Natural Language Toolkit (NLTK) Tutorial with Python

What is NLTK?

NLTK is a standard python library with prebuilt functions and utilities for the ease of use and implementation. It is one of the most used libraries for natural language processing and computational linguistics.

Reference Documentation: https://www.nltk.org/index.html

NLTK library based practical on Jupyter

Installing Jupyter:

- https://jupyter.org/install
- Macos: https://www.geeksforgeeks.org/how-to-install-jupyter-notebook-on-macos/conda
- Windows: https://www.geeksforgeeks.org/how-to-install-jupyter-notebook-in-windows/install -c anaconda nltk

Online editor of Jupyter:

https://hub.gke2.mybinder.org/user/ipython-ipython-in-depth-wexswrco/notebooks/binder/Index.ipynb

NLTK Installation Process:

Open a command prompt and type,

pip install nltk

Downloading Corpus:

```
import nltk
nltk.download()
```

```
print(os.listdir(nltk.data.find("corpora")))
```

Data pre-processing

```
NLP='''Natural language processing (NLP) refers to the branch of computer science—and more specifically, the branch of artificial intelligence  AI—concerned giving computers the ability to understand text  spoken words  much the same way human beings can NLP combines computational linguistics—rule-based modeling of human language—with statistical, machine learning,  deep learning models. Together, these technologies enable computers to process human language  the form of text voice data  to 'understand' its full meaning, complete with the speaker or writer's intent and sentiment.'''
```

```
type(NLP)
```

Sentence Tokenization:

```
import nltk
nltk.download('punkt')
from nltk.tokenize import sent_tokenize
tokenized_para = sent_tokenize(NLP)
print(tokenized_para)
```

Tokenization:

```
import nltk
from nltk.corpus import movie_reviews
movie reviews.words()

from nltk.corpus import brown
brown.words()


nltk.corpus.gutenberg.fileids()
```

```
hamlet=nltk.corpus.gutenberg.words('shakespeare-hamlet.txt')

for word in hamlet[:500]:
        print(word,sep=' ',end=' ')

for word in hamlet[:500]:
        print(word)


NLP="'Natural language processing (NLP) refers to the branch of computer
science—and more specifically, the branch of artificial intelligence  AI—concerned
giving computers the ability to understand text  spoken words  much the same way
human beings can NLP combines computational linguistics—rule-based modeling of
human language—with statistical, machine learning,  deep learning models. Together,
these technologies enable computers to process human language  the form of text
voice data  to 'understand' its full meaning, complete with the speaker or writer's intent
and sentiment.'"

type(NLP)



from nltk.tokenize import word_tokenize
nlp_tokens=word_tokenize(NLP)

Nlp_tokens
len(nlp_tokens)

#frequency distribution
from nltk.probability import FreqDist
fdist = FreqDist()

for word in nlp_tokens:
    fdist[word.lower()]+=1
fdist

#Specific word frequency
fdist['human']

#length of frequency list
```

```
len(fdist)

#top words with highest frequencies
comm=fdist.most_common(3)
comm

# new lines (lines)
from nltk.tokenize import blankline_tokenize
out = blankline_tokenize(NLP)
len(out)
```

Add \n and count the lines again

```
#printing 2nd paragraph
out[2]
```

Language Model

```
#using bigrams,  trigrams, ngrams

from nltk.util import bigrams, trigrams, ngrams

NLP='''Natural language processing (NLP) refers to the branch of computer
science—and more specifically, the branch of artificial intelligence  AI—concerned
giving computers the ability to understand text  spoken words  much the same way
human beings can NLP combines computational linguistics—rule-based modeling of
human language—with statistical, machine learning,  deep learning models. Together,
these technologies enable computers to process human language  the form of text
voice data  to ‘understand’ its full meaning, complete with the speaker or writer’s intent
and sentiment.'''



import nltk
from nltk.tokenize import word_tokenize
nltk.download('punkt')
#bigrams
nlp_tokens=nltk.word_tokenize(NLP)
nlpbi=list(nltk.bigrams(nlp_tokens))
```

```
nlpbi

#trigrams
nlptri=list(nltk.trigrams(nlp_tokens))
nlptri
#ngrams
nlpn=list(nltk.ngrams(nlp_tokens, N))
nlpn
```

Stemming

```
from nltk.stem import PorterStemmer
pst=PorterStemmer()

pst.stem('having')

inp=["give","giving","given","gave"]
for words in inp:
        print(pst.stem(words))


from nltk.stem import LancasterStemmer
lst=LancasterStemmer()

lst.stem(‘having’)

inp=["give","giving","given","gave"]
for words in inp:
        print(lst.stem(words))


from nltk.stem import SnowballStemmer
snst=SnowballStemmer('english')

snst.stem('having')


inp=["give","giving","given","gave"]
for words in inp:
```

```
    print(snst.stem(words))
```

Lemmatization

Eg of fish, fishes, fishing

```
from nltk.stem import wordnet
from nltk.stem import WordNetLemmatizer
lem=WordNetLemmatizer()

lem.lemmatize('corpora')

for words in inp:
    print(lem.lemmatize(words))
```

Stop words

```
from nltk.corpus import stopwords
stopwords.words('english')
len(stopwords.words('english'))


#Making the string of special symbols
import re
punctuation=re.compile(r'[-.?!,:;()|0-9]')


after_punctuation=[]
for words in nlp_tokens:
    word=punctuation.sub("",words)
    if len(words)>0:
        after_punctuation.append(word)




after_punctuation
```

POS: Parts of Speech

```
nltk.download('averaged_perceptron_tagger')
```

```
for tokens in nlp_tokens:
        print(nltk.pos_tag([tokens]))
```

Example:

```
Senten= "Ram is eating delicious cake"
temp_tokens=word_tokenize(Senten)

for token in temp_tokens:
        print(nltk.pos_tag([token]))
```

Named Entity Recognition

```
From nltk import ne_chunk
temp=”The Indian President stays in the Rashtrapati bhavan”
temp_tokens=word_tokenize(temp)
temp_tags=nltk.pos_tag(temp_tokens)

nltk.download('maxent_ne_chunker')

temp_NER = ne_chunk(temp_tags)
print(temp_NER)
```