Text Analytics

1. Extract Sample document and apply following document preprocessing methods: Tokenization, POS Tagging, stop words removal, Stemming and Lemmatization.
2. Create representation of document by calculating Term Frequency and Inverse Document Frequency.

Setup

In [1]:
```
import nltk
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('averaged_perceptron_tagger')
nltk.download('wordnet')
nltk.download('omw-1.4')
```

```
[nltk_data] Downloading package punkt to /home/pict/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to /home/pict/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /home/pict/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
[nltk_data] Downloading package wordnet to /home/pict/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package omw-1.4 to /home/pict/nltk_data...
[nltk_data]   Package omw-1.4 is already up-to-date!
```

Out[1]:
```
True
```

In [2]:
```
from nltk.tokenize import sent_tokenize, word_tokenize
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer, WordNetLemmatizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.metrics import pairwise_distances
from sklearn.metrics.pairwise import euclidean_distances
from scipy.spatial import distance
import pandas as pd
import numpy as np
```

Reading the data from the text file

In [3]:
```
with open('./paragraph.txt') as f:
    paragraph = f.read()
    paragraph = paragraph.lower()
```

In [4]:
```
paragraph
```

Out[4]:
```
'the european union said it had joined members of the council of the baltic sea state
(cbss) in suspending russia and belarus from the council\'s activities.\n\n"this decisi
on is a part of the european union\'s and like-minded partners response to russia\'s in
vasion of ukraine and the involvement of belarus in this unprovoked and unjustified agg
ression," it said on saturday.\n\nrussia declared a partial ceasefire on saturday to al
low humanitarian corridors out of the ukrainian cities of mariupol and volnovakha, russ
ia\'s defence ministry said.\n\nthe partial ceasefire will allow civilians to leave the
city during a five-hour period from saturday morning, the city authorities said. civili
ans will be allowed to leave mariupol between noon and 5 p.m. moscow time (0900 - 1400
gmt)\n\nwestern allies have moved to isolate russia\'s economy and financial system sin
ce its invasion of ukraine, including sanctioning its central bank and oligarchs who am
assed fortunes and political influence under vladimir putin.\nrussian president vladimi
r putin launched what he called a special military operation before dawn on february 2
4, ignoring western warnings and saying the "neo-nazis" ruling ukraine threatened russi
a\'s security. russia\'s assault is said to be the biggest on a european state since wo
rld war two and threatens to upend the continent\'s post-cold war order.\n'
```

Tokenization

Tokenization is the first step when working with language tasks, it simplifies the input data by splitting it into sentences or words, as per the requirement

In [5]:
```
# Sentence tokenization
sentence_tokens = sent_tokenize(paragraph)
```

In [6]:
```
print('Number of sentence tokens :', len(sentence_tokens))
print('Sentence tokens :', sentence_tokens)
```

```
Number of sentence tokens : 7
Sentence tokens : ["the european union said it had joined members of the council of the
baltic sea states (cbss) in suspending russia and belarus from the council's activitie
s.", '"this decision is a part of the european union\'s and like-minded partners respon
se to russia\'s invasion of ukraine and the involvement of belarus in this unprovoked a
nd unjustified aggression," it said on saturday.', "russia declared a partial ceasefire
on saturday to allow humanitarian corridors out of the ukrainian cities of mariupol and
volnovakha, russia's defence ministry said.", 'the partial ceasefire will allow civilia
ns to leave the city during a five-hour period from saturday morning, the city authorit
ies said.', "civilians will be allowed to leave mariupol between noon and 5 p.m. moscow
time (0900 - 1400 gmt)\n\nwestern allies have moved to isolate russia's economy and fin
ancial system since its invasion of ukraine, including sanctioning its central bank and
oligarchs who amassed fortunes and political influence under vladimir putin.", 'russian
president vladimir putin launched what he called a special military operation before da
wn on february 24, ignoring western warnings and saying the "neo-nazis" ruling ukraine
threatened russia\'s security.', "russia's assault is said to be the biggest on a europ
ean state since world war two and threatens to upend the continent's post-cold war orde
r."]
```

