**Team #32**

Sanjay Jagaeesh

Manohar Thagadur Natraju

Soumya Gowda

Pramod P. Setlur

CSCI–572 Assignment 1

*Crawling using Apache Nutch*

Dark and deep web are two potential places to sell guns illegally. We used Apache Nutch to crawl for images of guns, Apache Tika for content and metadata extraction, and Selenium to automate Ajax interaction and more.

CSCI–572 Assignment 1

## Question 1: Download and configure Nutch to crawl Weapons images as identified in the seed list.

1. The list of files we modified to deal with politeness are as follows:

Nutch-Site.xml:

* + 1. We set http.agent.rotate to ‘true’ and added 7 agent names in agents.txt.
    2. We increased the http.timeout from 10,000 to 80,000.
    3. We changed the http.content.limit to -1 to remove the limitation on the size of the crawling file.
    4. We increased the fetcher.threads.fetch from 10 to 20.

Our Nutch-Site.xml contains the other changes.

* 1. Regex-urlfilter.txt and Automaton-urlfilter.txt
     1. We skipped domains like facebook.com, tumblr.com, twitter.com, etc. to filter majority of images unwanted to us.
     2. We removed the image extensions to avoid skipping images.
  2. Mimetype-filter.txt:
     1. We added image/\* mimetype to allow crawling of documents with images/\* and text/html mimetype.
  3. Suffix-urlfilter.txt:
     1. We commented all the image suffixes to prevent them from exclusion.

1. We enabled rotating bots in agents.txt.
   1. Spider#32Crawler, MasosapaCrawler, etc

## Question 2: Perform crawls of Weapon’s images sites

1. We use the NutchPy library
2. We wrote a python script to generate all the image mimetypes that were found in our crawled data.
3. We found the following mime types:
   * 1. image/png
     2. image/jpg
     3. image/jpeg
     4. image/gif
     5. image/bmp
     6. image/x-icon
     7. image/tiff
     8. image/x-xbitmap
     9. image/pjpeg

Many of the URLs that weren’t fetched were mainly because of java.net.SocketTimeoutException, java.lag.IllegalArguementExeption, HTTP 403.

1. The list of 100 Urls that we found difficulty in fetching has been added in “*URLs not fetched.xlsx”*
2. We crawled all the 85 urls with 30 rounds

The crawl stats at this stage is tabulated as follows:-

$ bin/crawl urls/ ~/CSCI-572/guns\_crawl\_Step2 30

|  |  |
| --- | --- |
| Total URLS | 4710184 |
| retry 0: | 4625184 |
| retry 1: | 64302 |
| retry 2: | 20698 |
| Min Score | 0.0 |
| Avg Score | 2.9849576E-5 |
| Max Score | 1.173 |
| Status 1 (db\_unfetched) | 4348772 |
| Status 2 (db\_fetched) | 314453 |
| Status 3 (db\_gone) | 28243 |
| Status 4 (db\_redir\_temp) | 5114 |
| Status 5 (db\_redir\_perm) | 6146 |
| Status 6 (db\_notmodified) | 7 |
| Status 7 (db\_duplicate) | 7449 |

## Question 3: Use the information you learn in question 2 to extend the Nutch Selenium Protocol Plugin

1. We used the protocol-interactive-plugin for Nutch and enabled it go past the pages that require Ajax interaction.
2. We wrote *Custom Handlers* trying to generalize a pattern for the websites that required a login or those which were paginated.
3. Upon building and testing Nutch with the plugin enabled we could see Firefox pop open and see Nutch attempting to follow the instructions we had written in our *Custom Handlers.*