CS380 Information Retrieval and Web Search

Homework 1

Simple tokenization and analysis of your IR dataset

Due: Sunday, Feb 2 before midnight.

What to hand in (and how)**:**

Email me a copy of the reports described for Part 1 and Part 2. Also, send me a copy of your final plan from Lab 1. The reports may be in a single document (preferentially). The document(s) must be easily readable by me. Generally this means PDF. Pretty much the only alternative I would label "easy for me to read" is a web page of your own construction (in which case just send me a link). If you have another idea, give it a shot but keep in mind that I am the arbiter of "easy for me".

Overview:

In this assignment you will take the IR dataset you constructed in the first lab and do some processing to make it easily usable. Then you will analyze your dataset.

Part 0: Finish collecting your dataset

If you did not finish collecting the dataset in lab, do so.

Part 1: Create a clean text dataset.

Most text you get from the web has stuff which is uninteresting from an IR viewpoint. For instance, if you have web pages then you will need to remove all html tags. If you have pdf, get the text out of the PDF document. If your dataset has novel-length tems, break them up into chapters or in some other reasonable way.

Parts of this task may be most easily done by hand. For instance, if you downloaded documents from Gutenberg, they will have Gutenberg headers and trailers. You could write some code to eliminate these, but hand editing is OK too.

There are lots of public domain tools out that you can use for this task; feel free. Similarly, talk amongst yourselves. Find a friend and share tasks. One person comes up with a way to do task X, another person task Y.

Once you complete this step, write a report of more than 1 paragraph describing all of the things you needed to to do clean your dataset. You should cover at least the following points:

1. What you did
2. Why what you did was necessary and a good idea.
3. How you accomplished what you did.
    1. If you used public domain tools, document those tools (where you got them, etc).
    2. If you used something written by a classmate, who that was.
    3. If you did things by hand, why did that seem like a reasonable choice.
4. How long did all of this take. (Approximately). (I will not give any points, plus or minus for speed, or lack thereof. I simply want to know.)
5. You may have written a lot of code for this purpose. I do not need to see it. So things like comments and adhering to stylistic conventions are completely up to you. On the other hand if you get stuck and need help, making sure that your code is understandable is the first step to getting help.

Part 2: Analysis

Unlike Part 1, the work here should be completely your own. (That is, do not code swap with classmates, do not copy from the web, etc.)

As with Part 1 the product I will grade is a written report. The report for this section will almost certainly be longer than the report for Part 1.

Answer at least 3 of the following topics:

1. Reproduce a table like the one from class on Monday showing the number of tokens in you text under a number of processing scenarios (just split by whitespace, removing all punctuation, downcasing, etc). Also, come up with at least one column that I did not use. Discuss this table. What does the data in your table imply about searchability for your data? For instance, lots of unique words allow for very specific results, but tend to make documents hard to find.
2. Do the tokens in you set follow Zifp's law? Why? Why not? Graphs will certainly help. Additional stats will also help your answer. (For instance, if you do a simple linear fit to Zipf, what is the residual error.) A simple yes/no is **not** sufficient. For

any statistics you use, you must demonstrate some understanding of that statistic. For instance, do not just say “the correlation is 0.72”; also say the what that correlation level means for the relatedness of the series being compared.

3. How unique is your token distribution? Answer this question by comparing your token distribution to that of the Google N-gram dataset. For this task, just use the 1-gram set is available at
   `/home/dkumar/Trigrams/Data/1gms/vocab`
   Each line in this file has the form:
   token count
   where token is some string and count is the number of times it appears. This file has 13 million lines and is 185M so please do not make a local copy in your directory. (If you really want a local copy, use `ln -s` instead.) At the very least consider some of the following issues

   1. What is the rank correlation between your set and the google set (for the words in your set)? To compute this correlation, determine for each token in your dataset the rank order of that token in the google dataset where the ranks are only for those terms in you dataset. Then compute the correlation over the two rankings.

   2. What are the of the words in your set that are conspicuously higher ranked for you than for google? Why?

   3. What are words in the google set that are much more highly ranked than for you? (The rationale of your answer to Q3 may be very similar to you answer to Q2).

   4. A question of your own choosing about the tokens in your dataset.

4. A topic of your own choosing. This is especially important for those of you with a foreign language dataset as topic 3 will be a challenge. (Datasets similar to the one above may be available for other languages; you would have to find them.)