Bachelor of Technology (Computer Science & Engineering)

CSD358: Information Retrieval, Monsoon 2024

Assignment-1

DEVANGI JOSHI- 2110110183
KHUSHI GOYAL-2110110286

Corpus- 40 documents in a folder named “documents”

1. Write the program to construct an Inverted Index for a given document collection comprising of at least 40 documents with a total vocabulary size of at least 500 words. The program should take the input as boolean search queries using AND, OR and NOT operators and return the list of the documents satisfying the query need. While generating the tokens do case folding, normalization, stop word removal and lemmatization/stemming. (4)

Installing required libraries and appending it into the path of the directory being used

```
pip install nltk
[2] ✓ 2.0s                                                    Python
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: nltk in c:\users\devan\appdata\roaming\python\python312\site-packages (3.9.1)
Requirement already satisfied: click in c:\users\devan\appdata\roaming\python\python312\site-packages (from nltk) (8.1.7)
Requirement already satisfied: joblib in c:\users\devan\appdata\roaming\python\python312\site-packages (from nltk) (1.4.2)
Requirement already satisfied: regex>=2021.8.3 in c:\users\devan\appdata\roaming\python\python312\site-packages (from nltk) (2024.9.11)
Requirement already satisfied: tqdm in c:\users\devan\appdata\roaming\python\python312\site-packages (from nltk) (4.66.5)
Requirement already satisfied: colorama in c:\users\devan\appdata\roaming\python\python312\site-packages (from click->nltk) (0.4.6)
Note: you may need to restart the kernel to use updated packages.

pip install --upgrade nltk
[3] ✓ 2.3s                                                    Python
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: nltk in c:\users\devan\appdata\roaming\python\python312\site-packages (3.9.1)
Requirement already satisfied: click in c:\users\devan\appdata\roaming\python\python312\site-packages (from nltk) (8.1.7)
Requirement already satisfied: joblib in c:\users\devan\appdata\roaming\python\python312\site-packages (from nltk) (1.4.2)
Requirement already satisfied: regex>=2021.8.3 in c:\users\devan\appdata\roaming\python\python312\site-packages (from nltk) (2024.9.11)
Requirement already satisfied: tqdm in c:\users\devan\appdata\roaming\python\python312\site-packages (from nltk) (4.66.5)
Requirement already satisfied: colorama in c:\users\devan\appdata\roaming\python\python312\site-packages (from click->nltk) (0.4.6)
Note: you may need to restart the kernel to use updated packages.

nltk.download('punkt', download_dir='C:\\Users\\devan\\OneDrive\\Desktop\\Assignment_IR')
[4] ✓ 0.1s                                                    Python
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\devan\OneDrive\Desktop\Assignment_IR...
[nltk_data]   Package punkt is already up-to-date!
```

```
import nltk

nltk.download('stopwords')
from nltk.corpus import stopwords

print(stopwords.words('english'))
✓ 0.0s                                                        Python
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves',
[nltk_data] Downloading package stopwords to

import nltk
                                                              Python

import os
import re
import nltk
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer, WordNetLemmatizer
from collections import defaultdict
from nltk.tokenize import word_tokenize
from nltk.corpus import wordnet
                                                              Python

nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
                                                              Python
[nltk_data] Downloading package punkt to
```

