NIKITA VISPUTE
NETID: NXV170005
CS 6322.001

HOMEWORK 1

Problem 1: Tokenization

Output for Problem 1:

1. Number of tokens in the Cranfield Text Collection: **230865**

2. Number of unique words in the Cranfield Text Collection: **8858**

3. Number of words that occur only once in the Cranfield Text Collection: **3356**

4. 30 most frequent words in the Cranfield Text Collection:

	TOKEN	FREQUENCY
1.	the	19450
2.	of	12717
3.	and	6677
4.	a	6002
5.	in	4651
6.	to	4563
7.	is	4114
8.	for	3493
9.	are	2429
10.	with	2265
11.	on	1944
12.	flow	1849
13.	at	1834
14.	by	1756
15.	that	1570
16.	an	1388
17.	be	1272
18.	pressure	1207
19.	boundary	1156
20.	from	1116
21.	as	1114
22.	this	1081
23.	layer	1002
24.	which	975
25.	number	973
26.	results	885
27.	it	857
28.	mach	823
29.	theory	788
30.	shock	712

Total files tokenized: 1400

5. Average number of words per document: **164.9 ~ 165 words/document**

Program Description for Problem 1:

1. How long the program took to acquire the text characteristics.
Answer:
Time taken to tokenize the Cranfield Text Collection: 2.32 s (local machine).

2. How the program handles:

A. Upper and lower case words (e.g. "People", "people", "Apple", "apple");
Answer:
Entire text from the Cranfield collection is converted to lower case.
Therefore, People => people Apple => apple
people => people apple => apple

B. Words with dashes (e.g. "1996-97", "middle-class", "30-year", "tean-ager")
Answer:
All dashes/hyphens are removed and replaced with a space. All digits/numbers are removed.
Therefore, 1996-97 => __ middle-class => middle class
30-year => year tean-ager => tean ager

C. Possessives (e.g. "sheriff's", "university's")
Answer:
Possessives and words with apostrophes are removed.
Therefore, sheriff's => sheriff university's => university
churches' => churches o'clock => oclock

D. Acronyms (e.g., "U.S.", "U.N.")
Answer:
Periods, dots and other punctuations marks are removed.
Therefore, U.S. => US => us U.N. => UN => un

3. Algorithms and data structures used:

Tokenizer takes the Cranfield directory path as a command line input argument. It throws error if no argument is passed.

Preprocessing:

After retrieving the Cranfield directory path, each file is fetched and read one by one.

The preprocessing is the next step after reading all the file data.

1. The raw text is cleaned by removing the html tags
2. Special characters like "-/,():?;+^=%#&~$!@*_}{ " are replaced with space.
3. All digits/numbers are replaced with a space.
4. Punctuation marks like “.” are removed from the text.
5. Possessives/ contractions are removed from words having possessives/contractions.
6. Words ending with apostrophes have the apostrophes removed.
7. All extra spaces are replaced by a single space.
8. All the characters are changed to lower case.

Storing the tokens:

- The text collection is tokenized using split function.
- Tokens are stored in a hash map data structure and by calling the function map_book.
- map_book stores the tokens as key and frequency of occurrence of the token in the Cranfield Collection as its value.
- If the hash map already contains a particular token, then its frequency is incremented otherwise new key value for the token is created in the hash map and frequency is inputted as 1.
- Set is used to collect all the unique tokens occurring in the Collection.
- Counter is used to compute tokens having frequency only 1.
- Counters are used to compute total number of tokens as well as total files in the Cranfield Collection.
- Sorted hash map is used to find the 30 most frequent tokens in the Cranfield collection.
- The average number of tokens per document is computed by dividing the total number of tokens in the Cranfield Collection by the number of files in the Cranfield collection.

Libraries used:

a. glob: The glob module finds all the pathnames matching a specified pattern according to the rules used by the Unix shell.

b. re: The re module provides regular expression matching operations similar to those found in Perl. Both patterns and strings to be searched can be Unicode strings as well as 8-bit strings.

c. sys: The sys module enables the program to accept the directory path as the command line argument.

d. operator: The operator module exports a set of efficient functions corresponding to the intrinsic operators of Python.

e. time: to capture the runtime of the program.

Problem 2: Stemming

Output for Problem 2:

1. Number of distinct stems in the Cranfield Text Collection: **6071**

2. Number of stems that occur only once in the Cranfield Text Collection: **2246**

3. 30 most frequent stems in the Cranfield Collection:

	STEM	COUNT
1.	the	19450
2.	of	12717
3.	and	6677
4.	a	6002
5.	in	4651
6.	to	4563
7.	is	4114
8.	for	3493
9.	are	2429
10.	with	2265
11.	flow	2080
12.	on	1944
13.	at	1834
14.	by	1756
15.	that	1570
16.	an	1388
17.	pressur	1382
18.	be	1369
19.	number	1347
20.	boundari	1185
21.	layer	1134
22.	from	1116
23.	as	1114
24.	result	1087
25.	thi	1081
26.	it	1044
27.	effect	996
28.	which	975
29.	method	887
30.	theori	881

4. Average Number of stems per document: **164.9 ~ 165 stems/document**

Program Description for Problem 2:

Algorithms and Data Structures used:

- Porter Stemmer Algorithm from the nltk library is used to stem the words to its root.
- The program refers the hash map of tokens built by the tokenizer.
- Each token is retrieved and its stem with its frequency of occurrence in the Cranfield Collection is kept in another hash map called stems as key and its value.
- If the hash map already contains a particular stem, then its frequency is incremented otherwise a new key value is created for that stem in the hash map and the frequency is inputted as 1.
- All unique stems occurring in the Cranfield Collection are counted by number of entries (length) in the hash map for stems.
- Stems appearing only once are calculated by using a counter to compute stems having frequency is 1.
- Sorted hash map is used to find the 30 most frequent stems in the Cranfield collection.
- The average number of stems per document is computed by dividing the total number of stems in the Cranfield Collection by the number of files in the Cranfield collection.