In [7]:
```
# Word tokenization
word_tokens = word_tokenize(paragraph)
```

In [8]:
```
print('Number of word tokens :', len(word_tokens))
print('Word tokens :', word_tokens)
```

```
Number of word tokens : 236
Word tokens : ['the', 'european', 'union', 'said', 'it', 'had', 'joined', 'members', 'o
f', 'the', 'council', 'of', 'the', 'baltic', 'sea', 'states', '(', 'cbss', ')', 'in',
'suspending', 'russia', 'and', 'belarus', 'from', 'the', 'council', "'s", 'activities',
'.', '``', 'this', 'decision', 'is', 'a', 'part', 'of', 'the', 'european', 'union',
"'s", 'and', 'like-minded', 'partners', 'response', 'to', 'russia', "'s", 'invasion',
'of', 'ukraine', 'and', 'the', 'involvement', 'of', 'belarus', 'in', 'this', 'unprovoke
d', 'and', 'unjustified', 'aggression', ',', "''", 'it', 'said', 'on', 'saturday', '.',
'russia', 'declared', 'a', 'partial', 'ceasefire', 'on', 'saturday', 'to', 'allow', 'hu
manitarian', 'corridors', 'out', 'of', 'the', 'ukrainian', 'cities', 'of', 'mariupol',
'and', 'volnovakha', ',', 'russia', "'s", 'defence', 'ministry', 'said', '.', 'the', 'p
artial', 'ceasefire', 'will', 'allow', 'civilians', 'to', 'leave', 'the', 'city', 'duri
ng', 'a', 'five-hour', 'period', 'from', 'saturday', 'morning', ',', 'the', 'city', 'au
thorities', 'said', '.', 'civilians', 'will', 'be', 'allowed', 'to', 'leave', 'mariupo
l', 'between', 'noon', 'and', '5', 'p.m.', 'moscow', 'time', '(', '0900', '-', '1400',
'gmt', ')', 'western', 'allies', 'have', 'moved', 'to', 'isolate', 'russia', "'s", 'eco
nomy', 'and', 'financial', 'system', 'since', 'its', 'invasion', 'of', 'ukraine', ',',
'including', 'sanctioning', 'its', 'central', 'bank', 'and', 'oligarchs', 'who', 'amass
ed', 'fortunes', 'and', 'political', 'influence', 'under', 'vladimir', 'putin', '.', 'r
ussian', 'president', 'vladimir', 'putin', 'launched', 'what', 'he', 'called', 'a', 'sp
ecial', 'military', 'operation', 'before', 'dawn', 'on', 'february', '24', ',', 'ignori
ng', 'western', 'warnings', 'and', 'saying', 'the', '``', 'neo-nazis', "''", 'ruling',
'ukraine', 'threatened', 'russia', "'s", 'security', '.', 'russia', "'s", 'assault', 'i
s', 'said', 'to', 'be', 'the', 'biggest', 'on', 'a', 'european', 'state', 'since', 'wor
ld', 'war', 'two', 'and', 'threatens', 'to', 'upend', 'the', 'continent', "'s", 'post-c
old', 'war', 'order', '.']
```

