In [2]:
```
pip install nltk
```

```
Defaulting to user installation because normal site-packages is not writeable
Collecting nltk
  Using cached nltk-3.7-py3-none-any.whl (1.5 MB)
Requirement already satisfied: click in /usr/lib/python3/dist-packages (from nltk)
(7.0)
Collecting tqdm
  Using cached tqdm-4.64.0-py2.py3-none-any.whl (78 kB)
Collecting joblib
  Using cached joblib-1.1.0-py2.py3-none-any.whl (306 kB)
Collecting regex>=2021.8.3
  Using cached regex-2022.4.24-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.w
hl (764 kB)
Installing collected packages: tqdm, regex, joblib, nltk
  WARNING: The script tqdm is installed in '/home/csl2/.local/bin' which is not on PA
TH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning,
 use --no-warn-script-location.
  WARNING: The script nltk is installed in '/home/csl2/.local/bin' which is not on PA
TH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning,
 use --no-warn-script-location.
Successfully installed joblib-1.1.0 nltk-3.7 regex-2022.4.24 tqdm-4.64.0
Note: you may need to restart the kernel to use updated packages.
```

In [3]:
```
##using NLTK library, we can do lot of text preprocesing
import nltk
```

In [4]:
```
#Sentence Tokenization
#Sentence tokenizer breaks text paragraph into sentences.

from nltk.tokenize import sent_tokenize
text="""Hello Mr. Smith, how are you doing today? The weather is great, and city is aw
The sky is pinkish-blue. You shouldn't eat cardboard"""

tokenized_text=sent_tokenize(text)
print(tokenized_text)
```

```
---------------------------------------------------------------------------
LookupError                               Traceback (most recent call last)
Input In [4], in <cell line: 8>()
      4 from nltk.tokenize import sent_tokenize
      5 text="""Hello Mr. Smith, how are you doing today? The weather is great, and c
ity is awesome.
      6 The sky is pinkish-blue. You shouldn't eat cardboard"""
----> 8 tokenized_text=sent_tokenize(text)
      9 print(tokenized_text)

File ~/notebook/jupyterenv/lib/python3.8/site-packages/nltk/tokenize/__init__.py:106,
in sent_tokenize(text, language)
     96 def sent_tokenize(text, language="english"):
     97     """
     98     Return a sentence-tokenized copy of *text*,
     99     using NLTK's recommended sentence tokenizer
   (...)
    104     :param language: the model name in the Punkt corpus
    105     """
--> 106     tokenizer = load(f"tokenizers/punkt/{language}.pickle")
    107     return tokenizer.tokenize(text)

File ~/notebook/jupyterenv/lib/python3.8/site-packages/nltk/data.py:750, in load(reso
urce_url, format, cache, verbose, logic_parser, fstruct_reader, encoding)
    747     print(f"<<Loading {resource_url}>>")
    749 # Load the resource.
--> 750 opened_resource = _open(resource_url)
    752 if format == "raw":
    753     resource_val = opened_resource.read()

File ~/notebook/jupyterenv/lib/python3.8/site-packages/nltk/data.py:876, in _open(res
ource_url)
    873 protocol, path_ = split_resource_url(resource_url)
    875 if protocol is None or protocol.lower() == "nltk":
--> 876     return find(path_, path + [""]).open()
    877 elif protocol.lower() == "file":
    878     # urllib might not use mode='rb', so handle this one ourselves:
    879     return find(path_, [""]).open()

File ~/notebook/jupyterenv/lib/python3.8/site-packages/nltk/data.py:583, in find(reso
urce_name, paths)
    581 sep = "*" * 70
    582 resource_not_found = f"\n{sep}\n{msg}\n{sep}\n"
--> 583 raise LookupError(resource_not_found)

LookupError:
**********************************************************************
  Resource punkt not found.
  Please use the NLTK Downloader to obtain the resource:

  >>> import nltk
  >>> nltk.download('punkt')

  For more information see: https://www.nltk.org/data.html

  Attempted to load tokenizers/punkt/PY3/english.pickle

  Searched in:
    - '/home/csl2/nltk_data'
    - '/home/csl2/notebook/jupyterenv/nltk_data'
```

```
    - '/home/csl2/notebook/jupyterenv/share/nltk_data'
    - '/home/csl2/notebook/jupyterenv/lib/nltk_data'
    - '/usr/share/nltk_data'
    - '/usr/local/share/nltk_data'
    - '/usr/lib/nltk_data'
    - '/usr/local/lib/nltk_data'
    - ''
**********************************************************************
```

In [ ]:
```
#Word Tokenization
#Word tokenizer breaks text paragraph into words.

from nltk.tokenize import word_tokenize
tokenized_word=word_tokenize(text)
print(tokenized_word)
```

In [ ]:
```
#Frequency Distribution

from nltk.probability import FreqDist
fdist = FreqDist(tokenized_word)
print(fdist)
```

In [ ]:
```
fdist.most_common(2)
```

In [ ]:
```
# Frequency Distribution Plot
import matplotlib.pyplot as plt
fdist.plot(30,cumulative=False)
plt.show()
```

In [ ]:
```
#Stopwords
#Stopwords considered as noise in the text.
#Text may contain stop words such as is, am, are, this, a, an, the, etc.

from nltk.corpus import stopwords
stop_words=set(stopwords.words("english"))
print(stop_words)
```

In [ ]:
```
#Removing Stopwords
#In NLTK for removing stopwords, you need to create a list of stopwords
#and filter out your list of tokens from these words.

filtered_sent=[]
for w in tokenized_word:
    if w not in stop_words:
        filtered_sent.append(w)
print("Tokenized Sentence:",tokenized_word)
print("Filterd Sentence:",filtered_sent)
```

In [ ]:
```
# Stemming
from nltk.stem import PorterStemmer
from nltk.tokenize import sent_tokenize, word_tokenize

ps = PorterStemmer()

stemmed_words=[]
for w in filtered_sent:
    stemmed_words.append(ps.stem(w))
```

```
print("Filtered Sentence:",filtered_sent)
print("Stemmed Sentence:",stemmed_words)
```

In [ ]:
```
#Lexicon Normalization
#performing stemming and Lemmatization

#nltk.download('wordnet')
#nltk.download('omw-1.4')
from nltk.stem.wordnet import WordNetLemmatizer
lem = WordNetLemmatizer()

from nltk.stem.porter import PorterStemmer
stem = PorterStemmer()

word = "flying"
print("Lemmatized Word:",lem.lemmatize(word,"v"))
print("Stemmed Word:",stem.stem(word))
```

In [ ]:
```
#POS Tagging

sent = "Albert Einstein was born in Ulm, Germany in 1879."
tokens=nltk.word_tokenize(sent)
print(tokens)
```

In [ ]:
```
#nltk.download('averaged_perceptron_tagger')
nltk.pos_tag(tokens)
```

In [ ]:
```

```