Printing the inverted index as a part of the debugging process, to check that the pre-processing and document-parsing is executed properly.

```
Inverted Index:
adob: ['Adobe']
compani: ['Adobe', 'Amazon', 'Binance', 'Dell', 'HP', 'Lenovo', 'Ola', 'Uber', 'apple', 'blackberry', 'flipkart', 'google', 'huawei', 'levis', 'messeng
found: ['Adobe', 'Binance', 'Dell', 'HP', 'Lenovo', 'Uber', 'apple', 'blackberry', 'huawei', 'microsoft', 'motorola', 'nokia', 'puma', 'reliance', 'sam
1982: ['Adobe', 'HP', 'nokia', 'sony']
john: ['Adobe', 'shakespeare']
warnock: ['Adobe']
charl: ['Adobe', 'shakespeare']
geschk: ['Adobe']
employ: ['Adobe', 'HP', 'Ola', 'Uber', 'apple', 'messenger', 'puma', 'steam', 'telegram']
xerox: ['Adobe']
corpor: ['Adobe', 'Dell', 'HP', 'Lenovo', 'Ola', 'blackberry', 'microsoft', 'motorola', 'nike', 'nokia', 'paypal', 'puma', 'reliance', 'samsung', 'sony
palo: ['Adobe']
alto: ['Adobe']
california: ['Adobe', 'HP', 'apple', 'google']
research: ['Adobe', 'Amazon', 'Dell', 'Ola', 'bing', 'blackberry', 'google', 'huawei', 'microsoft', 'samsung', 'steam']
center: ['Adobe', 'HP', 'bing', 'huawei', 'nike']
parc: ['Adobe']
two: ['Adobe', 'Amazon', 'Binance', 'HP', 'Lenovo', 'Ola', 'blackberry', 'flipkart', 'levis', 'messenger', 'microsoft', 'motorola', 'nike', 'paypal', '
comput: ['Adobe', 'Amazon', 'Dell', 'Discord', 'HP', 'Lenovo', 'Ola', 'Uber', 'apple', 'canva', 'google', 'instagram', 'microsoft', 'motorola', 'operat
scientist: ['Adobe', 'Dell', 'Lenovo']
develop: ['Adobe', 'Amazon', 'Binance', 'Dell', 'HP', 'Ola', 'apple', 'bing', 'blackberry', 'google', 'huawei', 'levis', 'microsoft', 'motorola', 'nike
program: ['Adobe', 'Amazon', 'Ola', 'bing', 'google', 'huawei', 'microsoft', 'motorola', 'operating', 'reddit', 'reliance', 'skype', 'sony', 'spotify',
languag: ['Adobe', 'google', 'microsoft', 'operating', 'reddit', 'shakespeare', 'steam', 'yahoo']
special: ['Adobe', 'Dell', 'apple', 'motorola', 'paypal', 'samsung', 'skype', 'telegram', 'zomato']
...
yelp: ['zomato']
seo: ['zomato']
```

Boolean model:

Q1:

```
technology and phone
Enter your boolean search query (use AND, OR, NOT): (Press 'Enter' to confirm or 'Escape' to cancel)
# Import necessary libraries
```

```
Documents matching the query: ['Adobe', 'Amazon', 'Binance', 'Dell', 'HP', 'Lenovo', 'Ola', 'Uber', 'apple', 'blackberry', 'google', 'huawei',
'instagram', 'messenger', 'microsoft', 'motorola', 'nike', 'nokia', 'operating', 'samsung', 'skype', 'sony', 'steam', 'telegram', 'whatsapp', 'zomato']
```

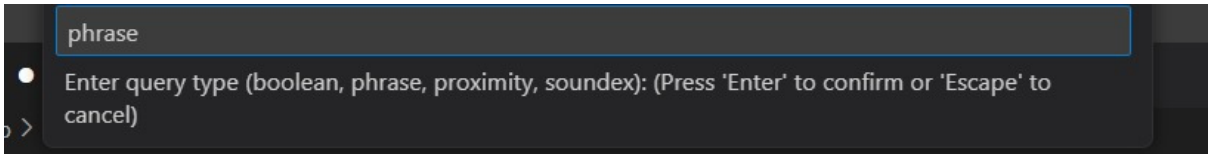
We have checked in the documents, all these documents contain both the words “technology” and “phone”

Q2:

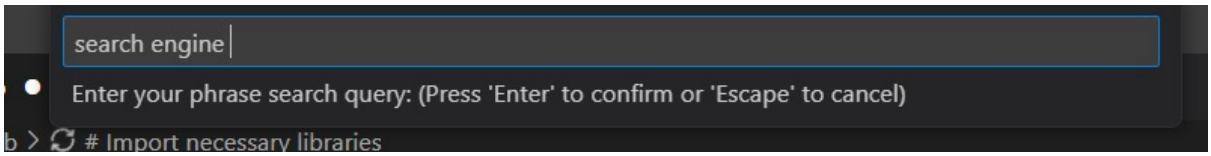
```
deliveries or foods
Enter your boolean search query (use AND, OR, NOT): (Press 'Enter' to confirm or 'Escape' to cancel)
```

```
Documents matching the query: ['blackberry', 'flipkart', 'google', 'swiggy', 'zomato']
```