POS Tagging and Stop words removal

In [9]:
```python
stop_words = set(stopwords.words('english'))
print('Stop words :', stop_words)
```

```
Stop words : {'won', 'in', 'through', 'so', 'yourself', 'hadn', 'but', 'nor', "didn't
'only', 'o', 'herself', 'up', 'doing', 'any', 'most', 'with', 'by', 'were', 'me', 'ver
y', 'an', 'to', "you'd", "she's", 'again', 'don', 'itself', 'while', 'whom', 'll', "nee
dn't", 'ourselves', 'himself', 'it', 'themselves', 'was', "isn't", 'or', 'from', 'whic
h', 'why', 's', 'each', 'wouldn', 'between', 'wasn', 'm', 'been', 'a', 'have', 'where',
"weren't", "mightn't", 'do', 'into', 'under', 'too', 'my', 'you', 'he', "aren't", 'ma',
'what', 'your', 'our', 'can', 'i', "wasn't", 'some', 'as', 'out', 'until', 'that', "hav
en't", 'because', 'below', 'she', "you've", "hasn't", 'having', "mustn't", 'not', 'shou
ld', 'haven', 'had', 'who', 'will', 'has', 'them', "wouldn't", 're', 'both', 'be', 'hi
m', 'weren', 'ain', 'its', 'at', "that'll", 'shouldn', 've', "hadn't", 'against', "do
n't", 'down', 'how', 'other', "doesn't", 'same', 'didn', 'about', 'yourselves', "you'r
e", 'those', 'shan', 'couldn', 'above', "shouldn't", 'before', 'y', "couldn't", 'we',
'are', 'yours', 'hasn', 'is', 'then', 'for', 'does', 'if', 'during', 'ours', 'after',
'there', 'd', 'few', 'mustn', 'off', 'mightn', 'needn', 'now', 'than', 'such', 'when',
'these', 'here', 'her', 'his', "it's", 'over', 'their', "shan't", 'of', 'myself', 'furt
her', "should've", 'just', 'am', 'on', 'isn', "you'll", 'more', 'aren', 'this', 'they',
'being', "won't", 'once', 'no', 'the', 'doesn', 'own', 't', 'did', 'hers', 'and', 'thei
rs', 'all'}
```

In [10]:
```python
word_tokens = [word_token for word_token in word_tokens if word_token not in stop_w
```

In [11]:
```python
print('Filtered word tokens :', word_tokens)
```

```
Filtered word tokens : ['european', 'union', 'said', 'joined', 'members', 'council',
altic', 'sea', 'states', '(', 'cbss', ')', 'suspending', 'russia', 'belarus', 'counci
l', "'s", 'activities', '.', '``', 'decision', 'part', 'european', 'union', "'s", 'like
-minded', 'partners', 'response', 'russia', "'s", 'invasion', 'ukraine', 'involvement',
'belarus', 'unprovoked', 'unjustified', 'aggression', ',', "''", 'said', 'saturday',
'.', 'russia', 'declared', 'partial', 'ceasefire', 'saturday', 'allow', 'humanitarian',
'corridors', 'ukrainian', 'cities', 'mariupol', 'volnovakha', ',', 'russia', "'s", 'def
ence', 'ministry', 'said', '.', 'partial', 'ceasefire', 'allow', 'civilians', 'leave',
'city', 'five-hour', 'period', 'saturday', 'morning', ',', 'city', 'authorities', 'sai
d', '.', 'civilians', 'allowed', 'leave', 'mariupol', 'noon', '5', 'p.m.', 'moscow', 't
ime', '(', '0900', '-', '1400', 'gmt', ')', 'western', 'allies', 'moved', 'isolate', 'r
ussia', "'s", 'economy', 'financial', 'system', 'since', 'invasion', 'ukraine', ',', 'i
ncluding', 'sanctioning', 'central', 'bank', 'oligarchs', 'amassed', 'fortunes', 'polit
ical', 'influence', 'vladimir', 'putin', '.', 'russian', 'president', 'vladimir', 'puti
n', 'launched', 'called', 'special', 'military', 'operation', 'dawn', 'february', '24',
',', 'ignoring', 'western', 'warnings', 'saying', '``', 'neo-nazis', "''", 'ruling', 'u
kraine', 'threatened', 'russia', "'s", 'security', '.', 'russia', "'s", 'assault', 'sai
d', 'biggest', 'european', 'state', 'since', 'world', 'war', 'two', 'threatens', 'upen
d', 'continent', "'s", 'post-cold', 'war', 'order', '.']
```

