Tweet URL Analysis

Liyan Li, Kehan Lyu, Guoxin Sun
Client: Liuqing Li
Professor: Edward Alan Fox

May 2, 2018
## Contents

### List of Figures

<table>
<thead>
<tr>
<th>Figures</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>iv</td>
<td></td>
</tr>
</tbody>
</table>

### List of Tables

<table>
<thead>
<tr>
<th>Tables</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>iv</td>
<td></td>
</tr>
</tbody>
</table>

### 1 Abstract/Executive Summary

1

### 2 Objectives

2

#### 2.1 Objectives General

2

#### 2.2 Objectives Specifications

2

### 3 User Manual

3

#### 3.1 Discussion of the use environment

3

##### 3.1.1 Software Requirement and Dependencies

3

#### 3.2 Dataset

4

#### 3.3 Methodology

4

##### 3.3.1 Architecture

4

##### 3.3.2 Work-flow

5

#### 3.4 Results

6

##### 3.4.1 Keyword in URLs

6

##### 3.4.2 Tweets with URLs per year

7

##### 3.4.3 Number of URLs in Tweets

8

##### 3.4.4 Unique URLs in Tweets

9

##### 3.4.5 Unique URLs with different status codes

10

##### 3.4.6 Wayback Machine retrieved URLs per year

11

##### 3.4.7 Time interval between Tweet Post Date and Wayback Machine Archive Date

12

##### 3.4.8 Time interval between Web-page Post Date and Wayback Machine Archive Date

13

##### 3.4.9 Top-K domain names in all URLs

14

##### 3.4.10 Top-K domain names in unique URLs

16

##### 3.4.11 Top-K domain names in retweets

18

#### 3.5 Tutorials on use

20

### 4 Testing

21

#### 4.1 Approach

21

#### 4.2 Introduction of Testing Collection

21

#### 4.3 Results

21

#### 4.4 Interpretations of Results

22
5 Developer Manual 23
5.1 Inventory of all program files ........................................... 23
5.2 Tutorials on installing software to rebuild or makes changes ........................................... 23
  5.2.1 Python packages installation ........................................... 23
  5.2.2 Useful commands ........................................... 24

6 Reflections 26
6.1 Schedule ........................................... 26
  6.1.1 Role assignment ........................................... 26
  6.1.2 Team meeting ........................................... 26

7 Conclusions and Future Plans 27
7.1 Conclusions ........................................... 27
7.2 Future Plans/ Possible Improvement ........................................... 28
  7.2.1 Utilizing idle machines ........................................... 28
  7.2.2 Solutions for current issues ........................................... 29
    7.2.2.1 Sustained Internet Connection ........................................... 29
    7.2.2.2 Dirty URLs ........................................... 29
    7.2.2.3 Bad separator ........................................... 29
    7.2.2.4 Halt caused by using articleDateExtractor library ........................................... 30
  7.2.3 Analyzing more collections ........................................... 30

8 Acknowledgements 31

References 32

A Appendix 33
  A.1 Project Milestones ........................................... 33
  A.2 A Tweet Report File Example ........................................... 34
List of Figures

2. Simplified Work-flow for the Project .......................................................... 5
3. Percentage of the URL(s) with Keyword per year ........................................ 6
4. Percentage of Tweets with URLs per year ................................................. 7
5. Tweets with Different Number of URL(s) .................................................. 8
6. Percentage of Unique URL(s) in Tweet Collections .................................... 9
7. Percentage of Unique URL(s) with different status codes .......................... 10
8. Percentage of successful retrieved URL(s) per year ................................ 11
9. Time interval between Tweet Post Date and Wayback Machine Archive Date ... 12
10. Time interval between Webpage Post Date and Wayback Machine Archive Date 13
11. A typical checker result ........................................................................... 25
12. A snapshot of progress on each node ....................................................... 28

