NYU | TANDON SCHOOL OF ENGINEERING

CS6913: Web Search Engines

Assignment #1

Torsten Suel

Computer Science and Engineering
NYU Tandon School of Engineering, Brooklyn

Goal of Assignment #1:

- Build a multi-threaded web crawler that downloads pages with priorities based on some combination of novelty and importance.
- Learn about crawling and web protocols hands-on
- Learn about ranking functions
- Learn about Python
- See resources on course page
- Start now!

Basic Concepts:

- Given a URL, a crawler:
  - Checks the URL to decide if it should be crawled
  - Does DNS lookup for name resolution
  - Fetches robots.txt from site unless robots file cached
  - Fetches the page from the server
  - Parses page to find new hyperlinks
  - Updates novelty and importance scores of other pages based on the newly crawled page, as needed
  - Inserts newly found links into crawl priority queue if warranted
  - Then removes the next URL from priority queue …

Using Search Results as Seed Pages

- Your crawler should take a search query as input
- Your crawler should then fetch the top-10 results from a search engine (e.g., google, bing)
- Uses some appropriate library to access engine
- Then your crawler should put these 10 results into the queue as seed pages

Using Novelty and Importance to Guide Crawl

- You may define novelty of a page based on how many pages from the same domain have already been crawled.
- For example, 1 if no page crawled, 0 otherwise
- Or: 1 if 0 pages crawled, and 1/(k+1) if k pages crawled
- Or some other measure that takes number of URLs from that site that are currently in the queue into account
- Importance could be number of other already crawled pages that have a hyperlink to this page
- Or something more complicated like running Pagerank on the already crawled subgraph? (This gets tricky)

Defining Page Priorities

- Next, you need to combine novelty and importance to get a single priority score. (Higher score meaning better.)
- Maybe a weighted linear combination of novelty and importance, with suitable weights?
- You can make a good choice on your own.
- Maybe use priority queue for URLs that have yet to be crawled, organized by priority score.
- So in each crawl step, extract the one with highest priority

Updating Priorities in the Queue

- Note: when we crawl a new page, this can influence the priorities of many other pages currently in the queue.
- All pages on the same site will have their priority lowered as their novelty scores decrease.
- All pages pointed to by this page will have their importance increased, so priority will increase.
- How to efficiently update all the pages that are impacted?
- Hint: Organize priority queue based on importance and a potentially outdated estimate of the novelty.
- When dequeuing a page for download, update novelty, check if after update still highest priority. If not, push back into PQ.

Python:

- An easy to learn but powerful programming language
- Scripting language (compare to Perl, tcl, PHP, etc.)
- Interpreted and slower than C/Java for many tasks
- But easy to pick up and use
- Very relaxed about types (can assign anything to anything)
- Nice data structures, string/parsing utilities, web programming → many, many libraries available