NIKITA VISPUTE
Net ID: NXV170005
CS 6322.001

Homework 2

Program Description:

The Cranfield corpus is first preprocessed to:

1. Remove the SGML tags
2. Replace special characters and digits space
3. Remove punctuation marks
4. Remove word possessives and contractions
5. Remove apostrophes in words ending with apostrophes
6. Remove extra spaces
7. Convert all text to lower case

Then the corpus is tokenized into a hashmap of tokens.

Dictionary data structure is used to create the index from the Cranfield corpus after removing the stop words. The stop word list is used from the nltk library.

For every entry in the dictionary in both versions of the indexes, there is also:

- Document frequency (df): The number of documents that the term/stem occurs in.
- The list of documents containing the term/stem.

For each document in the posting lists, there is also:

- the document ID.
- the Term/stem frequency (tf): The number of times that the term/stem occurs in that document.
- he frequency of the most frequent term or stem in that document (max_tf).
- the total number of word occurrences in the document (doclen).

Both indexes are built using the Single-pass in-memory indexing (SPIMI) algorithm.

Index_Version1.uncompress

- Dictionary of lemmas.
- Lemmatization using Word Net Lemmatizer from nltk library.

Index_Version2.uncompress

- Dictionary of stems.
- Stemming done using Porter Stemmer algorithm from nltk library.

Index_Version1.compressed

- Blocked compression with k=4 for dictionary of lemmas/terms.
- Gamma encoding for the gaps between document-ids for the compression of postings files.

Index_Version2.compressed

- Front-coding with the block size k=8 for compression of dictionary of stems.
- Delta encoding for the gaps between document-ids for the compression of postings files.

Resources Used:

- Wordnet Lemmatizer, Porter Stemmer, Stopwords from nltk library
- Pickle package for serializing and de-serializing a Python object structure so that it can be saved on disk. Pickle serializes the object (list, dict, etc.) before writing it to file.

Required Files: HW2.py and Cranfield directory

Python version used: 2.7.5 (UTD csgrads1 Server), 3.7.1 (Windows 10 personal machine)

The zip folder Homework2 contains:

i. HW2.py
ii. HW2.txt
iii. Index_Version1.compressed
iv. Index_Version1.uncompress
v. Index_Version2.compressed
vi. Index_Version2.uncompress
vii. Report
viii. README
ix. Term_smallest_df_Index1.txt
x. Stem_smallest_df_Index2.txt

The entire output for the program will get printed into the HW2.txt file and the other 4 files will be generated.

For convenience,

1. Output for “Term with the smallest df from index 1” is stored in the text file “Term_smallest_df_Index1.txt”
2. Output for “Stem with the smallest df from index 2 ” is stored in the text file “Stem_smallest_df_Index2.txt”

HW2.txt contains answers to all the 12 additional extra-credit questions (100 points) along with the dictionary of lemmas (Index_Version1) as well as the dictionary of stems (Index_Version2) with maximum 3 postings files printed out for the sake of convenience.

Output Explanation:

The results are according to the HW2.txt output file generated after executing the program on the csgrads1 server. This file is in the zipped folder.

1. The elapsed time ("wall-clock time") required to build any version of your index

Answer:

Time taken to create **uncompressed Index Version 1 using lemmas**: **6.12175512314 secs.**
Time taken to create **uncompressed Index Version 2 using stems**: **9.38028001785 secs.**
Time taken to create **compressed Index Version 1 using lemmas**: **0.4323890209197998 secs.**
Time taken to create **compressed Index Version 2 using stems**: **0.0404660701751709 secs.**

2. The size of the index Version 1 uncompressed (in bytes)

Answer: Size of the **Index version 1 uncompressed** (in bytes): **3776456**

3. The size of the index Version 2 uncompressed (in bytes)

Answer: Size of the **Index version 2 uncompressed** (in bytes): **3619165**

4. The size of the index Version 1 compressed (in bytes)

Answer: Size of the **Index version 1 compressed** (in bytes): **1460247**

5. The size of the index Version 2 compressed (in bytes)

Answer: Size of the **Index version 2 compressed** (in bytes): **74335**

6. The number of postings in each version of the index

Answer:

Number of inverted lists in Index version 1: 7855
Number of inverted lists in Index version 2: 5965

7. The df, tf, and inverted list length (in bytes) for the terms: "Reynolds", "NASA", "Prandtl", "flow", "pressure", "boundary", "shock" (or stems that correspond to them)

Answer:

Term	Document Frequency	Total Term Frequency	Inverted List Length
Reynolds	200	384	1680
NASA	145	148	1256
Prandtl	63	80	648
flow	730	2080	6240
pressure	551	1382	4856
boundary	467	1185	4280
shock	239	737	2224

8. The df, for “NASA” as well as the tf, the doclen and the max_tf, for the first 3 entries in its posting list.

Answer:

Term	Document Frequency
NASA	145

Postings files for NASA:

Doc-ID	Term Frequency	Max term frequency	Doc length
13	1	15	207
20	1	6	137
31	1	8	207

9. The dictionary term from index 1 with the largest df and the dictionary term with the lowest df.

Answer:

Term with the largest df from index 1
Largest DF terms from index 1
Freq : 728
Term : ['flow']

Term with the smallest df from index 1
Smallest DF terms from index 1
Freq : 1
List of terms given in the text file Term_smallest_df_Index1

10. The stem from index 2 with the largest df and the dictionary term with the lowest df.

Stem with the largest df from index 2
Largest DF terms from index 2
Freq : 730
Term : ['flow']

Stem with the smallest df from index 2
Smallest DF terms from index 2
Freq : 1
List of stems given in the text file Stem_smallest_df_Index2

11. The document with the largest max_tf in collection.

Answer:

```
Largest Max Frequency Document is:
            DocID     Max_tf
Cranfield0  988    -    27
```

12. The document with the largest doclen in the collection.

Answer:

```
Document with largest doc-len in collection:
            DocID     Doc_len
Cranfield0  1061   -   665
```