Curated scrapes: Designing a web crawler for ethnographic research data

Kim Pham (Librarian), Kirsta Stapelfeldt (Librarian), Alejandro Paz (Assistant Professor, Anthropology) - University of Toronto Scarborough

Background

This pilot project brings together anthropologist Alejandro Paz, faculty and students from the Department of Computer & Mathematical Science, and the Library's Digital Scholarship Unit to investigate the increasingly central role that Israeli online journalism in English plays in the global dissemination of news stories about the Middle East. The stories in English from online newspapers circulate widely, and inform the vigorous public debate about Israel and Palestine. Online digital media are now recognized as a central, if poorly understood, aspect of globalization. In order to understand how this digital news is being produced, disseminated, and used across a global digital newsscape, we formed a team to build MediaCAT (Media Crawler & Archive Technology), a web crawler and archive application suite. As product manager, the Digital Scholarship Unit plays crucial role in the development process to ensure the application is designed in a responsible, sustainable manner to support future web archiving services provided through the library.

Technology

MediaCAT is an open-source Python based application that uses the Django framework. Django was originally developed for a news environment and is designed for database-driven sites that require search and indexing of large volumes of content. Django provides a customizable front-end and authentication system that allows the creation of multiple users to access the site. The web crawler is integrated in the framework and uses the Newspaper API to perform a scrape of a target list of sites if it finds a positive match for a given set of sites and/or keywords.

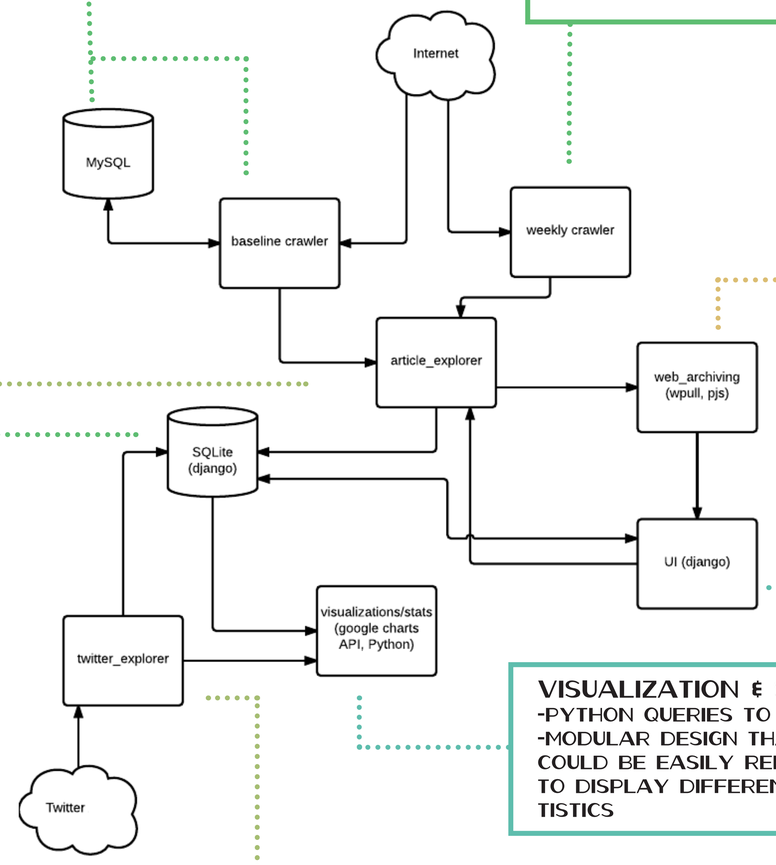
By utilizing the Newspaper API, the application extracts content, unlike most other open-source web crawlers. The Newspaper API uses natural language processing (via NLTK) to identify and parse content that is written into the database which is then made searchable. Crawling is also more efficient by only capturing sites that match given queries. These URLs are then captured using wpull and PhantomJS to store WARCs for in-depth technical analysis. The continuously updating data will be used to investigate the process of producing news for a global public sphere. A second function is to monitor a set of Twitter handles for the same set of sources and keywords. The result is a list of individual URLs or tweets with references, either mentions or hyperlinks, to one of the sources.

