20BCE204 IRS Practical 2

Consider a static corpus of 10 documents from a political news domain. Implement a Boolean retrieval model by considering the term document incidence matrix. Your implemented Boolean model should retrieve the required document/documents by processing AND & OR queries. Display sparsity value of term-document incidence matrix.

In [1]:
```
pol1 = open("pol1.txt")

pol1_str=pol1.read()

pol1_str
```

Out[1]:
```
'India's ruling Bharatiya Janata Party (BJP) scored a landslide election wi
n in Prime Minister Narendra Modi's home state of Gujarat on Thursday, in a
boost for the leader and his Hindu-nationalist party ahead of the 2024 gene
ral election.The BJP took 156 of 182 total seats in the Gujarat state assem
bly following voting earlier this month, marking the party's best-ever perf
ormance in the state, a longtime BJP stronghold.The main opposition Indian
National Congress won 17 seats, while the Aam Aadmi Party (AAP) took five.
'
```

In [2]:
```
pol=[]

for i in range(10):
    poly=open(f"pol{i+1}.txt",encoding="utf8")
    pol.append(poly.read())


print(len(pol))
print(pol[1])
```

```
10
When world leaders at the Group of 20 summit in Bali, Indonesia, issued a j
oint statement condemning Russia's war in Ukraine, a familiar sentence stoo
d out from the 1,186-page document."Today's era must not be of war," it sai
d, echoing what Indian Prime Minister Narendra Modi told Russian leader Vla
dimir Putin during a face-to-face meeting in September.Media and officials
in the country of 1.3 billion were quick to claim the inclusion as a sign t
hat the world's largest democracy had played a vital role in bridging diffe
rences between an increasingly isolated Russia, and the United States and i
ts allies."How India united G20 on PM Modi's idea of peace," ran a headline
in the Times of India, the country's largest English-language paper. "The P
rime Minister's message that this is not the era of war… resonated very dee
ply across all the delegations and helped bridge the gap across different p
arties," India's Foreign Secretary Vinay Kwatra told reporters Wednesday.
```

In [3]:
```
import nltk
# nltk.download('punkt')
```

In [4]:

```
import nltk
import numpy as np

nltkpol=[]

for i in range(10):
    nltk_pol=nltk.word_tokenize(pol[i])
    nltkpol.append(nltk_pol)

print(nltkpol[2])

# nltkpol = list(np.concatenate(nltkpol))
# print(nltkpol)
```

```
['The', 'declaration', 'came', 'as', 'Indonesian', 'President', 'Joko', 'Wi
dodo', 'handed', 'over', 'the', 'G20', 'presidency', 'to', 'Modi', ',', 'wh
o', 'will', 'host', 'the', 'next', 'leaders', '’', 'summit', 'in', 'the', '
Indian', 'capital', 'New', 'Delhi', 'in', 'September', '2023', '—', 'about'
, 'six', 'months', 'before', 'he', 'is', 'expected', 'to', 'head', 'to', 't
he', 'polls', 'in', 'a', 'general', 'election', 'and', 'contest', 'the', 'c
ountry', '’', 's', 'top', 'seat', 'for', 'a', 'third', 'time', '.', 'As', '
New', 'Delhi', 'deftly', 'balances', 'its', 'ties', 'to', 'Russia', 'and',
'the', 'West', ',', 'Modi', ',', 'analysts', 'say', ',', 'is', 'emerging',
'as', 'a', 'leader', 'who', 'has', 'been', 'courted', 'by', 'all', 'sides',
',', 'winning', 'him', 'support', 'at', 'home', ',', 'while', 'cementing',
'India', 'as', 'an', 'international', 'power', 'broker.', '“', 'The', 'dome
stic', 'narrative', 'is', 'that', 'the', 'G20', 'summit', 'is', 'being', 'u
sed', 'as', 'a', 'big', 'banner', 'in', 'Modi', '’', 's', 'election', 'camp
aign', 'to', 'show', 'he', '’', 's', 'a', 'great', 'global', 'statesmen', '
,', '”', 'said', 'Sushant', 'Singh', ',', 'a', 'senior', 'fellow', 'at', 'N
ew', 'Delhi-based', 'think', 'tank', 'Center', 'for', 'Policy', 'Research',
'.', '“', 'And', 'the', 'current', 'Indian', 'leadership', 'now', 'sees', '
themselves', 'as', 'a', 'powerful', 'country', 'seated', 'at', 'the', 'high
', 'table', '.', '”']
```