List of Tables

1. 9 Different Categories of Tweet Collections ............................................. 4
2. Top 10 domains in Nature category .......................................................... 14
3. Top 10 domains in Health category ........................................................... 14
4. Top 10 domains in Man-made category .................................................... 15
5. Top 10 domains in Particular category ...................................................... 15
6. Top 10 domains in unique URLs of Nature category ............................... 16
7. Top 10 domains in unique URLs of Health category ................................. 16
8. Top 10 domains in unique URLs of Man-made category .......................... 17
9. Top 10 domains in unique URLs of Particular category ............................ 17
10. Top 10 domains in retweets of Nature category ....................................... 18
11. Top 10 domains in retweets of Health category ....................................... 18
12. Top 10 domains in retweets of Man-made category ............................... 19
13. Top 10 domains in retweets of Particular category .................................. 19
14. The Fixed Test Results ........................................................................... 21
15. The Fluctuating Test Results .................................................................. 21
16. Top 10 Domains .................................................................................... 22
17. Inventory of all data files, program files ................................................ 23
1 Abstract/Executive Summary

The goal of the GETAR project is to devise interactive, integrated, digital library/archive systems coupled with linked and expert-curated web-page/tweet collections. In this class team project, the URL analysis system we designed takes a Tweet Collection as input and uses Hadoop and Spark to extract short URLs. We expanded them, fetched their web-page with the corresponding long URL, and applied the WayBack CDX Server API to attempt to restore the most likely snapshot. Then, we conducted a systematic URL analysis, for different types of events. We analyzed nine tweet collections in four categories: Nature, Health, Man-made, and Particular Event. Each tweet collection contains the tweet content from 2013-2017 that related to a specific keyword. For each collection, we analyzed several characteristics in URLs, top-k domains of the URLs, URL retrieve rate, and URL retrieve rate boosted by using the WayBack CDX Server API. We provided several visualizations of the results we analyzed from these nine tweet collections. We have refined this project so that it is easy to build on; see section 5 (Developer Manual) in the final report for details.
2 Objectives

2.1 Objectives General

Global Event and Trend Archive Research (GETAR) has been supported by NSF (IIS-1619028 and 1619371) starting last year. The goal of this project is to devise interactive, integrated, digital library/archive systems coupled with linked and expert-curated web-page/tweet collections. Currently, we had more than 1,400 tweet collections and over 2 billion tweets. Based on the previous research [1], there are about 30% of tweets with embedded URLs. Meanwhile, more than 50% of tweets have embedded URLs in our event-related collections. In this project, the URL analysis system we are designing takes a tweet collection as input and uses Hadoop and Spark to extract short URLs. We expanded them, fetched their web-page with the corresponding long URL, and applied the WayBack CDX Server API [2] to attempt to restore the most likely snapshot. Then, we conducted a systematic URL analysis, for different types of events.

2.2 Objectives Specifications

The following contents are the requirements from the client.

Basic Analysis

(1) Percentage of tweets with URLs
(2) Percentage of tweets with different number of URLs
(3) Percentage of unique URLs in all URLs
(4) Percentage of unique URLs with different status codes
(5) Percentage of URLs with code 200 per year
(6) Top-K domain names in all URLs
(7) Top-K domain names in unique URLs
(8) Top-K domain names in retweets

Advanced Analysis

(1) Percentage of unique URLs that can be retrieved
(2) Percentage of unique URLs that can be retrieved per year
(3) Distribution of time interval between tweet posted date and long URL date
(4) Distribution of time interval between tweet posted date and WayBack Machine nearest date
3 User Manual

In this section, we will discuss the environment, the input dataset, the methodology, and the results of this project. We will also include the tutorial on how to run our programs.

3.1 Discussion of the use environment

3.1.1 Software Requirement and Dependencies

Python version 2.7

Java Runtime version 1.7

3.1.1.1 Python packages

Beautiful Soup

readability

articleDateExtractor

Numpy

sciki-learn

3.1.2 Running Environment Requirement

Users need a Unix environment to install this system. The disk storage needs to be large enough to store Tweet Collections (50Gb for current Tweet Collections). Users need VT wireless access to upload Tweet Collections to the Hadoop Distributed File System. Using the Virginia Tech VPN service is not recommended since uploading large Tweet Collections requires a large upload bandwidth and a stable Internet connection.