In [12]:
```python
'''
CC coordinating conjunction
CD cardinal digit
DT determiner
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’
LS list marker 1)
MD modal – could, will
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
TO – to go ‘to’ the store.
UH interjection – errrrrrrrm
VB verb, base form – take
VBD verb, past tense – took
VBG verb, gerund/present participle – taking
VBN verb, past participle – taken
VBP verb, sing. present, non-3d – take
VBZ verb, 3rd person sing. present – takes
WDT wh-determiner – which
WP wh-pronoun – who, what
WP$ possessive wh-pronoun, eg- whose
WRB wh-abverb, eg- where, when
'''
tagged = nltk.pos_tag(word_tokens)
```

In [13]:

```
print('POS Tagged form of filtered word tokens :')
for tag in tagged:
    print(tag)
```

```
POS Tagged form of filtered word tokens :
('european', 'JJ')
('union', 'NN')
('said', 'VBD')
('joined', 'JJ')
('members', 'NNS')
('council', 'VBP')
('baltic', 'JJ')
('sea', 'NN')
('states', 'NNS')
('(', '(')
('cbss', 'NN')
(')', ')')
('suspending', 'VBG')
('russia', 'JJ')
('belarus', 'NN')
('council', 'NN')
("'s", 'POS')
('activities', 'NNS')
('.', '.')
('``', '``')
('decision', 'NN')
('part', 'NN')
('european', 'VBP')
('union', 'NN')
("'s", 'POS')
('like-minded', 'JJ')
('partners', 'NNS')
('response', 'NN')
('russia', 'NN')
("'s", 'POS')
```

```
('invasion', 'NN')
('ukraine', 'JJ')
('involvement', 'NN')
('belarus', 'NN')
('unprovoked', 'VBD')
('unjustified', 'JJ')
('aggression', 'NN')
(',', ',')
("''", "''")
('said', 'VBD')
('saturday', 'NN')
('.', '.')
('russia', 'NN')
('declared', 'VBD')
('partial', 'JJ')
('ceasefire', 'NN')
('saturday', 'NN')
('allow', 'VB')
('humanitarian', 'JJ')
('corridors', 'NNS')
('ukrainian', 'JJ')
('cities', 'NNS')
('mariupol', 'VBP')
('volnovakha', 'NN')
(',', ',')
('russia', 'NN')
("'s", 'POS')
('defence', 'NN')
('ministry', 'NN')
('said', 'VBD')
('.', '.')
('partial', 'JJ')
('ceasefire', 'NN')
('allow', 'NN')
('civilians', 'NNS')
('leave', 'VBP')
('city', 'NN')
('five-hour', 'JJ')
('period', 'NN')
('saturday', 'JJ')
('morning', 'NN')
(',', ',')
('city', 'NN')
('authorities', 'NNS')
('said', 'VBD')
('.', '.')
('civilians', 'NNS')
('allowed', 'VBN')
('leave', 'VBP')
('mariupol', 'VBN')
('noon', 'RB')
('5', 'CD')
('p.m.', 'NN')
('moscow', 'NN')
('time', 'NN')
('(', '(')
('0900', 'CD')
('-', ':')
('1400', 'CD')
('gmt', 'NN')
(')', ')')
('western', 'JJ')
('allies', 'NNS')
('moved', 'VBD')
('isolate', 'JJ')
('russia', 'NN')
("'s", 'POS')
('economy', 'NN')
('financial', 'JJ')
```

```
('system', 'NN')
('since', 'IN')
('invasion', 'NN')
('ukraine', 'NN')
(',', ',')
('including', 'VBG')
('sanctioning', 'VBG')
('central', 'JJ')
('bank', 'NN')
('oligarchs', 'NNS')
('amassed', 'VBN')
('fortunes', 'NNS')
('political', 'JJ')
('influence', 'NN')
('vladimir', 'NN')
('putin', 'NN')
('.', '.')
('russian', 'JJ')
('president', 'NN')
('vladimir', 'NN')
('putin', 'NN')
('launched', 'VBN')
('called', 'VBN')
('special', 'JJ')
('military', 'JJ')
('operation', 'NN')
('dawn', 'VBD')
('february', 'JJ')
('24', 'CD')
(',', ',')
('ignoring', 'VBG')
('western', 'JJ')
('warnings', 'NNS')
('saying', 'VBG')
('``', '``')
('neo-nazis', 'JJ')
("''", "''")
('ruling', 'VBG')
('ukraine', 'JJ')
('threatened', 'VBN')
('russia', 'NN')
("'s", 'POS')
('security', 'NN')
('.', '.')
('russia', 'NN')
("'s", 'POS')
('assault', 'NN')
('said', 'VBD')
('biggest', 'JJS')
('european', 'JJ')
('state', 'NN')
('since', 'IN')
('world', 'NN')
('war', 'NN')
('two', 'CD')
('threatens', 'NNS')
('upend', 'VBP')
('continent', 'NN')
("'s", 'POS')
('post-cold', 'JJ')
('war', 'NN')
('order', 'NN')
('.', '.')
```

