Programming Assignment #1 (due in 1 week)

Corpus Statistics

Zipf's law (cf. Section 5.1 in the textbook) predicts that the number of times the *ith* most frequent word will be seen is about k / i times the frequency of the most common word, for *some k*. You can investigate whether this is so by examining a collection of text and counting the number of occurrences for each word. For this assignment, links to two electronic texts have been placed on the course web page: (a) 1862 passages of text from the Jane Austen novel, *Sense and Sensibility*, and (b) 100k snippets of news from *Radio Free Asia*. Download these files and write a program that you will run separately on each text. Your program should:

- Perform some normalization of the text. For example, split on spaces, remove punctuation, and lower-case words. Give some thought to it. You may, but you are not expected or required to use a stemming algorithm. Do not perform stopword removal.
- Report the number of 'paragraphs' processed; we'll consider each paragraph to be a 'document', even though all paragraphs are contained in a single file.
- Report the number of unique words observed (*vocabulary size*), and the total number of words encountered (*collection size*, in words).
- Calculate both the total number of times each word is seen (*collection frequency* of the word) and the number of documents which the word occurs in (*document frequency* of the word).
- Sort the words by collection frequency, and print out the rank, the word itself, its collection frequency, and its document frequency (one word per line) for the 100 most frequent words by collection frequency.
- Print the same information for the 500th, 1,000th, and 5,000th ranked words. (But do **not** just printout the top 5,000 lines.)
- Calculate and print the *number* of words that occur in only one document. I do not want to see the words themselves. For *Sense,* I believe *crossness*, *pecuniary*, and *puppism* are examples of words that only appear in one of the documents in the collection. Finally, calculate the percentage of the dictionary terms that have this property of only occurring in just one document. Such words are known as *hapex legomena*.

Run your program on both datasets, and in addition to the output described above, provide a brief writeup that: (a) describes how you normalized the text and determined what a "word" is; and, (b) summarizes any similarities and differences in the top-100 terms from *Sense* and *RFA*.

You can (and should) create the required lexicon[1] in one pass over the input text. After sorting terms by frequency it should be easy to extract the pieces of information that I am asking you to report. In the input files paragraphs are indicated with <P ID=XXXX> tags indicating the start of each new paragraph. Some 'paragraphs' are short, some are longer; it may be that none are empty, but I have not verified this.

Lexicons are a key data structure for IR systems. In future assignments you will need a lexicon, so you might find it worthwhile to make your dictionary modular and reusable. In particular you will want it to fit in memory, to be storable on disk for subsequent reloading, and to support efficient term lookups. Some representations you might consider are in-memory hashtables or binary trees (*e.g.,* in Java HashMaps or TreeMaps; in Python, dicts) Provide your source code, program output, and brief writeup (as described above) all in a single PDF file.

I have coded a solution to the exercise above in ~240 lines of (verbose, commented) Java code and in less than 100 lines of Python. Many students in the past have been able to submit good, readable solutions in about two pages of code. Of the five programming assignments, this is by far the simplest.

I obtained the *Sense* text from Project Gutenberg (https://www.gutenberg.org/). The *RFA* data was obtained by downloading articles from the Radio Free Asia website.[2] The files are about 38 MB (*RFA*) and < 1 MB (*Sense*), and should be encoded in UTF-8.

---

[1] You are building a dictionary. Here the words dictionary and lexicon are interchangeable.

[2]

If you have questions about the assignment you can post them to the discussion forum, or contact me by email.

Below are some short samples from the texts:

From *Sense:*

<P ID=127>
"Infirmity!" said Elinor, "do you call Colonel Brandon
infirm? I can easily suppose that his age may appear much
greater to you than to my mother; but you can hardly
deceive yourself as to his having the use of his limbs!"
</P>

<P ID=128>
"Did not you hear him complain of the rheumatism?
and is not that the commonest infirmity of declining life?"
</P>

<P ID=129>
"My dearest child," said her mother, laughing,
"at this rate you must be in continual terror of MY decay;
and it must seem to you a miracle that my life has been
extended to the advanced age of forty."
</P>

From *RFA*:

<P ID=20706>
But it was also preceded by an unusual wave of criticism from Russian media, television broadcasts and Web sites, experts said.
</P>

<P ID=20707>
"Clearly, Russia was interfering," said S. Frederick Starr, chairman of the Central Asia-Caucasus Institute at Washington's Johns Hopkins University.
</P>

<P ID=20708>
Starr cited a "publicity campaign against Bakiyev" and "a large FSB [Russian security agency] presence in Bishkek, which is still there."
</P>

<P ID=20709>
Moscow's role has stirred fears of power plays and unrest among bordering countries including China, Kazakhstan, Uzbekistan, and Tajikistan, Starr said.
</P>

<P ID=20710>
"Russia's hyperactivity has managed not only to very much raise concerns among Kyrgyzstan's real neighbors....but I'm sure it is also raising a serious concern in Beijing," he said.
</P>

Note the “P” tags are on separate lines from the text. Other datasets used in later assignments will be formatted similarly. For this assignment it is not important to parse the docids, but you will need to do this on future programs.