For detailed installation guide, check Section 5.2.
3.2 Dataset

We chose nine Tweet Collections from the tweet archives provided by DLRL at Virginia Tech. Each Tweet Collection contains the tweet content from 2013-2017 that related to a specific keyword. Tweet Collections are categorized in four General Types: Nature, Health, Man-made, and Particular Event; see Table 1. The first three are general, while the fourth covers specific events.

Table 1: 9 Different Categories of Tweet Collections

<table>
<thead>
<tr>
<th>General Type</th>
<th>Keyword</th>
<th>Number of Tweets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nature</td>
<td>hurricane</td>
<td>10,520,692</td>
</tr>
<tr>
<td></td>
<td>typhoon</td>
<td>5,794,665</td>
</tr>
<tr>
<td>Health</td>
<td>obesity</td>
<td>6,244,587</td>
</tr>
<tr>
<td>Man-made</td>
<td>gun control</td>
<td>6,042,155</td>
</tr>
<tr>
<td></td>
<td>gun violence</td>
<td>3,920,488</td>
</tr>
<tr>
<td></td>
<td>terrorism</td>
<td>7,825,216</td>
</tr>
<tr>
<td>Particular Event</td>
<td>Hurricane Isaac</td>
<td>95,706</td>
</tr>
<tr>
<td></td>
<td>Hurricane Sandy</td>
<td>1,929,396</td>
</tr>
<tr>
<td></td>
<td>Connecticut school shooting</td>
<td>71,400</td>
</tr>
</tbody>
</table>

The name convention of the tweet collection file is Dataset_z_<id>_tweets.csv.

3.3 Methodology

This section will cover the architecture and the work-flow of this project.

3.3.1 Architecture

The input Tweet Collections, which are generated using yourTwapperKeeper[3] are in gigabyte-level. To handle this huge amount of data, we will use Hadoop and Spark distributed computing technologies. Since each Tweet Collection contains a large span of time, there will be some URLs that have expired. To retrieve the original web-pages, we will use WayBack Machine with a parameter indicating the time the tweet was post. The URL-extraction jobs are distributed among the Hadoop cluster. In order to speed up the URL processing and prevent frequent Internet access warning, we use 22 Virtual Machines (VMs) with different IP addresses to convert short URLs to long URLs and access the Wayback Machine Archives.
We used the URL Analysis System proposed by Li and Fox [1]; see Figure 1. We deployed yourTweeplerKepper [3] for tweet collections. For this project, we exported nine Tweet Collections from MySQL. The resulting raw file has fields like archivesource, text, and id. We then uploaded the file into our Hadoop cluster, using Bock’s framework [4] to extract short URLs. Each short URL record contains four fields: tweet id, re-tweet flag, tweet posted date, and short URL(s). Following that, we expanded short URLs into expanded long URLs (l_URLs). Using the WayBack CDX Server API [2], we retrieved snapshots (wb_URLs) for the URLs in the Internet Archive. We applied a URL cache to avoid duplicate processing.

![Figure 1: Architecture of the URL Analysis System [1]](image)

3.3.2 Work-flow

The work-flow can be simplified as Figure 2.

![Figure 2: Simplified Work-flow for the Project](image)
3.4 Results

3.4.1 Keyword in URLs

For each year, we calculated the percentage of URLs that contains the keyword of the Tweet collection; see Figure 3. There is no clear trend of changing percentage of keyword in URLs among different years. Most of the Tweet Collections have a similar percentage from 2013-2017. “Connecticut school shooting” and “Hurricane Issac” collections have a similar trend that the percentage of 2016 is clearly higher than those of other years. The percentage of 2016 in “Connecticut school shooting” collection is the highest compared to those of other collections, which is 64.4%.

![Figure 3: Percentage of the URL(s) with Keyword per year](image)
3.4.2 Tweets with URLs per year

We calculated the percentage of Tweets contain URLs for each year; see Figure 4. For most of the Tweet collections, around 50% of Tweets have URLs. The percentage of Tweets that contain URLs do not change a lot from year to year. However, we do observe a trend that people were more interested in embedding URLs in tweets from 2013-2015, and the interest faded away from 2015-2017.
### 3.4.3 Number of URLs in Tweets

![Figure 5: Tweets with Different Number of URL(s)](image)

