Collection Management Webpages
Final Presentation

CS 5604 Information Storage & Retrieval
Professor Edward A. Fox

M. Eagan, X. Liang, L. Michael, S. Patil
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061

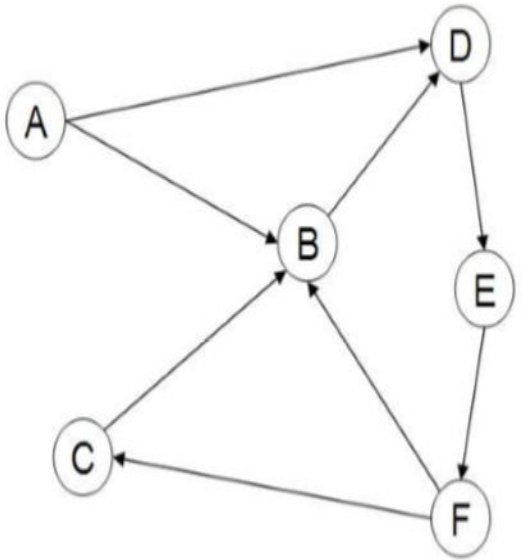
December 7, 2017

Review and Background

The Web as a Graph

Represented as a directed graph with nodes being webpages and edges being links.

1. The anchor text pointing to page B is a good description of page B.
2. The hyperlink from A to B represents an endorsement of page B by the creator of page A.



Web Crawlers

Must have Robustness (resilient to traps) and Politeness

Should Be - Distributed, Scalable, Efficient, Biased towards Quality, Continually Operate, and Extensible

Web Crawler Architecture

---

1. Start with a seed set of URLs
2. Fetch content from a URL and parse content
3. Text is fed to a text indexer
4. Extracted links added to the URL frontier
5. If continuous, URL is added back to the frontier

Event Focussed Crawler (EFC) Overview

Designed by Mohamed Farag

Event: “Something that happened at a certain place on a specific date”

Generates a list of URLs based off

of user provided seed URLs

Uses much of the standard crawling principles like anchor text for information

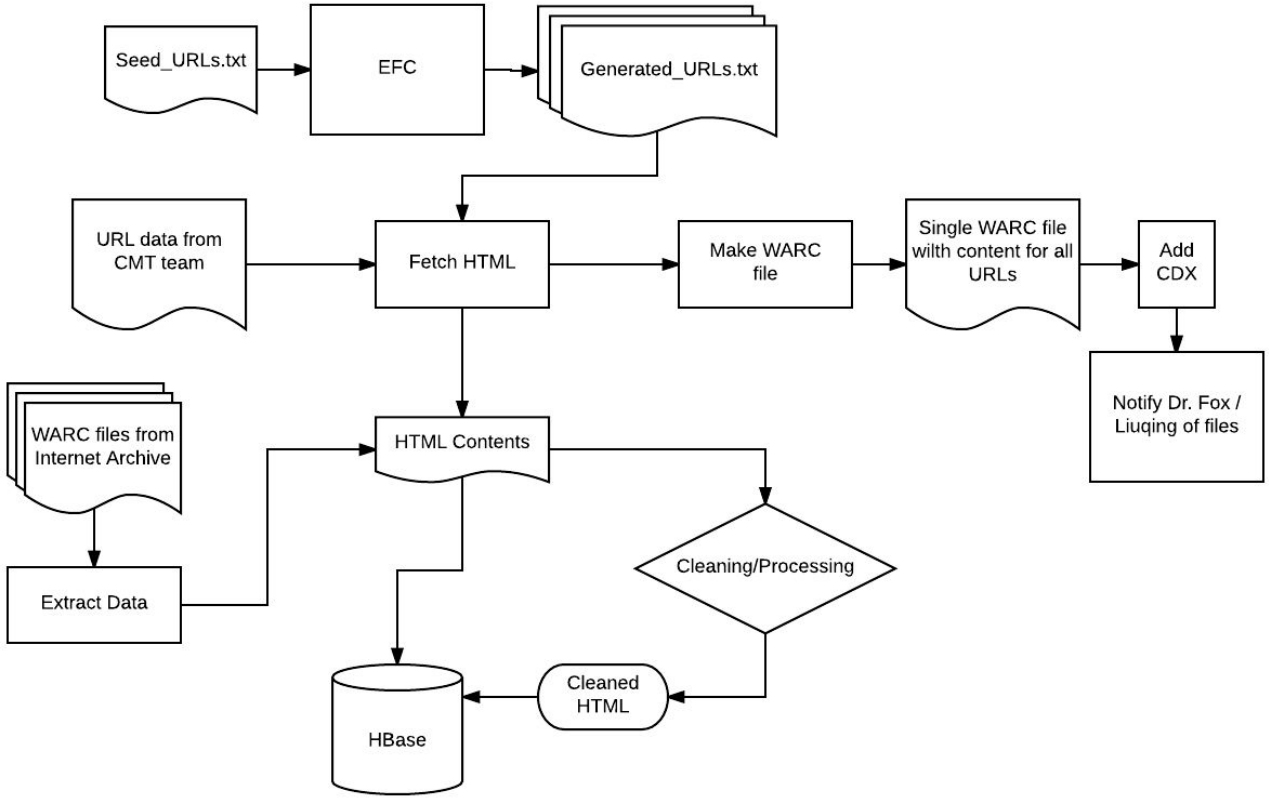
Input

Collection Management Tweets (CMT) team

URLs generated collected by hand

WARC files from organizations such as Internet Archive

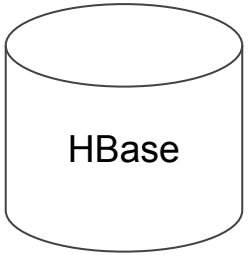
Current System Design



Output

WARC Files - to be uploaded to the Internet Archive

Raw and Cleaned HTML along with associated entities such as ‘collection_name’, ‘location’ etc. - stored in HBase