In [8]:

```
punpol=[]

for i in range(10):
    pun_pol=[c for c in nltkpol[i] if c.isalpha() ]
    punpol.append(pun_pol)
# fil_text= [w for w in nltkpol[0] if w not in spw_words]


print(punpol[1])
```

```
['When', 'world', 'leaders', 'at', 'the', 'Group', 'of', 'summit', 'in', 'B
ali', 'Indonesia', 'issued', 'a', 'joint', 'statement', 'condemning', 'Russ
ia', 's', 'war', 'in', 'Ukraine', 'a', 'familiar', 'sentence', 'stood', 'ou
t', 'from', 'the', 'Today', 's', 'era', 'must', 'not', 'be', 'of', 'war', '
it', 'said', 'echoing', 'what', 'Indian', 'Prime', 'Minister', 'Narendra',
'Modi', 'told', 'Russian', 'leader', 'Vladimir', 'Putin', 'during', 'a', 'm
eeting', 'in', 'and', 'officials', 'in', 'the', 'country', 'of', 'billion',
'were', 'quick', 'to', 'claim', 'the', 'inclusion', 'as', 'a', 'sign', 'tha
t', 'the', 'world', 's', 'largest', 'democracy', 'had', 'played', 'a', 'vit
al', 'role', 'in', 'bridging', 'differences', 'between', 'an', 'increasingl
y', 'isolated', 'Russia', 'and', 'the', 'United', 'States', 'and', 'its', '
How', 'India', 'united', 'on', 'PM', 'Modi', 's', 'idea', 'of', 'peace', 'r
an', 'a', 'headline', 'in', 'the', 'Times', 'of', 'India', 'the', 'country'
, 's', 'largest', 'paper', 'The', 'Prime', 'Minister', 's', 'message', 'tha
t', 'this', 'is', 'not', 'the', 'era', 'of', 'resonated', 'very', 'deeply',
'across', 'all', 'the', 'delegations', 'and', 'helped', 'bridge', 'the', 'g
ap', 'across', 'different', 'parties', 'India', 's', 'Foreign', 'Secretary'
, 'Vinay', 'Kwatra', 'told', 'reporters', 'Wednesday']
```

In [9]:

```
# nltk.download()
```

In [10]:

```
from nltk.corpus import stopwords as spw
spw_words=spw.words('english')

filtpol=[]

for i in range(10):
    filt_pol=[w for w in punpol[i] if w not in spw_words]
    filtpol.append(filt_pol)
# fil_text= [w for w in nltkpol[0] if w not in spw_words]


print(filtpol[1])
```

```
['When', 'world', 'leaders', 'Group', 'summit', 'Bali', 'Indonesia', 'issue
d', 'joint', 'statement', 'condemning', 'Russia', 'war', 'Ukraine', 'famili
ar', 'sentence', 'stood', 'Today', 'era', 'must', 'war', 'said', 'echoing',
'Indian', 'Prime', 'Minister', 'Narendra', 'Modi', 'told', 'Russian', 'lead
er', 'Vladimir', 'Putin', 'meeting', 'officials', 'country', 'billion', 'qu
ick', 'claim', 'inclusion', 'sign', 'world', 'largest', 'democracy', 'playe
d', 'vital', 'role', 'bridging', 'differences', 'increasingly', 'isolated',
'Russia', 'United', 'States', 'How', 'India', 'united', 'PM', 'Modi', 'idea
', 'peace', 'ran', 'headline', 'Times', 'India', 'country', 'largest', 'pap
er', 'The', 'Prime', 'Minister', 'message', 'era', 'resonated', 'deeply', '
across', 'delegations', 'helped', 'bridge', 'gap', 'across', 'different', '
parties', 'India', 'Foreign', 'Secretary', 'Vinay', 'Kwatra', 'told', 'repo
rters', 'Wednesday']
```

In [22]:

```
# from nltk.stem import WordNetLemmatizer
# lemmatizer= WordNetLemmatizer()




# finalpol=[]
# for i in range(10):
#     final=[]
#     for j in range (len(punpol[i])):
#         final_pol=lemmatizer.lemmatize(punpol[i][j])
#         final.append(final_pol)
#     finalpol.append(final)
# # fil_text= [w for w in nltkpol[0] if w not in spw_words]


# print(finalpol)

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

finalpol=[]
for i in range(10):
    final=[]
    for j in range (len(filtpol[i])):
            final_pol=ps.stem(filtpol[i][j])
            final.append(final_pol)
    finalpol.append(final)
print(finalpol[1])
```

```
['when', 'world', 'leader', 'group', 'summit', 'bali', 'indonesia', 'issu',
'joint', 'statement', 'condemn', 'russia', 'war', 'ukrain', 'familiar', 'se
ntenc', 'stood', 'today', 'era', 'must', 'war', 'said', 'echo', 'indian', '
prime', 'minist', 'narendra', 'modi', 'told', 'russian', 'leader', 'vladimi
r', 'putin', 'meet', 'offici', 'countri', 'billion', 'quick', 'claim', 'inc
lus', 'sign', 'world', 'largest', 'democraci', 'play', 'vital', 'role', 'br
idg', 'differ', 'increasingli', 'isol', 'russia', 'unit', 'state', 'how', '
india', 'unit', 'pm', 'modi', 'idea', 'peac', 'ran', 'headlin', 'time', 'in
dia', 'countri', 'largest', 'paper', 'the', 'prime', 'minist', 'messag', 'e
ra', 'reson', 'deepli', 'across', 'deleg', 'help', 'bridg', 'gap', 'across'
, 'differ', 'parti', 'india', 'foreign', 'secretari', 'vinay', 'kwatra', 't
old', 'report', 'wednesday']
```