For each Tweet Collection, we are interested in how many URLs are embedded in a tweet; see Figure 5. Of all the tweets with URLs, it is clear that the most of the tweets have one URL, which is 90% of all the tweets with URLs. Also, 10% of tweets have two URLs on average, and it is less than 1% of the tweets have three or more URLs embedded.
3.4.4 Unique URLs in Tweets

![Figure 6: Percentage of Unique URL(s) in Tweet Collections](image)

For each collection, we found the percentage of unique URLs; see Figure 6. In general, the **Nature** collections have a relatively high percentage compared to other collections, which are all above 15%. The **Particular Event** collections have relatively low percentages, which are all below 10%.
### 3.4.5 Unique URLs with different status codes

![Percentage of Unique URL(s) with different status codes](image)

For each collection, we found the percentage of unique URLs with different status codes; see Figure 7. The status codes are classified as successful responses (2xx), client error responses (4xx), and other responses. For all collections, the percentage of successful responses is the greatest, the percentage of client error responses are the second greatest, and the lowest is the other responses. Speaking of the specific percentage, the successful responses have percentages around 55% to 70%. The client error responses have percentages around 25% to 42%. The other responses have percentages around 1%.
3.4.6 Wayback Machine retrieved URLs per year

For each collection, we found the percentage of successful retrieved URLs from 2013 to 2017; see Figure 8. The nature event collection is less than the average level of other three event collections. Among other three event collections, Wayback Machine retrieved URLs in 2013 and 2014, where the sum of the percentages reached around 70% to 80%. URLs retrieved in 2017 are generally very low among all collections.
3.4.7 Time interval between Tweet Post Date and Wayback Machine Archive Date

For each collection, we found the time interval between the tweet date and the WayBack Machine archive date; see Figure 9. In general, most of the URLs are archived within the same day of the tweet post, which is around 27% to 37% of URLs. The chance of URLs archived within five days is also high, which is around 17% to 26%.
3.4.8 Time interval between Web-page Post Date and Wayback Machine Archive Date

Figure 10: Time interval between Webpage Post Date and Wayback Machine Archive Date

Figure 10 shows the time interval between long URL and Wayback Machine archived URLs for the “particular event” collection. Around 20% to 25% of the URLs were archived within one day, around 21% to 27% of URLs were archived within two days to five days, and 19% to 27% of URLs were archived after a year. However, the data is not completely accurate since crawling the date from the web-pages is not a safe approach. Sometimes date information is not in correct format, or the date information does not indicate the post date of the web-pages.
3.4.9 Top-K domain names in all URLs

Table 2: Top 10 domains in **Nature** category

<table>
<thead>
<tr>
<th>Rank</th>
<th>Domain</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>twitter.com</td>
<td>twitter.com</td>
</tr>
<tr>
<td>2</td>
<td><a href="http://www.youtube.com">www.youtube.com</a></td>
<td><a href="http://www.youtube.com">www.youtube.com</a></td>
</tr>
<tr>
<td>3</td>
<td><a href="http://www.nhc.noaa.gov">www.nhc.noaa.gov</a></td>
<td><a href="http://www.facebook.com">www.facebook.com</a></td>
</tr>
<tr>
<td>4</td>
<td><a href="http://www.instagram.com">www.instagram.com</a></td>
<td><a href="http://www.instagram.com">www.instagram.com</a></td>
</tr>
<tr>
<td>5</td>
<td><a href="http://www.facebook.com">www.facebook.com</a></td>
<td><a href="http://www.cnn.com">www.cnn.com</a></td>
</tr>
<tr>
<td>6</td>
<td>vine.co</td>
<td>fw.to</td>
</tr>
<tr>
<td>7</td>
<td>itunes.apple.com</td>
<td>mashable.com</td>
</tr>
<tr>
<td>8</td>
<td><a href="http://www.amazon.com">www.amazon.com</a></td>
<td><a href="http://www.usatoday.com">www.usatoday.com</a></td>
</tr>
<tr>
<td>9</td>
<td>weather.com</td>
<td>agora.ex.nii.ac.jp</td>
</tr>
<tr>
<td>10</td>
<td><a href="http://www.nytimes.com">www.nytimes.com</a></td>
<td>abcnews.go.com</td>
</tr>
</tbody>
</table>

