IR ASSIGNMENT 1

Janak Kapuriya (MT22032)
Kirtirajsinh Mahida(MT22104)
Mansi Patel(MT22109)

Question 1

Preprocessing

- Used OS library to iterate over each files in folder.
- Opened 1400 file and performed various operations on the.
- Used NLPK and RE for data preprocessing
- Used re.fiindall() function to find string between <title> </title> and <text> </text>.
- After that used python lower() function to lowercase the text.
- Used nltk.word_tokenize() function to convert text to tokens.
- For removing the stop words we have used NLTK library.
- After that we have removed all punctuation using str.maketrans() function in python
- Expression re.sub(r"\s+", " ", b, flags=re.UNICODE) used to remove all blank spaces.

Assumption

- User will not enter a query in which word are not present in 1400 files.

Results

- This is how data looks like when it's in the file:

```
Before Operations

File  1


<DOC>
<DOCNO>
1
</DOCNO>
<TITLE>
experimental investigation of the aerodynamics of a
wing in a slipstream .
</TITLE>
<AUTHOR>
brenckman,m.
</AUTHOR>
<BIBLIO>
j. ae. scs. 25, 1958, 324.
</BIBLIO>
<TEXT>
  an experimental study of a wing in a propeller slipstream was
made in order to determine the spanwise distribution of the lift
increase due to slipstream at different angles of attack of the wing
and at different free stream to slipstream velocity ratios .  the
results were intended in part as an evaluation basis for different
theoretical treatments of this problem .
  the comparative span loading curves, together with supporting
evidence, showed that a substantial part of the lift increment
produced by the slipstream was due to a /destalling/ or boundary-layer-control
effect .  the integrated remaining lift increment,
after subtracting this destalling lift, was found to agree
well with a potential flow theory .
  an empirical evaluation of the destalling effects was made for
the specific configuration of the experiment .
</TEXT>
</DOC>
```

- **Now we convert all text into lower case so after performing that operation we get this data**

```
File  1


experimental investigation aerodynamics wing slipstream experimental study wing pr
opeller slipstream made order determine spanwise distribution lift increase due sl
ipstream different angles attack wing different free stream slipstream velocity ra
tios results intended part evaluation basis different theoretical treatments probl
em comparative span loading curves together supporting evidence showed substantial
 part lift increment produced slipstream due destalling boundarylayercontrol effec
t integrated remaining lift increment subtracting destalling lift found agree well
 potential flow theory empirical evaluation destalling effects made specific confi
guration experiment
```

- **After that we perform tokenization and convert string into tokens**

```
File  1


['experimental', 'investigation', 'aerodynamics', 'wing', 'slipstream', 'experimental'
, 'study', 'wing', 'propeller', 'slipstream', 'made', 'order', 'determine', 'spanwise'
, 'distribution', 'lift', 'increase', 'due', 'slipstream', 'different', 'angles', 'att
ack', 'wing', 'different', 'free', 'stream', 'slipstream', 'velocity', 'ratios', 'resu
lts', 'intended', 'part', 'evaluation', 'basis', 'different', 'theoretical', 'treatmen
ts', 'problem', 'comparative', 'span', 'loading', 'curves', 'together', 'supporting',
'evidence', 'showed', 'substantial', 'part', 'lift', 'increment', 'produced', 'slipstr
eam', 'due', 'destalling', 'boundarylayercontrol', 'effect', 'integrated', 'remaining'
, 'lift', 'increment', 'subtracting', 'destalling', 'lift', 'found', 'agree', 'well',
'potential', 'flow', 'theory', 'empirical', 'evaluation', 'destalling', 'effects', 'ma
de', 'specific', 'configuration', 'experiment']
```

- **Then we remove stopwords**

```
After Removing Stop Words


File  1


experimental investigation aerodynamics wing slipstream experimental study wing propeller slip
stream made order determine spanwise distribution lift increase due slipstream different angle
s attack wing different free stream slipstream velocity ratios results intended part evaluatio
n basis different theoretical treatments problem comparative span loading curves together supp
orting evidence showed substantial part lift increment produced slipstream due destalling boun
darylayercontrol effect integrated remaining lift increment subtracting destalling lift found
agree well potential flow theory empirical evaluation destalling effects made specific configu
ration experiment
```