In [12]:

```
strpol=[]

for i in range(10):

    str=""
    for j in range(len(finalpol[i])):
        str+=finalpol[i][j]
        str+=" "
    strpol.append(str)

print(strpol[1])

# res = [' '.join(ele) for ele in finalpol]
# res=str(res)
# print("The String of list is : " + str(res))
# len(res)
```

```
when world leader group summit bali indonesia issu joint statement condemn
russia war ukrain familiar sentenc stood today era must war said echo india
n prime minist narendra modi told russian leader vladimir putin meet offici
countri billion quick claim inclus sign world largest democraci play vital
role bridg differ increasingli isol russia unit state how india unit pm mod
i idea peac ran headlin time india countri largest paper the prime minist m
essag era reson deepli across deleg help bridg gap across differ parti indi
a foreign secretari vinay kwatra told report wednesday
```

In [13]:

```
from sklearn.feature_extraction.text import CountVectorizer
import pandas as pd

count = CountVectorizer()

matrix = count.fit_transform(strpol)
df = pd.DataFrame(matrix.toarray(), columns=count.get_feature_names())
```

In [14]:

```
df
```

Out[14]:

	aadmi	aam	aap	abdel	abvp	accord	accus	achiev	across	address	...	widodo	wi
0	1	1	1	0	0	0	0	0	0	0	...	0	
1	0	0	0	0	0	0	0	0	2	0	...	0	
2	0	0	0	0	0	0	0	0	0	0	...	1	
3	0	0	0	0	0	0	0	0	0	1	...	0	
4	0	0	0	0	2	0	1	0	0	0	...	0	
5	0	0	0	0	0	0	0	0	0	0	...	0	
6	0	0	0	1	0	0	0	0	0	0	...	0	
7	0	0	0	0	0	0	0	1	0	0	...	0	
8	0	0	0	0	0	1	0	1	0	0	...	0	
9	0	0	0	0	0	0	0	0	0	1	...	0	

10 rows × 504 columns

In [15]:
```
df_t = df.T
```

In [17]:
```
df_t.head()
```

Out[17]:

	0	1	2	3	4	5	6	7	8	9
aadmi	1	0	0	0	0	0	0	0	0	0
aam	1	0	0	0	0	0	0	0	0	0
aap	1	0	0	0	0	0	0	0	0	0
abdel	0	0	0	0	0	0	1	0	0	0
abvp	0	0	0	0	2	0	0	0	0	0

In [18]:
```
for i in range(len(df_t)):
    for j in range(10):
        if(df_t.iloc[i,j]>=1):
            df_t.iloc[i,j]=1

df_t
```

Out[18]:

	0	1	2	3	4	5	6	7	8	9
aadmi	1	0	0	0	0	0	0	0	0	0
aam	1	0	0	0	0	0	0	0	0	0
aap	1	0	0	0	0	0	0	0	0	0
abdel	0	0	0	0	0	0	1	0	0	0
abvp	0	0	0	0	1	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...
women	0	0	0	0	1	0	0	0	0	0
work	0	0	0	0	0	0	0	0	0	1
world	0	1	0	1	0	0	0	0	0	0
would	0	0	0	0	0	0	0	0	0	1
year	0	0	0	0	0	1	1	1	0	0

504 rows × 10 columns

In [25]:

```
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer


final = [True,True,True,True,True,True,True,True,True,True]
ip = input("Enter query: ")
words = ip.split(" ")

stop = set(stopwords.words('english'))
s=[]
for i in range(len(words)):
    #words = ip[i].split(" ")
    if words[i] not in stop:
        #s = s + words[i] + " "
        s.append(words[i])

new_ip=s

#print(len(new_ip))
#new_words=new_ip.split(" ")
s=[]
for i in range(len(new_ip)):
    #words = ip[i].split(" ")
    root = ps.stem(new_ip[i])
    s.append(root)

new_ip=s

#print(len(new_ip))
words = new_ip
#print(len(words))
#print(df_t.T.columns.values)
for i in range(len(words)):
    if words[i] not in df_t.T.columns.values:
        final = [False,False,False,False,False,False,False,False,False,Fals
        break
    for j in range(10):
        if df_t.loc[words[i],j]!=1:
            final[j]=False


flag=0

for i in range(len(final)):
    if final[i]==True:
        flag=1
if flag==0:
    print("No such file found!")
else:
    for i in range(len(final)):
        if final[i]==True:
            print(i)
            print(pol[i])
```