Table 3: Top 10 domains in **Health** category

<table>
<thead>
<tr>
<th>Rank</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>twitter.com</td>
</tr>
<tr>
<td>2</td>
<td><a href="http://www.huffingtonpost.com">www.huffingtonpost.com</a></td>
</tr>
<tr>
<td>3</td>
<td><a href="http://www.theguardian.com">www.theguardian.com</a></td>
</tr>
<tr>
<td>4</td>
<td><a href="http://www.nytimes.com">www.nytimes.com</a></td>
</tr>
<tr>
<td>5</td>
<td>t.co</td>
</tr>
<tr>
<td>6</td>
<td>healthland.time.com</td>
</tr>
<tr>
<td>7</td>
<td>healthhabits.ca</td>
</tr>
<tr>
<td>8</td>
<td>healthhotsolution.blogspot.com</td>
</tr>
<tr>
<td>9</td>
<td><a href="http://www.youtube.com">www.youtube.com</a></td>
</tr>
<tr>
<td>10</td>
<td>wellblogs.nytimes.com</td>
</tr>
</tbody>
</table>
### Table 4: Top 10 domains in Man-made category

<table>
<thead>
<tr>
<th>Rank</th>
<th>Domain</th>
<th>Domain</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>twitter.com</td>
<td>twitter.com</td>
<td>twitter.com</td>
</tr>
<tr>
<td>2</td>
<td><a href="http://www.youtube.com">www.youtube.com</a></td>
<td><a href="http://www.huffingtonpost.com">www.huffingtonpost.com</a></td>
<td><a href="http://www.youtube.com">www.youtube.com</a></td>
</tr>
<tr>
<td>3</td>
<td><a href="http://www.breitbart.com">www.breitbart.com</a></td>
<td><a href="http://www.youtube.com">www.youtube.com</a></td>
<td><a href="http://www.amazon.com">www.amazon.com</a></td>
</tr>
<tr>
<td>4</td>
<td><a href="http://www.huffingtonpost.com">www.huffingtonpost.com</a></td>
<td><a href="http://www.motherjones.com">www.motherjones.com</a></td>
<td><a href="http://www.facebook.com">www.facebook.com</a></td>
</tr>
<tr>
<td>7</td>
<td><a href="http://www.washingtonpost.com">www.washingtonpost.com</a></td>
<td><a href="http://www.washingtonpost.com">www.washingtonpost.com</a></td>
<td>is.gd</td>
</tr>
<tr>
<td>9</td>
<td>atomiktiger.blogspot.com</td>
<td>t.co</td>
<td><a href="http://www.telegraph.co.uk">www.telegraph.co.uk</a></td>
</tr>
<tr>
<td>10</td>
<td>dailycaller.com</td>
<td><a href="http://www.theguardian.com">www.theguardian.com</a></td>
<td>linkis.com</td>
</tr>
</tbody>
</table>

### Table 5: Top 10 domains in Particular category

<table>
<thead>
<tr>
<th>Rank</th>
<th>Domain</th>
<th>Domain</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>twitter.com</td>
<td>twitter.com</td>
<td>patch.com</td>
</tr>
<tr>
<td>2</td>
<td><a href="http://www.theguardian.com">www.theguardian.com</a></td>
<td><a href="http://www.nhc.noaa.gov">www.nhc.noaa.gov</a></td>
<td>twitter.com</td>
</tr>
<tr>
<td>4</td>
<td>mashable.com</td>
<td><a href="http://www.youtube.com">www.youtube.com</a></td>
<td>apne.ws</td>
</tr>
<tr>
<td>6</td>
<td>RoyalRestrooms.com</td>
<td><a href="http://www.theguardian.com">www.theguardian.com</a></td>
<td><a href="http://www.youtube.com">www.youtube.com</a></td>
</tr>
<tr>
<td>7</td>
<td><a href="http://www.nola.com">www.nola.com</a></td>
<td><a href="http://www.gofundme.com">www.gofundme.com</a></td>
<td>reuters.us.feedsportal.com</td>
</tr>
<tr>
<td>9</td>
<td>weather.com</td>
<td><a href="http://www.nj.com">www.nj.com</a></td>
<td>people.com</td>
</tr>
</tbody>
</table>
3.4.10 Top-K domain names in unique URLs