Stemming

In [14]:

```
ps = PorterStemmer()
```

In [15]:

```
print('Results of Stemming')
stemmed = {word: ps.stem(word) for word in word_tokens}
for pair in stemmed.items():
    print('{0} --> {1}'.format(pair[0], pair[1]))
```

```
Results of Stemming
european --> european
union --> union
said --> said
joined --> join
members --> member
council --> council
baltic --> baltic
sea --> sea
states --> state
( --> (
cbss --> cbss
) --> )
suspending --> suspend
russia --> russia
belarus --> belaru
's --> 's
activities --> activ
. --> .
`` --> ``
decision --> decis
part --> part
like-minded --> like-mind
partners --> partner
response --> respons
invasion --> invas
ukraine --> ukrain
involvement --> involv
unprovoked --> unprovok
unjustified --> unjustifi
aggression --> aggress
, --> ,
'' --> ''
saturday --> saturday
declared --> declar
partial --> partial
ceasefire --> ceasefir
allow --> allow
humanitarian --> humanitarian
corridors --> corridor
ukrainian --> ukrainian
cities --> citi
mariupol --> mariupol
volnovakha --> volnovakha
defence --> defenc
ministry --> ministri
civilians --> civilian
leave --> leav
city --> citi
five-hour --> five-hour
period --> period
morning --> morn
authorities --> author
allowed --> allow
noon --> noon
5 --> 5
p.m. --> p.m.
moscow --> moscow
time --> time
0900 --> 0900
```

```
- --> -
1400 --> 1400
gmt --> gmt
western --> western
allies --> alli
moved --> move
isolate --> isol
economy --> economi
financial --> financi
system --> system
since --> sinc
including --> includ
sanctioning --> sanction
central --> central
bank --> bank
oligarchs --> oligarch
amassed --> amass
fortunes --> fortun
political --> polit
influence --> influenc
vladimir --> vladimir
putin --> putin
russian --> russian
president --> presid
launched --> launch
called --> call
special --> special
military --> militari
operation --> oper
dawn --> dawn
february --> februari
24 --> 24
ignoring --> ignor
warnings --> warn
saying --> say
neo-nazis --> neo-nazi
ruling --> rule
threatened --> threaten
security --> secur
assault --> assault
biggest --> biggest
state --> state
world --> world
war --> war
two --> two
threatens --> threaten
upend --> upend
continent --> contin
post-cold --> post-cold
order --> order
```

