CS 410 – Text Information Systems

Expert Search Automated Crawler – Self Evaluation

By Cheerla3 (Mohana Venkata Kalyan Cheerla)

Self-evaluation:

- Have you completed what you have planned?

  Yes. I have accomplished everything that I planned to do for "Expert Search - Automated Crawler". Mentioning the successful accomplishments below -

  1) Automated the Crawler using powerful and efficient Scrapy Spider library in Python.
  2) Made the Crawler dynamic in nature so that, it is be driven by the University home/domain URL, starts the crawling activity from home page URL, and crawls all the web pages in University websites.
  3) Integrated with "Bio Page Classifier" model to check whether the crawled/scraped web page is a Faculty Bio-page before saving it into the Bio-pages folder.
  4) Implemented the Hashing technique in saving the Bio-page files to avoid the duplication of Faculty Bio pages inside the Bio-page folder. This concept helped to override the previously crawled bio-page file of a faculty inside the Bio-page folder even if his/her Bio-text get updated between the previous iteration of crawling and the current iteration of crawling.
  5) Optimization: To reduce the burden on the Crawler and Bio-page Classifier, I did use a filtration concept to avoid some web pages (False Positives in terms of Bio-pages) based on their URL strings that are making sure that a web page is a Bio-page (example: publications, news, sports). This optimization technique became very successful and it led to a very good performance.
  6) Implemented a URL Tree technique to trace its execution.
  7) Made the crawler accessible to a Batch/Cron job to allow its scheduling in future.
  8) Documented everything at Github Readme page.
  9) Also, uploaded the Project report to make it easy for its user understand its usage.

- Have you got the expected outcome? If not, discuss why.

  Yes. I got the expected outcome. It is crawling/scraping the university websites very much faster than expected. It is working very much efficient. However, it is taking longer time when we do not put any filter on the count of Universities and the depth of the Webpages in each University, which is expected due to the presence of so many Universities in the world and their giant websites hosting hundreds of thousands (or) millions of web pages.

  In that scenario, it is draining the system resources when I run it in a developer's machine. However, it can handle that much big web page population as well if we deploy it in a server using a "Parallelization over multiple machines" concept. Hence, I am completely satisfied with what has been achieved in a short span of time.

Thank you so much for helping us with the list of great projects you planned for us in this course. Grateful to you!!