Biword index:

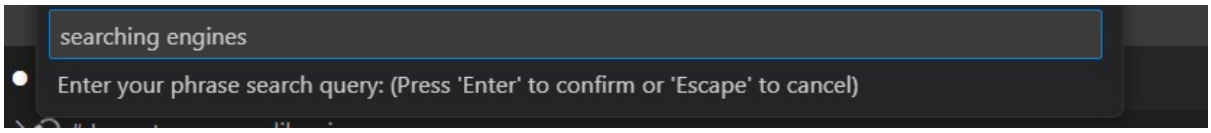


Q1:



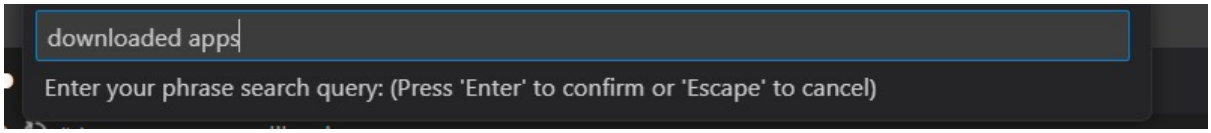

```
Documents matching the phrase query: ['bing', 'google', 'yahoo', 'zomato']
```

Q2:



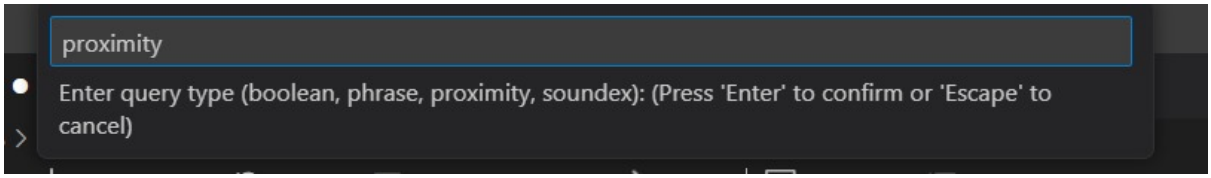

```
Documents matching the phrase query: ['bing', 'google', 'yahoo', 'zomato']
```

Q3:




```
Documents matching the phrase query: ['google', 'instagram', 'messenger', 'telegram']
```

Proximity:



Q1:

```
easy and passenger
Enter your proximity search query (two terms): (Press 'Enter' to confirm or 'Escape' to cancel)
```

```
10
Enter proximity distance (k): (Press 'Enter' to confirm or 'Escape' to cancel)
```

```
Documents matching the proximity query: ['Ola']
```

Q2:

```
graphic and capabilities
Enter your proximity search query (two terms): (Press 'Enter' to confirm or 'Escape' to cancel)
```

```
5
Enter proximity distance (k): (Press 'Enter' to confirm or 'Escape' to cancel)
```

```
Documents matching the proximity query: ['canva']
```

```
10
Enter proximity distance (k): (Press 'Enter' to confirm or 'Escape' to cancel)
```

```
Documents matching the proximity query: ['Adobe', 'canva']
```

<u>Soundex:</u>

If in our code (for Question 2) we choose soundex, we can enter 1 word, after which we have applied the functionality of soundex into the boolean option in question 2

```
boolean
Enter query type (boolean, phrase, proximity, soundex): (Press 'Enter' to confirm or 'Escape' to
cancel)
```

Q1:

```
lehri and stainford
Enter your boolean search query (use AND, OR, NOT): (Press 'Enter' to confirm or 'Escape' to cancel)
```

```
Documents matching the query: ['google']
```

Q2:

```
yahu and dauwnloads
Enter your boolean search query (use AND, OR, NOT): (Press 'Enter' to confirm or 'Escape' to cancel)
# Import necessary libraries
```

```
Documents matching the query: ['huawei', 'yahoo']
```

We have checked, both these documents satisfy the query condition.