Twitter Data Flow

CMT provided CSV files for tweets stored in HBase

↓

We pulled out only tweets with URLs

↓

We check these URLs for uniqueness

↓

We fetch these pages, clean them

↓

Upload processed data to HBase

Processing Completed and now in HBase

Processing Pipeline for August21

	CSV from CMT	URL CSV	Unique URL CSV	Processed TSV
File Name	August21_final.csv	August21_finalURL.csv	August21_finalURL_condensed.csv.csv	August21_finalURL_rst_bk.tsv
Number of Rows	2,017	690	510	510

Processing Pipeline for oreclipse

	CSV from CMT	URL CSV	Unique URL CSV	Processed TSV
File Name	oreclipse_final.csv	oreclipse_finalURL.csv	oreclipse_finalURL_condensed.csv	oreclipse_finalURL_rst_bk.tsv
Number of Rows	15,243	2,411	992	992

Processing Pipeline for eclipseglasses

	CSV from CMT	URL CSV	Unique URL CSV	Processed TSV
File Name	eclipseglasses_final.csv	eclipseglasses_finalURL.csv	oclipseglasses_finalURL_condensed.csv.csv	eclipseglasses_finalURL_rst.tsv
Number of Rows	20,618	3,364	2,399	2,399

Processing Pipeline for totaleclipse

	CSV from CMT	URL CSV	Unique URL CSV	Processed TSV
File Name	totaleclipse_final.csv	totaleclipse_finalURL.csv	totaleclipse_finalURL_condensed.csv.csv	totaleclipse_finalURL_rst.tsv
Number of Rows	1,005,938	41,735	8,683	8,683

	CSV from CMT	URL csv	Unique URL CSV	Processed TSV
File Name	totalsolareclipse_final.csv	totalsolareclipse_finalURL.csv	totalsolareclipse_finalURL_condensed.csv.csv	totalsolareclipse_finalURL_rst.tsv
Number of Lines	26,788	4,703	2,941	2,941

WARC Files Processed For HBase

Process Overview

Provided with WARC files from EFC runs in CS 6604

HTML is extracted and cleaned

Resulting information is uploaded to HBase

Collection Name	oregon school shooting	Dunbar High School shooting	NIU shooting	University Alabama shooting	Virginia Tech shooting
Number of Rows	521	258	19	328	3442

These collections were based on WARC files provided to us, we speculate their sizes were a function iteration, growing bigger as the CS 6604 group worked throughout the semester

Crawling We Did

Process Overview

URL List

HTML is extracted and cleaned

Resulting information is uploaded to HBase

Collection Name	Eclipse2017	VegasShooting	Hurricane Irma
Number of Lines	1344	913	2726

These crawls were conducted throughout the semester and their size was based of hand examining urls to evaluate their relevance and setting the EFC to conduct crawls that produce quality results

Parallel Processing

- Big data!
- Better performance
- We have finally exploited the advantage of cluster

PySpark

What is **Spark**:

Apache Spark is an open-source cluster-computing framework.

What is **PySpark**:

Python Spark API that allows us to use Spark with Python
(our processing script is written in Python)

Parallel Processing I - PySpark

Total time used (crawling + cleaning), **without** NER and language detector
Parallelized using PySpark, default number of nodes
Unit: **seconds**

URL size	Baseline(s)	Parallel Stemming(s)	Parallel Crawling(s)
13	12.08366489	16.3642930984 21.1652750969	15.4960489273 15.6808428764 20.149545908
46	-	-	39.8726541996 26.4784898758
1000	650.2592189	714.77546	480.311785936 497.93241787

Parallel Crawling did show speed up but was most likely bottled necked by network constraints

Parallel Processing II - PySpark

Crawling time used
Paralleled using PySpark, default number of nodes
Unit: **seconds**

URL size	Parallel Crawling(s)
992	259.0064449
1000	288.0420101
2399	2097.015636

Parallel Processing III - Processes and Pipes

WARC Processing time used, **populating all fields**
Parallelized using Python multiprocessing module to create additional processes and pipes for communication
Unit: **minutes**

WARC File/Number of Records	No Parallelization on Local Machine	Parallelized on Local Machine	Parallelized on Cluster
Dunbar High Shooting / 259	11.83 minutes	10.74 minutes	6.68 minutes
Reynolds High Shooting / 522	23.09 minutes	21.91 minutes	13.75 minutes
VT Shooting / 1941	n/a	78.81 minutes	52.14 minutes

Parallel Crawling did show speed up but was most likely bottled necked by network constraints

Future Work

- Build up a single pipeline for all sources of inputs (WARC, EFC, tweets)
- Parallelized EFC (feasible using PySpark + URL queues)
- Speed up cleaning
- Phrases in NER (now separated words)

Acknowledgements

Our team would like to thank Dr. Fox for his guidance during this project. We would also like to thank Liuqing Li for answering our questions and helping us to get up and running. Our efforts contribute and benefit from the work done on the Global Event and Trend Archive Research project funded by NSF Grant IIS-1619028.