Lemmatization

In [16]:

```
lemmatizer = WordNetLemmatizer()
```

In [17]:

```
print('Results of Lemmatization')
lemmatized = {word: lemmatizer.lemmatize(word) for word in word_tokens}
for pair in lemmatized.items():
    print('{0} --> {1}'.format(pair[0], pair[1]))
```

```
Results of Lemmatization
european --> european
union --> union
said --> said
joined --> joined
```

```
members --> member
council --> council
baltic --> baltic
sea --> sea
states --> state
( --> (
cbss --> cbss
) --> )
suspending --> suspending
russia --> russia
belarus --> belarus
's --> 's
activities --> activity
. --> .
`` --> ``
decision --> decision
part --> part
like-minded --> like-minded
partners --> partner
response --> response
invasion --> invasion
ukraine --> ukraine
involvement --> involvement
unprovoked --> unprovoked
unjustified --> unjustified
aggression --> aggression
, --> ,
'' --> ''
saturday --> saturday
declared --> declared
partial --> partial
ceasefire --> ceasefire
allow --> allow
humanitarian --> humanitarian
corridors --> corridor
ukrainian --> ukrainian
cities --> city
mariupol --> mariupol
volnovakha --> volnovakha
defence --> defence
ministry --> ministry
civilians --> civilian
leave --> leave
city --> city
five-hour --> five-hour
period --> period
morning --> morning
authorities --> authority
allowed --> allowed
noon --> noon
5 --> 5
p.m. --> p.m.
moscow --> moscow
time --> time
0900 --> 0900
- --> -
1400 --> 1400
gmt --> gmt
western --> western
allies --> ally
moved --> moved
isolate --> isolate
economy --> economy
financial --> financial
system --> system
since --> since
including --> including
sanctioning --> sanctioning
central --> central
```

```
bank --> bank
oligarchs --> oligarch
amassed --> amassed
fortunes --> fortune
political --> political
influence --> influence
vladimir --> vladimir
putin --> putin
russian --> russian
president --> president
launched --> launched
called --> called
special --> special
military --> military
operation --> operation
dawn --> dawn
february --> february
24 --> 24
ignoring --> ignoring
warnings --> warning
saying --> saying
neo-nazis --> neo-nazis
ruling --> ruling
threatened --> threatened
security --> security
assault --> assault
biggest --> biggest
state --> state
world --> world
war --> war
two --> two
threatens --> threatens
upend --> upend
continent --> continent
post-cold --> post-cold
order --> order
```

Term-Frequency and Inverse Document Frequency

In [18]:

```python
def arr_convert_1d(arr):
    arr = np.array(arr)
    arr = np.concatenate( arr, axis=0 )
    arr = np.concatenate( arr, axis=0 )
    return arr
```

In [19]:

```python
cos = []
def cosine(trans):
    cos.append(cosine_similarity(trans[0], trans[1]))
```

In [20]:

```python
manhatten = []
def manhatten_distance(trans):
    manhatten.append(pairwise_distances(trans[0], trans[1], metric = 'manhattan'))
```

In [21]:

```python
euclidean = []
def euclidean_function(vectors):
    euc=euclidean_distances(vectors[0], vectors[1])
    euclidean.append(euc)
```

In [22]:

```python
def tfidf(str1, str2):
    vect = TfidfVectorizer()
    vect.fit(word_tokens)
    corpus = [str1,str2]
    trans = vect.transform(corpus)
    euclidean_function(trans)
    cosine(trans)
    manhatten_distance(trans)
    return convert()
```

In [23]:

```
def convert():
    dataf = pd.DataFrame()
    lis2 = arr_convert_1d(manhatten)
    dataf['manhatten'] = lis2
    lis2 = arr_convert_1d(cos)
    dataf['cos_sim'] = lis2
    lis2 = arr_convert_1d(euclidean)
    dataf['euclidean'] = lis2
    return dataf
```

In [24]:

```
str1 = 'russia'
str2 = 'ukraine'
newData = tfidf(str1,str2);
print(newData);
```

```
   manhatten  cos_sim  euclidean
0        2.0      0.0   1.414214
```