Assignment 3

CS 5154 – Information Retrieval

# Results before changes

A computer screen capture

Description automatically generated with low confidence

# Results after changes

## Hardcoded to line 26813 to check verify results from previous run of code

A computer screen capture

Description automatically generated with low confidence

## Random Line of Text

A computer screen capture

Description automatically generated with low confidence

# Updated Code

# IR3C.py CS5154/6054 cheng 2022  
# read lines from a text file as docs  
# tokenize each as a set of words  
# make the inverted index  
# name those words with postings list longer than 1000 stopwords  
# remove stopwords from the sets representing docs  
# randomly select a doc as the query with at least 8 words  
# using the inverted index  
# retrieve sets of docs containing each of the word in query  
# update a dictionary called intersection   
# list docs with intersection to query > 3  
# Usage: python IR3C.py  
  
import re  
import random  
from collections import Counter  
  
f = open("bible.txt", "r")  
docs = f.readlines()  
f.close()  
invertedIndex = {}  
for i in range(len(docs)):  
 for s in set(re.findall('\w+', docs[i])):  
 if invertedIndex.get(s) == None:  
 invertedIndex.update({s : {i}})  
 else:  
 invertedIndex.get(s).add(i)  
  
stopwords = set()  
for k, v in invertedIndex.items():  
 if len(v) > 1000:  
 stopwords.add(k)  
  
N = len(docs)  
sets = list(map(lambda s: set(re.findall('\w+', s)) - stopwords, docs))  
for iter in range(12):  
 query = random.randint(0, N)  
 if len(sets[query]) > 8:  
 break  
  
print(query)  
print(docs[query])  
A = sets[query]  
print(A)  
intersections = Counter()  
for t in A: # the following six lines can be replaced with one using Counter  
 intersections.update(invertedIndex.get(t))  
  
for k, v in intersections.items():  
 if v > 3:  
 print(v, k, docs[k])