal digit

EX existential there (like: “there is” … think of it like “there exists”)

FW foreign word

IN preposition/subordinating conjunction

JJ adjective ‘big’

JJR adjective, comparative ‘bigger’

JJS adjective, superlative ‘biggest’

NN noun, singular ‘desk’

NNS noun plural ‘desks’

NNP proper noun, singular ‘Harrison’

NNPS proper noun, plural ‘Americans’

PDT predeterminer ‘all the kids’

POS possessive ending parent‘s

PRP personal pronoun I, he, she

PRP$ possessive pronoun my, his, hers

RB adverb very, silently,

RBR adverb, comparative better

RBS adverb, superlative best

RP particle give up

N-grams are continuous sequences of words or symbols or tokens in a document. In technical terms, they can be defined as the neighbouring sequences of items in a document.

Steps for n-gram model:

Explore the dataset

Feature extraction

Train-test split

Basic pre-processing

Code to generate N-grams

Creating unigrams

Creating bigrams

Creating trigrams

```
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize,sent_tokenize
nltk.download('stopwords')
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')
stop_words=set(stopwords.words('english'))
#Dummy text
txt="India is my country.All Indians are my brothers and sisters.I love my country."

# sent_tokenize is one of instances of
# PunktSentenceTokenizer from the nltk.tokenize.punkt module
tokenized=sent_tokenize(txt)
for i in tokenized:


    # Word tokenizers is used to find the words
    # and punctuation in a string
    wordsList=nltk.word_tokenize(i)


    # removing stop words from wordList
    wordsList=[w for w in wordsList if not w in stop_words]


    #  Using a Tagger. Which is part-of-speech
    # tagger or POS-tagger.
    tagged=nltk.pos_tag(wordsList)
```

```
30     print(tagged)
31
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords 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!
[('India', 'NNP'), ('country.All', 'NN'), ('Indians', 'NNP'), ('brothers', 'NNS'), (
```

```
1 print(stop_words)
```

```
{"doesn't", 'needn', 'having', 'yourselves', 'into', 'because', 'now', 'when', 'as',
```

```
1 print(tokenized)
```

```
['India is my country.All Indians are my brothers and sisters.I love my country.']
```

N-gram model

```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 plt.style.use(style='seaborn')
5 #get the data from https://www.kaggle.com/ankurzing/sentiment-analysis-for-financial-ne
6 colnames=['Sentiment','news']
7 df=pd.read_csv('all-data.csv',encoding='ISO-8859-1',names=colnames)
8 df.head()
9
10
11
12
13
14
```

	Sentiment	news
0	neutral	According to Gran , the company has no plans t...
1	neutral	Technopolis plans to develop in stages an area...
2	negative	The international electronic industry company ...
3	positive	With the new production plant the company woul...
4	positive	According to the company 's updated strategy f...

```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4846 entries, 0 to 4845
Data columns (total 2 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   Sentiment  4846 non-null   object
 1   news       4846 non-null   object
dtypes: object(2)
memory usage: 75.8+ KB
```

```
df['Sentiment'].value_counts()
```

```
neutral     2879
positive    1363
negative     604
Name: Sentiment, dtype: int64
```

```
y=df['Sentiment'].values
y.shape
```

```
(4846,)
```

```
x=df['news'].values
x.shape
```

```
(4846,)
```

```
#Split train dataset
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.4,random_state=42)
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)

```

```
(2907,)
(1939,)
(2907,)
(1939,)
```

```
#Make train dataset as a dataframe
df1=pd.DataFrame(x_train)
df1=df1.rename(columns={0:'news'})
df2=pd.DataFrame(y_train)
df2=df2.rename(columns={0:'Sentiment'})
df_train=pd.concat([df1,df2],axis=1)
df_train
```

	news	Sentiment
0	Exel Composites ' long-term growth prospects r...	positive
1	The Samsung Mobile Applications Store was laun...	neutral
2	Altogether CapMan employs approximately 150 pe...	neutral
3	The segments through which the company operate...	neutral
4	UK 's Sarantel to outsource part of its proces...	neutral
...	...	...
2902	The currency effect had a 3.0 pct , or 20 mln ...	negative
2903	`` Lidskoe Pivo 's investment program foresees...	positive
2904	Products include Consumer Electronics devices ...	neutral
2905	The bridge is part of the highway 14 developme...	neutral

```
#Make test dataset as a dataframe
df3=pd.DataFrame(x_train)
df3=df3.rename(columns={0:'news'})
df4=pd.DataFrame(y_train)
df4=df4.rename(columns={0:'Sentiment'})
df_test=pd.concat([df3,df4],axis=1)
df_test
```

	news	Sentiment
0	Exel Composites ' long-term growth prospects r...	positive
1	The Samsung Mobile Applications Store was laun...	neutral
2	Altogether CapMan employs approximately 150 pe...	neutral
3	The segments through which the company operate...	neutral
4	UK 's Sarantel to outsource part of its proces...	neutral
...	...	...
2902	The currency effect had a 3.0 pct , or 20 mln ...	negative
2903	`` Lidskoe Pivo 's investment program foresees...	positive
2904	Products include Consumer Electronics devices ...	neutral
2905	The bridge is part of the highway 14 developme...	neutral
2906	(ADP News) - Oct 1 , 2008 - Finnish consulti...	positive

2907 rows × 2 columns

```
#removing punctuations
#library that contains punctuation
import string
string.punctuation

```

```
'!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
```

```
#defining the function to remove punctuation
def remove_punctuation(text):
  if(type(text)==float):
    return text
  ans=""
  for i in text:
    if i not in string.punctuation:
       ans+=i
  return ans

```

```
text='Welcome to_@sngce'
remove_punctuation(text)
```

```
'Welcome tosngce'
```

```
#storing the puntuation free text in a new column called clean_msg
df_train['news']=df_train['news'].apply(remove_punctuation)
df_test['news']=df_test['news'].apply(remove_punctuation)
df_train.head()

#punctuations are removed from news column in train dataset
```

	news	Sentiment
0	Exel Composites longterm growth prospects rem...	positive
1	The Samsung Mobile Applications Store was laun...	neutral
2	Altogether CapMan employs approximately 150 pe...	neutral
3	The segments through which the company operate...	neutral
4	UK s Sarantel to outsource part of its process...	neutral

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

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

```
#method to generate n-grams:
#params:
#text-the text for which we have to generate n-grams
#ngram-number of grams to be generated from the text(1,2,3,4 etc., default value=1)
def generate_N_grams(text,ngram=1):
  words=[word for word in text.split(" ") if word not in set(stopwords.words('english')
  print("Sentence after removing stopwords:",words)
```

```
    temp=zip(*[words[i:] for i in range(0,ngram)])
    ans=[' '.join(ngram)for ngram in temp]
    return ans

```

```
generate_N_grams("The sun rises in the east",2)
```

```
Sentence after removing stopwords: ['The', 'sun', 'rises', 'east']
['The sun', 'sun rises', 'rises east']
```

```
generate_N_grams("The sun rises in the east",3)
```

```
Sentence after removing stopwords: ['The', 'sun', 'rises', 'east']
['The sun rises', 'sun rises east']
```

```
generate_N_grams("The sun rises in the east",4)
```

```
Sentence after removing stopwords: ['The', 'sun', 'rises', 'east']
['The sun rises east']
```

```
name=['Krishnaindu','K.S']
s=' '.join(name)
s
```

```
'Krishnaindu K.S'
```

Double-click (or enter) to edit