Course Project

CS 242: Information Retrieval & Web Search

Winter 2020

Build a Search Engine

You must work in teams of five. If you cannot find a partner, email the TA (Merlin if you are ground student and Luxun if you are online) to connect you to other students who are looking for a partner. Teams must be formed by end of 2nd week of classes, and their composition emailed to the corresponding TA.

Each project report must have a section called "Collaboration Details" where you should clearly specify the contributions of each member of the team.

Part A: Collect your data and Index with Lucene

A1: You have the following options:

a. Crawl the Web to get Web pages using jsoup (http://jsoup.org/). You may also use Scrapy (https://scrapy.org/) if you prefer Python. You may restrict pages to some category, e.g., edu pages, or pages with at least five images, etc.
b. Crawl the Web to get images with their captions and names (to be used for indexing in next parts) using jsoup or Scrapy. Only use smaller imaged (<200KB) so you don't stress our Hadoop cluster later.
c. Use Twitter Streaming API (https://developer.twitter.com/en/docs/tutorials/consuming-streaming-data.html) to get Tweets. You can also use Tweepy (tweepy.org) if you prefer Python. (hint: Filter to only collect geotagged tweets, so you can then display them on a map in Part B.)
d. Your own ideas for a dataset are also acceptable, pending instructor approval.

Collect at least 1 GB of data, but no more than 5 GB.

We recommend using Java, but not required.

A2: Index your data using Lucene (not Solr)

You will be graded on the correctness and efficiency of your solution (e.g., how does the crawler handle duplicate pages? Is the crawler multi-threaded? How do you store the incoming tweets to maximize throughput?), and the design choices made when using Lucene (e.g., did you remove stop words, and why? Or did you index hashtags separately from keywords and why?).

Deliverables:

1: Report (5-10 pages) in pdf that includes:

a. Collaboration Details: Description of contribution of each team member.
b. Overview of the crawling system, including (but not limited to).
    i. Architecture.
    ii. The Crawling Strategy.
c. Overview of the Lucene indexing strategy, including (but not limited to).

i. Fields in the Lucene index, with justification (e.g., indexing hash tags separately due to their special meaning in Twitter).

ii. Text analyzer choices, with justification (e.g., removing stop words from web documents; using separate analyzers for hashtags and keywords).

iii. Report the run time of the Lucene index creation process. E.g., a graph with run time on the y axis and number of documents on the x axis.

d. Limitations (if any) of system.

e. Obstacles and solutions

f. Instruction on how to deploy the crawler.

- Ideally, you should include a crawler.bat (Windows) or crawler.sh (Unix/Linux) executable file that takes as input all necessary parameters.
- *Example:* [user@server]./crawler.sh <seed-File:seed.txt> <num-pages: 10000> <hops-away: 6> <output-dir>

g. Instruction on how to build the Lucene index.

- Ideally you should include an executable file that takes as input all necessary parameters. *Example:* [user@serer] ./indexbuilder.sh <input_dir> <Analyzer Options>