Algorithms for Intelligence Web and Information Retrieval

Course Code: UE19CS332

Department of Computer Science Engineering

PES University

WEEK3:

SUBJECT: -POSITIONAL INDEX(WITH ALL OPERATION)

TEAM NUMBER: 18 TEAM MEMBERS:

KUNTAL GORAI	PES2UG19CS198
KIRAN B S	PES2UG19CS186
KOUSHIK P L	PES2UG19CS193
JEEVAN ANIL	PES2UG19CS166
HEAMANTH PRASAD HEGDE	PES2UG19CS148

CODE:

We are imported the libraries to work

week2modified.py > ...

```python
#-------------IMPORTING THE LIBRARIES---------------------
import numpy as np
import os
import nltk
from nltk.stem import PorterStemmer
from nltk.tokenize import TweetTokenizer
from natsort import natsorted
import string
```

We first the read the documents /dataset and implemented with the header and footer function

```python
#---------------READING THE CONTENTS OF FILE-----------------------
def read_file(filename):
    with open(filename, 'r', encoding ="ascii") as f:
        stuff = f.read()
    f.close()
    stuff = remove_header_footer(stuff)
    return stuff
```

We just removed the stopwords , punctions ,and we converted to all upper case letter to lower letter .

```python
#----------------PREPROCESSING LIKE REMOVAL OF PUNCTION,STOPWORDS ETC-------------------------
def preprocessing(final_string):
    tokenizer = TweetTokenizer()
    token_list = tokenizer.tokenize(final_string)

    # Remove punctuations.
    table = str.maketrans('', '', '\t')
    token_list = [word.translate(table) for word in token_list]
    punctuations = (string.punctuation).replace("'", "")
    trans_table = str.maketrans('', '', punctuations)
    stripped_words = [word.translate(trans_table) for word in token_list]
    token_list = [str for str in stripped_words if str]

    # Change to lowercase.
    token_list =[word.lower() for word in token_list]
    return token_list
```

This function helpful for Positional index here calculation

```
#--------------CALCULATING THE POSITIONAL INDEX OF A TERM IN FOLDERS--------------------
folder_names = ["soc.religion.christian"]
stemmer = PorterStemmer()
pos_index = {}
file_map = {}
for folder_name in folder_names:
    file_names = natsorted(os.listdir("20_newsgroups/" + folder_name))

    for fileno,file_name in enumerate(file_names):
        stuff = read_file("20_newsgroups/" + folder_name + "/" + file_name)
        final_token_list = preprocessing(stuff)
        # For position and term in the tokens.
        for pos, term in enumerate(final_token_list):
                    term = stemmer.stem(term)
                    # If term already exists in the positional index dictionary.
                    if term in pos_index:
                        # Increment total freq by 1.
                        pos_index[term][0] = pos_index[term][0] + 1
                        # Check if the term has existed in that DocID before.
                        if fileno in pos_index[term][1]:
                            pos_index[term][1][fileno].append(pos)

                        else:
                            pos_index[term][1][fileno] = [pos]

                    # If term does not exist in the positional index dictionary
                    # (first encounter).
                    else:

                        # Initialize the list.
                        pos_index[term] = []
                        # The total frequency is 1.
                        pos_index[term].append(1)
                        # The postings list is initially empty.
                        pos_index[term].append({})
                        # Add doc ID to postings list.
                        pos_index[term][1][fileno] = [pos]

        # Map the file no. to the file name.
        file_map[fileno] = "20_newsgroups/" + folder_name + "/" + file_name

        # Increment the file no. counter for document ID mapping
        fileno += 1
```

```
#-----------------PRINTING A POSITIONAL INDEX OF A TERM----------------------
sample_pos_idx = pos_index["christian"]
print("Positional Index")
print(sample_pos_idx)
```

```
#------------PRINTING THE FILE PATH AND RESPECTIVE POSITION OF THE OCCURANCE OF THE TERM--------------------
file_list = sample_pos_idx[1]
print("Filename, [Positions]")
for fileno, positions in file_list.items():
    print(file_map[fileno], positions)
```

OUTPUT:

```
PS C:\Users\Kuntal Gorai\OneDrive - PESUNIVERSITY\6th Semester\Algorithm for Web Infromation and Information Retrival\Week2> py
thon3 week2.py
Positional Index
[1415, {2: [71, 95, 105], 10: [13], 15: [65], 16: [85], 18: [15], 20: [34, 64], 23: [151], 24: [106, 247, 257, 277, 289, 311, 3
16, 417, 468], 27: [13, 18, 31, 60, 73, 118, 146, 277, 300, 691], 28: [114], 29: [161, 199, 227, 241, 277, 293, 319, 326, 411],
 30: [66, 215, 222], 34: [22, 63], 35: [49, 60, 85, 187, 204, 218], 36: [107, 128], 38: [214, 256, 397, 682, 697, 718], 40: [69
```

```
Filename, [Positions]
20_newsgroups/soc.religion.christian/20363 [71, 95, 105]
20_newsgroups/soc.religion.christian/20492 [13]
20_newsgroups/soc.religion.christian/20497 [65]
20_newsgroups/soc.religion.christian/20498 [85]
20_newsgroups/soc.religion.christian/20500 [15]
20_newsgroups/soc.religion.christian/20502 [34, 64]
20_newsgroups/soc.religion.christian/20505 [151]
20_newsgroups/soc.religion.christian/20506 [106, 247, 257, 277, 289, 311, 316, 417, 468]
20_newsgroups/soc.religion.christian/20509 [13, 18, 31, 60, 73, 118, 146, 277, 300, 691]
20_newsgroups/soc.religion.christian/20510 [114]
20_newsgroups/soc.religion.christian/20511 [161, 199, 227, 241, 277, 293, 319, 326, 411]
20_newsgroups/soc.religion.christian/20512 [66, 215, 222]
20_newsgroups/soc.religion.christian/20516 [22, 63]
20_newsgroups/soc.religion.christian/20517 [49, 60, 85, 187, 204, 218]
20_newsgroups/soc.religion.christian/20518 [107, 128]
```