Table 6: Top 10 domains in unique URLs of **Nature** category

<table>
<thead>
<tr>
<th>Rank</th>
<th>Domain</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>twitter.com</td>
<td>twitter.com</td>
</tr>
<tr>
<td>2</td>
<td><a href="http://www.instagram.com">www.instagram.com</a></td>
<td><a href="http://www.facebook.com">www.facebook.com</a></td>
</tr>
<tr>
<td>3</td>
<td><a href="http://www.youtube.com">www.youtube.com</a></td>
<td><a href="http://www.instagram.com">www.instagram.com</a></td>
</tr>
<tr>
<td>4</td>
<td><a href="http://www.facebook.com">www.facebook.com</a></td>
<td><a href="http://www.youtube.com">www.youtube.com</a></td>
</tr>
<tr>
<td>5</td>
<td><a href="http://www.nhc.noaa.gov">www.nhc.noaa.gov</a></td>
<td>gigaom.com</td>
</tr>
<tr>
<td>6</td>
<td>ask.fm</td>
<td>restorecosm.bid</td>
</tr>
<tr>
<td>7</td>
<td>vine.co</td>
<td>twib.in</td>
</tr>
<tr>
<td>8</td>
<td>restorecosm.bid</td>
<td>linkis.com</td>
</tr>
<tr>
<td>9</td>
<td><a href="http://www.swarmapp.com">www.swarmapp.com</a></td>
<td>t.co</td>
</tr>
<tr>
<td>10</td>
<td><a href="http://www.amazon.com">www.amazon.com</a></td>
<td><a href="http://www.google.com">www.google.com</a></td>
</tr>
</tbody>
</table>

Table 7: Top 10 domains in unique URLs of **Health** category

<table>
<thead>
<tr>
<th>Rank</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>twitter.com</td>
</tr>
<tr>
<td>2</td>
<td><a href="http://www.facebook.com">www.facebook.com</a></td>
</tr>
<tr>
<td>3</td>
<td><a href="http://www.google.com">www.google.com</a></td>
</tr>
<tr>
<td>4</td>
<td><a href="http://www.youtube.com">www.youtube.com</a></td>
</tr>
<tr>
<td>5</td>
<td><a href="http://www.instagram.com">www.instagram.com</a></td>
</tr>
<tr>
<td>6</td>
<td><a href="http://www.medicalnewstoday.com">www.medicalnewstoday.com</a></td>
</tr>
<tr>
<td>7</td>
<td><a href="http://www.sciencedaily.com">www.sciencedaily.com</a></td>
</tr>
<tr>
<td>8</td>
<td><a href="http://www.bioportfolio.com">www.bioportfolio.com</a></td>
</tr>
<tr>
<td>9</td>
<td>restorecosm.bid</td>
</tr>
<tr>
<td>10</td>
<td><a href="http://www.huffingtonpost.com">www.huffingtonpost.com</a></td>
</tr>
</tbody>
</table>
Table 8: Top 10 domains in unique URLs of **Man-made** category

<table>
<thead>
<tr>
<th>Rank</th>
<th>Domain</th>
<th>Domain</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>twitter.com</td>
<td>twitter.com</td>
<td>twitter.com</td>
</tr>
<tr>
<td>2</td>
<td><a href="http://www.youtube.com">www.youtube.com</a></td>
<td><a href="http://www.facebook.com">www.facebook.com</a></td>
<td><a href="http://www.youtube.com">www.youtube.com</a></td>
</tr>
<tr>
<td>3</td>
<td><a href="http://www.facebook.com">www.facebook.com</a></td>
<td>linkis.com</td>
<td><a href="http://www.facebook.com">www.facebook.com</a></td>
</tr>
<tr>
<td>4</td>
<td>linkis.com</td>
<td><a href="http://www.youtube.com">www.youtube.com</a></td>
<td>terrorism.trendolizer.com</td>
</tr>
<tr>
<td>5</td>
<td>restorecosm.bid</td>
<td><a href="http://www.huffingtonpost.com">www.huffingtonpost.com</a></td>
<td>linkis.com</td>
</tr>
<tr>
<td>6</td>
<td>t.co</td>
<td><a href="http://www.instagram.com">www.instagram.com</a></td>
<td>restorecosm.bid</td>
</tr>
<tr>
<td>7</td>
<td><a href="http://www.huffingtonpost.com">www.huffingtonpost.com</a></td>
<td><a href="http://www.google.com">www.google.com</a></td>
<td><a href="http://www.google.com">www.google.com</a></td>
</tr>
<tr>
<td>8</td>
<td><a href="http://www.rightrelevance.com">www.rightrelevance.com</a></td>
<td>restorecosm.bid</td>
<td>t.co</td>
</tr>
<tr>
<td>9</td>
<td><a href="http://www.google.com">www.google.com</a></td>
<td>t.co</td>
<td><a href="http://www.instagram.com">www.instagram.com</a></td>
</tr>
</tbody>
</table>