- **Remove Punctuations**

```
After Removing Punctuations


File  1


experimental investigation aerodynamics wing slipstream experimental study wing propeller slip
stream made order determine spanwise distribution lift increase due slipstream different angle
s attack wing different free stream slipstream velocity ratios results intended part evaluatio
n basis different theoretical treatments problem comparative span loading curves together supp
orting evidence showed substantial part lift increment produced slipstream due destalling boun
darylayercontrol effect integrated remaining lift increment subtracting destalling lift found
agree well potential flow theory empirical evaluation destalling effects made specific configu
ration experiment
```

- **Remove blank spaces**

```
After Removing blank space


File  1


experimental investigation aerodynamics wing slipstream experimental study wing propeller slip
stream made order determine spanwise distribution lift increase due slipstream different angle
s attack wing different free stream slipstream velocity ratios results intended part evaluatio
n basis different theoretical treatments problem comparative span loading curves together supp
orting evidence showed substantial part lift increment produced slipstream due destalling boun
darylayercontrol effect integrated remaining lift increment subtracting destalling lift found
agree well potential flow theory empirical evaluation destalling effects made specific configu
ration experiment
```

- **So after performing this task we got 1 data which looks like this and this process is called as a preprocessing of data.**

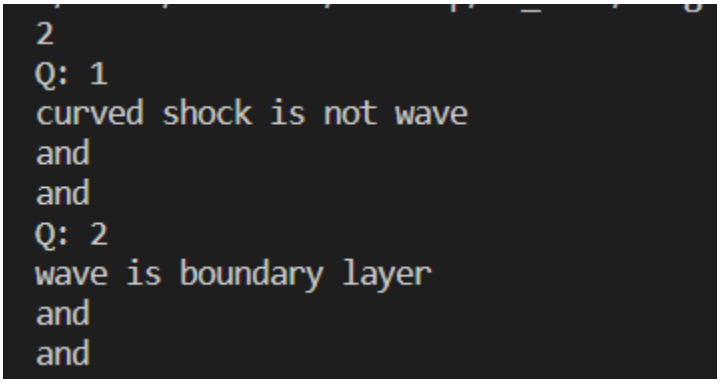
```
File  1


experimental investigation aerodynamics wing slipstream experimental study wing propeller slip
stream made order determine spanwise distribution lift increase due slipstream different angle
s attack wing different free stream slipstream velocity ratios results intended part evaluatio
n basis different theoretical treatments problem comparative span loading curves together supp
orting evidence showed substantial part lift increment produced slipstream due destalling boun
darylayercontrol effect integrated remaining lift increment subtracting destalling lift found
agree well potential flow theory empirical evaluation destalling effects made specific configu
ration experiment
```

Question 2

Methodology

- We need to design a algorithm that will Create a unigram inverted index of the dataset which we obtained in preprocessing task
- Used python language and pickle module for storing or loading the data file which is obtained after preprocessing so we don't need to apply preprocessing every time .
- After that we need to perform some given query on the data.
- For, query we build the logic of OR , AND , OR NOT , AND NOT of the 2 lists.
- Also we keep the count value to get how many comparisons that are done to perform all the operations.
- This is the input of query:



- Output of the code:

```
Query 1 :  curved and shock and wave
Number of documents retrieved for query 1:  5
Names of documents retrieved for query 1:  ['cranfield0002', 'cranfield0334', 'cranfield0401',
 'cranfield1225', 'cranfield1307']
Number of comparisons required for query 1:  365
Query 2 :  wave and boundary and layer
Number of documents retrieved for query 2:  34
Names of documents retrieved for query 2:  ['cranfield0002', 'cranfield0071', 'cranfield0072',
 'cranfield0170', 'cranfield0192', 'cranfield0207', 'cranfield0209', 'cranfield0256', 'cranfie
ld0308', 'cranfield0309', 'cranfield0329', 'cranfield0334', 'cranfield0417', 'cranfield0439',
'cranfield0504', 'cranfield0568', 'cranfield0569', 'cranfield0798', 'cranfield0974', 'cranfiel
d0976', 'cranfield1107', 'cranfield1157', 'cranfield1212', 'cranfield1220', 'cranfield1225', '
cranfield1228', 'cranfield1257', 'cranfield1274', 'cranfield1300', 'cranfield1307', 'cranfield
1310', 'cranfield1313', 'cranfield1319', 'cranfield1364']
Number of comparisons required for query 2:  832
```

