Homework #1 (due in 1 week)

Corpus Statistics (70 points)

Zipf's law (cf. Section 5.1 in the text) 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: the *Bible* (American Standard Version) and Victor Hugo's *Les Misérables*. Download these files, and write a program (or programs) that you will run separately on each text. Your program should do the following:

- Perform some normalization of the text. For example, split on spaces, remove punctuation, and lower-case words. Give some thought to it, and be sure to describe how you determine what constitutes a word.
- 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[1].
- Report the number of unique words observed (*vocabulary size*), and the total number of words encountered (*collection size*, in words).
- The program should 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).
- Identify the 30 most frequent words (by total count, also known as collection frequency) and report both the collection frequency and the document frequency for each. Order them from most frequent to least.
- Also print the 100th, 500th, and 1000th most-frequent words and their frequencies of occurrence. (But please do not just printout the top 1000.)
- Calculate and print the *number* of words that occur in exactly one document. (For *Les Mis*, I believe *reptile*, *silkworm*, and *signatures* are such words.) What percentage of the dictionary terms occur in just one document?

You can (and should) create the required lexicon[2] in one pass over the input text. After sorting terms by frequency it should be fairly 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 enable efficient lookups. Some representations you might consider are in-memory hashtables, binary trees, or tries. If you use Java, using the built-in HashMap or a TreeMap classes is a very reasonable idea. Provide your source code and the requested output described above.

I have coded a solution to the exercise above in ~ 240 lines of (verbose, commented) Java code and in previous classes I have had former students submit good, readable solutions in about 2 pages of quality Perl or Python. Of the programming assignments, this is by far the most simple.

The Bible text was obtained from http://unbound.biola.edu/; versions in numerous languages are available. Hugo's novel was obtained from Project Gutenberg: http://www.gutenberg.org/. Both texts are less than 5MB in size.

---

[1] This is a lot easier than for me to give you 20,000 separate files to work with.

[2] You are building a dictionary. The words dictionary and lexicon are interchangeable here. (This is an example of synonymy.)

Below are some short samples from the texts:

From *Les Misérables*.

<P ID=205>
He was indulgent towards women and poor people, on whom the burden
of human society rest. He said, "The faults of women, of children,
of the feeble, the indigent, and the ignorant, are the fault
of the husbands, the fathers, the masters, the strong, the rich,
and the wise."
</P>
<P ID=206>
He said, moreover, "Teach those who are ignorant as many things
as possible; society is culpable, in that it does not afford
instruction gratis; it is responsible for the night which it produces.
This soul is full of shadow; sin is therein committed. The guilty
one is not the person who has committed the sin, but the person
who has created the shadow."
</P>

From the Bible (the first three verses in Genesis):

<P ID=1>
In the beginning God created the heavens and the earth.
</P>

<P ID=2>
And the earth was waste and void; and darkness was upon the face of the deep: and the Spirit of
God moved upon the face of the waters
</P>

<P ID=3>
And God said, Let there be light: and there was light.
</P>

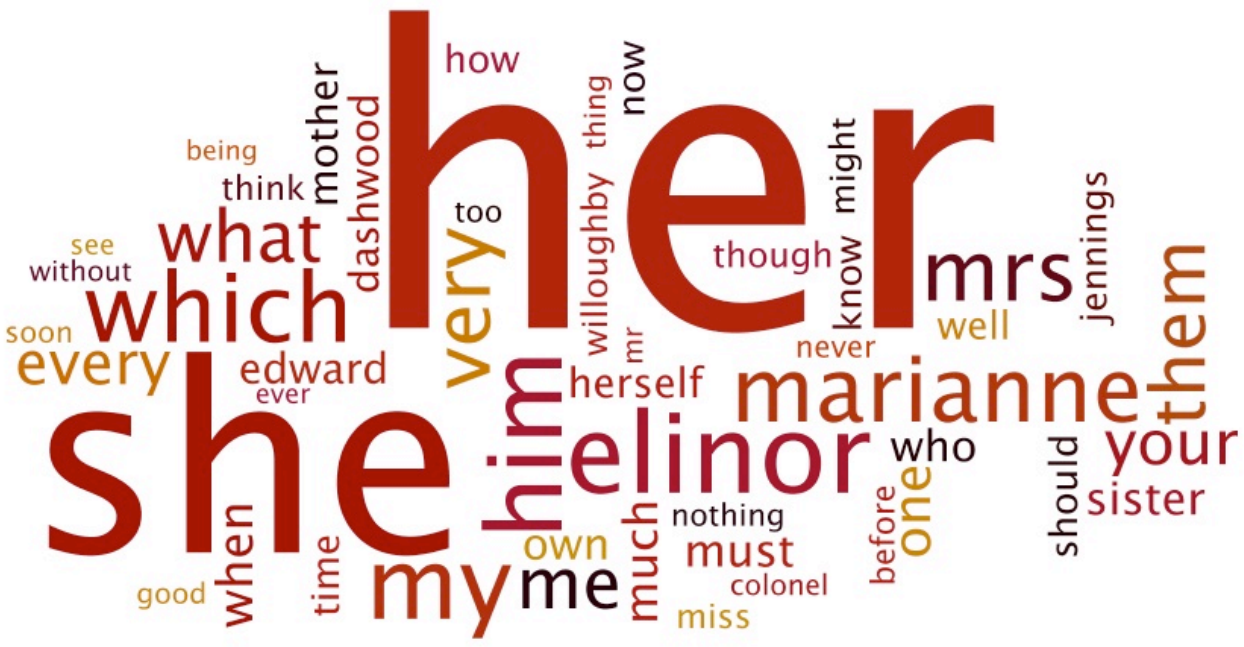
Note the “P” tags are on separate lines from the text. Other datasets used in later assignments will be formatted similarly.

For More Fun (extra credit, 3 scant points)

Tag clouds. Use the results from your program to create a pretty or interesting tag cloud. For example, the cloud below was created using Wordle (TM) by supplying weights from the text of Jane Austen's *Sense and Sensibility* novel. I removed 30 or so 'stopwords' (clearly not enough for the best tag-cloud generation) and used total count as 'weights'. I used the tool at: http://www.wordle.net/advanced (you may use this tool, or another). I pasted the top 50 remaining words from my program which looked like this (to accomodate Wordle's weight format):

her:2551
she:1613
elinor:685
mrs:530
which:593

Your results will vary, depending on if you remove stopwords, how you define what is a word, how many words you use, which text corpus you use, any options used by the tool you use, etc...



Short Problems (10 points each)

1. Problem 1.7 (IIR, pg. 13)

2. Problem 2.5 (IIR, pg. 35)

3. Problem 2.8 (IIR, pg. 41)