Table 9: Top 10 domains in unique URLs of **Particular** category

<table>
<thead>
<tr>
<th>Rank</th>
<th>Domain</th>
<th>Domain</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>twitter.com</td>
<td>twitter.com</td>
<td>twitter.com</td>
</tr>
<tr>
<td>2</td>
<td><a href="http://www.youtube.com">www.youtube.com</a></td>
<td><a href="http://www.nhc.noaa.gov">www.nhc.noaa.gov</a></td>
<td>patch.com</td>
</tr>
<tr>
<td>4</td>
<td><a href="http://www.facebook.com">www.facebook.com</a></td>
<td><a href="http://www.youtube.com">www.youtube.com</a></td>
<td><a href="http://www.youtube.com">www.youtube.com</a></td>
</tr>
<tr>
<td>5</td>
<td><a href="http://www.instagram.com">www.instagram.com</a></td>
<td>threadsphere.bid</td>
<td>apne.ws</td>
</tr>
<tr>
<td>8</td>
<td>louisianarecord.com</td>
<td><a href="http://www.huffingtonpost.com">www.huffingtonpost.com</a></td>
<td><a href="http://www.google.com">www.google.com</a></td>
</tr>
<tr>
<td>10</td>
<td>star94star.blogspot.com</td>
<td>patch.com</td>
<td>connecticut.news12.com</td>
</tr>
</tbody>
</table>
3.4.11 Top-K domain names in retweets

Table 10: Top 10 domains in retweets of Nature category

<table>
<thead>
<tr>
<th>Rank</th>
<th>Domain</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>twitter.com</td>
<td>twitter.com</td>
</tr>
<tr>
<td>2</td>
<td><a href="http://www.youtube.com">www.youtube.com</a></td>
<td>fw.to</td>
</tr>
<tr>
<td>3</td>
<td>vine.co</td>
<td><a href="http://www.youtube.com">www.youtube.com</a></td>
</tr>
<tr>
<td>4</td>
<td>itunes.apple.com</td>
<td><a href="http://www.usatoday.com">www.usatoday.com</a></td>
</tr>
<tr>
<td>5</td>
<td>weather.com</td>
<td><a href="http://www.cnn.com">www.cnn.com</a></td>
</tr>
<tr>
<td>6</td>
<td><a href="http://www.facebook.com">www.facebook.com</a></td>
<td>news.abs-cbn.com</td>
</tr>
<tr>
<td>7</td>
<td><a href="http://www.amazon.com">www.amazon.com</a></td>
<td><a href="http://www.facebook.com">www.facebook.com</a></td>
</tr>
<tr>
<td>8</td>
<td>us.news-you-need-to-know.com</td>
<td><a href="http://www.instagram.com">www.instagram.com</a></td>
</tr>
<tr>
<td>9</td>
<td><a href="http://www.nhc.noaa.gov">www.nhc.noaa.gov</a></td>
<td>abcnews.go.com</td>
</tr>
<tr>
<td>10</td>
<td>t.co</td>
<td>apne.ws</td>
</tr>
</tbody>
</table>