Question 3

Methodology for Bigram inverted index

- Used dictionary of dictionary to store the bigram inverted index.
- First iterate over each preprocessed file and read the content of each file and convert it into tokens.
- Now each two tokens considered as term if it is an unique bigram words pair.
- Dictionary have term as key and value of each term is dictionary itself that keep track of document frequency and list of document id's.
- Resultant bigram invderted index is looks like below.
- I have used pickle library for storing Bigram inverted index.


```
'study wing': {'doc_freq': 1, 'docs': ['cranfield0001']},
'wing propeller': {'doc_freq': 1, 'docs': ['cranfield0001']},
'propeller slipstream': {'doc_freq': 5,
 'docs': ['cranfield0001',
  'cranfield0453',
  'cranfield1064',
  'cranfield1094',
  'cranfield1164']},
'slipstream made': {'doc_freq': 1, 'docs': ['cranfield0001']},
'made order': {'doc_freq': 2, 'docs': ['cranfield0001', 'cranfield0222']},
```

Methodology for Positional Index

- We have used nested dictionary for storing positional index.
- Each unique tokens of files are considered as term in positional index.
- For each term in index we have dictionary as value.
- After that we store in which documnet and in which position that term has occurred are stored inside the dictionary of term.
- Output of Positional index is given below

```
{'lift': {'doc_freq': 1,
  'docs': {'cranfield0001': [15, 48, 58, 62],
   'cranfield0069': [46],
   'cranfield0086': [51, 58, 60],
   'cranfield0141': [30],
   'cranfield0146': [90, 97],
   'cranfield0147': [10, 18],
   'cranfield0163': [127, 148],
   'cranfield0164': [41],
   'cranfield0200': [11],
   'cranfield0203': [23, 28],
   'cranfield0204': [64],
   'cranfield0206': [87, 101, 133, 144],
   'cranfield0225': [98, 133, 157],
   'cranfield0226': [7, 27],
   'cranfield0229': [101],
   'cranfield0230': [85, 97],
   'cranfield0234': [9, 18],
   'cranfield0235': [70],
```

Methodology of Phrase Query Retrieval

- Given the user input query break its into an tokens.

- Now for bigram inverted index phrase query for each bigram check in which docs that bigram occurs and take the intersection of all the docs retrieved from all bigrams.
- We have used document intersection function to find common documents between two tokens.
- Intersection of docs will give result where each bigrams are encounters.

- For positional index we check one more level deep than bigram inverted index.
- After we get all documents in which all tokens are encounter then each time we do position wise comparison of 2 tokens.
- If two tokens comes one after the other then difference between the two tokens position is 1. So we put that position of second token in temporary list.
- And compare that temporary list with next tokens positions.
- At the end if temporary list is not empty means we got a document in which all the tokens are present one after the another.
- So that doc will be in our answer.

Assumption

- User will enter the length of query length is less than or equls to 5.

Q3 output

```
Enter number of queries to excecute : 3
Enter Query 1 : wave boundary layer
Enter Query 2 : curved shock wave
Enter Query 3 : slipstream experimental investigation

Number of documents retrieved for query 1 using bigram inverted index: 7
Names of documents retrieved for query 1 using bigram inverted index: cranfield0002, cranfield0170, cranfield0256, cranfield030
8, cranfield0309, cranfield0569, cranfield1157
Number of documents retrieved for query 1 using positional index: 7
Names of documents retrieved for query 1 using positional index: cranfield0002, cranfield0170, cranfield0256, cranfield0308, cra
nfield0309, cranfield0569, cranfield1157

Number of documents retrieved for query 2 using bigram inverted index: 2
Names of documents retrieved for query 2 using bigram inverted index: cranfield0002, cranfield0401
Number of documents retrieved for query 2 using positional index: 1
Names of documents retrieved for query 2 using positional index: cranfield0002

Number of documents retrieved for query 3 using bigram inverted index: 1
Names of documents retrieved for query 3 using bigram inverted index: cranfield0001
Number of documents retrieved for query 3 using positional index: 0
Names of documents retrieved for query 3 using positional index: None
```

Comparison of Bigram Inverted Index and Positional Index

- From result of phrase retrieval queries we can see that bigram inverted index gives fasle positive in result.
- **Example** :
- doc1 = [“ I will go to school. How you will manage your time”]
- User query : **I will manage**
- Now **bigram inverted index** returns doc1 as final answer because bigram “I will” present in starting of text in doc1 and bigram “will manage” also present in second sentence of doc1.
- That will result in False Positive.

- **Positional index** will not return Doc1 in final result as it will check the position difference between the two bigram if its 1 then it will add it to the final answer. So positional index answer in this exampe is empty list of docs.{}