ARTICLE_EXPLORER
-GETS ARTICLE LINKS FROM CRAWLER
-DETERMINES IF THEY MEET CRITERIA
-ADDS AND UPDATES TO SQLITE DB
-CAN INPUT AND RUN PRIORITY QUEUE FROM FILE TO CRAWL
-EXTRACTS AND PARSES ARTICLE TEXT
-MATCHES KEYWORDS & DOMAINS
-USES NEWSPAPER API FOR METADATA MATCHING BASED ON HEURISTICS AND CSS SELECTOR & REGEX

BASELINE CRAWLER & MYSQL DATABASE
-STORES STATE ON MYSQL DB (VISITED OR TO VISIT)
-DATA IS NOT PERSISTENT; LASTS FOR DURATION OF CRAWL
-RECURSIVELY VISITS PAGES ON SITE

WEEKLY CRAWLER
-FINDS RSS FEEDS AND EXTRACTS ARTICLE LINKS USING NEWSPAPER API
-ARTICLE LINKS SENT TO ARTICLE_EXPLORER TO DETERMINE IF THEY SHOULD BE STORED

SQLITE DATABASE
-STORES MOST OF THE DATA
-SOURCE\REFERRING SITE INFORMATION
-MATCHING ARTICLES & METADATA
-USER ACCOUNT INFORMATION



WEB ARCHIVING
-GENERATES WARC FILES USING WPULL (DESIGNED FOR LARGER CRAWLS THAN WGET WITH WITHOUT ADDITIONAL REQUIRING DEMAND ON RESOURCES)
-SCREEN CAPTURE OF HTML, CSS AND SVG ELEMENTS IN PNG AND PDF FORMAT IS GENERATED USING PHANTOMJS
-ARCHIVING INITIATED AT THE SAME TIME THAT ARTICLE_EXPLORER FINDS AN ARTICLE.

USER INTERFACE
-DJANGO PROVIDES ADMINISTRATIVE INTERFACE TO ADD USERS, GROUPS, AND MANAGE DATA
-CRAWLERS CAN BE STOPPED, PAUSED AND STARTED

VISUALIZATION & STATS
-PYTHON QUERIES TO DB.
-MODULAR DESIGN THAT COULD BE EASILY REPLACED TO DISPLAY DIFFERENT STATISTICS

TWITTER_EXPLORER
-GETS TWEETS
-PARSES RESULTS FROM TWITTER API
-MATCHES TWITTER URL AND @ MENTIONS

Considerations

Quality assurance and testing is constantly being performed to verify if the crawler is parsing and extracting appropriate information for the fields being indexed (such as Aritcle Title, Author, Body, Published Date). The crawler is quite accurate, but on sometimes encounters difficulty parsing content from certain domains. For these domains, the implementation of CSS selectors was necessary to manually input the location of article fields found in the source code of an webpage. However, there is still no guarantee however that these CSS selectors will work for every article in a domain, which may serve as potential area of research to improve the NLTK machine learning algorithm to detect these fields. A crucial task in the QA process is to manually look through the dataset and identify crawling rules or exceptions that should be implemented to improve the dataset. For instance, different news sites have different policies and conventions to cite information from competing news sources. The application ideally will indicate multiple hits in an article if it is matching on various keywords or different hyperlinks within an article (for instance finding http://haaretz.com/news-story-1 and http://haaretz.com/news-story-2 in a single article) but not if it refers to the same article (e.g. keyword 'Haaretz' and link http://haaretz.com/news-story refers to the same article). This and many other parsing and extraction issues to discern differences in content remains a significant challenge.

...and More Considerations

We are in the process of migrating the application to a stable server from Digital Ocean. In anticipation of this move, a product roadmap was created as tool to manage the strategic direction of application. Crawler performance is considered to be a major priority and constant area for improvement. Recently, in-memory paging was swapped out and tested with a default Django SQLite Django database without success. Afterwards, a manual MySQL database was implemented to manage visited site information which has been working but still requires further testing. Multithreading across many domains is now possible, but not within a single domain. Additional hardware upgrades (cores and RAM) will be necessary to scale up. We anticipate running our first baseline crawl on three to four domains for articles since 2004 by Spring 2016. We hope to incorporate external tools as part of our workflow to analyse our content and have been investigating warcbase to explore our WARC files and JavaScript libraries such as d3 and p5 (processing) to visualize networked information.

Github: github.com/utmediacat
Site (in-development): mediacat.utsc.utoronto.ca
Contact: Kim Pham (@tolloid), kim.pham@utoronto.ca

Library
UNIVERSITY OF TORONTO
SCARBOROUGH