CSE 6331   Cloud Computing
Summer 2023, © DL, UTA, 2023

Programming Assignment 5

Cloud Assignment 5 – dealing with "large" textual information

Description:

Much of the available, digitized information is in the form of text.
(We will consider web, and similar, as a variation of text.)
We will simplify handling to only text, but the concepts
could be extended.

More:

Given several (say a few hundred or thousand) text files, which we will generically call "documents", we want to find those documents relevant to a user's needs (requests.)
(This is, in general, what "search engines" do.)
For this assignment we will simplify many of the interesting details, but try to emphasize many of the issues and interesting approaches to searching text.
Almost all original, source documents need to be "cleaned" (may include removing pictures, font details, pagination, and similar.)
Here ALL non ASCII information will be removed.
(Note, we will assume that docs are in English, but Spanish, French and similar languages are not difficult to extend processing, Chinese is more difficult.)

Doing a simple word scan is simple, very time consuming, and usually gives poor results. Preprocessing the original docs is important.
The following are simple original doc processing:
Dealing with upper or lower case letters, usually changing to lower case.
Remove punctuation (or most)
Remove very common words ("stop words" such as the, or, and.)
Additionally (optionally):
Word stemming (cats -> cat)
(Many others)

Then the words in the documents are extracted and indexed, with "pointers" back to the individual relevant documents, and where found.

Then, through some sort of interface, a "search" of relevant documents can be done.
The simplest, and usually poorest result is a single word, such as "cloud".
Combinations of words, in close proximity, usually do better "cloud computing".
(These are sometimes referred to as bi-grams, and tri-grams, and extensions.)

One, free, source of thousands of texts is: https://www.gutenberg.org/

Notice that small optimizations will often give much better results:
For example "red hot" may be the same as "hot red" (interchange word order)
Letters in search may be transposed (or missing): "teh" instead of the,
questionble instead of questionable
There are word lists that may help: (MIT site, github, many others,
depending on what you want.)

Users of this service will interact with your service through web page interfaces, all processing and web service hosting is (of course) cloud based.

Additional Details:

A user should be able to do searches based on words or word combinations to find Relevant documents and where in that document (such as line or offset.) (Show lines or paragraphs that match.)

https://www.ranks.nl/stopwords

https://www.semrush.com/blog/seo-stop-words/

https://towardsdatascience.com/python-libraries-for-natural-language-processing-be0e5a35dd64

https://www.infoworld.com/article/3519413/8-great-python-libraries-for-natural-language-processing.html

Please, submit through Canvas.
All work must be your own, or from a group.
(Same as previous assignments)