Table 11: Top 10 domains in retweets of Health category

<table>
<thead>
<tr>
<th>Rank</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>twitter.com</td>
</tr>
<tr>
<td>2</td>
<td><a href="http://www.theguardian.com">www.theguardian.com</a></td>
</tr>
<tr>
<td>3</td>
<td><a href="http://www.youtube.com">www.youtube.com</a></td>
</tr>
<tr>
<td>4</td>
<td><a href="http://www.nytimes.com">www.nytimes.com</a></td>
</tr>
<tr>
<td>5</td>
<td><a href="http://www.huffingtonpost.com">www.huffingtonpost.com</a></td>
</tr>
<tr>
<td>6</td>
<td>well.blogs.nytimes.com</td>
</tr>
<tr>
<td>7</td>
<td><a href="http://www.independent.co.uk">www.independent.co.uk</a></td>
</tr>
<tr>
<td>8</td>
<td><a href="http://www.medicalnewstoday.com">www.medicalnewstoday.com</a></td>
</tr>
<tr>
<td>9</td>
<td><a href="http://www.sciencedaily.com">www.sciencedaily.com</a></td>
</tr>
<tr>
<td>10</td>
<td>time.com</td>
</tr>
</tbody>
</table>
### Table 12: Top 10 domains in retweets of **Man-made** category

<table>
<thead>
<tr>
<th>Rank</th>
<th>gun control</th>
<th>gun violence</th>
<th>terrorism</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>twitter.com</td>
<td>twitter.com</td>
<td>twitter.com</td>
</tr>
<tr>
<td>2</td>
<td><a href="http://www.breitbart.com">www.breitbart.com</a></td>
<td><a href="http://www.huffingtonpost.com">www.huffingtonpost.com</a></td>
<td><a href="http://www.youtube.com">www.youtube.com</a></td>
</tr>
<tr>
<td>3</td>
<td><a href="http://www.youtube.com">www.youtube.com</a></td>
<td><a href="http://www.motherjones.com">www.motherjones.com</a></td>
<td><a href="http://www.amazon.com">www.amazon.com</a></td>
</tr>
<tr>
<td>4</td>
<td>atomiktiger.blogspot.com</td>
<td><a href="http://www.cnn.com">www.cnn.com</a></td>
<td>terrorism.trendolizer.com</td>
</tr>
<tr>
<td>5</td>
<td><a href="http://www.americanthinker.com">www.americanthinker.com</a></td>
<td>t.co</td>
<td><a href="http://www.theguardian.com">www.theguardian.com</a></td>
</tr>
<tr>
<td>7</td>
<td><a href="http://www.washingtonpost.com">www.washingtonpost.com</a></td>
<td><a href="http://www.vox.com">www.vox.com</a></td>
<td>t.co</td>
</tr>
<tr>
<td>8</td>
<td>dailycaller.com</td>
<td><a href="http://www.washingtonpost.com">www.washingtonpost.com</a></td>
<td><a href="http://www.washingtonpost.com">www.washingtonpost.com</a></td>
</tr>
<tr>
<td>9</td>
<td>t.co</td>
<td><a href="http://www.barackobama.com">www.barackobama.com</a></td>
<td><a href="http://www.independent.co.uk">www.independent.co.uk</a></td>
</tr>
<tr>
<td>10</td>
<td><a href="http://www.infowars.com">www.infowars.com</a></td>
<td>park.io</td>
<td><a href="http://www.telegraph.co.uk">www.telegraph.co.uk</a></td>
</tr>
</tbody>
</table>

### Table 13: Top 10 domains in retweets of **Particular** category

<table>
<thead>
<tr>
<th>Rank</th>
<th>hurricane issac</th>
<th>hurricane sandy</th>
<th>connecticut school shooting</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><a href="http://www.theguardian.com">www.theguardian.com</a></td>
<td>twitter.com</td>
<td><a href="http://www.nytimes.com">www.nytimes.com</a></td>
</tr>
<tr>
<td>2</td>
<td>twitter.com</td>
<td><a href="http://www.redgage.com">www.redgage.com</a></td>
<td><a href="http://www.cnn.com">www.cnn.com</a></td>
</tr>
<tr>
<td>4</td>
<td><a href="http://www.youtube.com">www.youtube.com</a></td>
<td><a href="http://www.youtube.com">www.youtube.com</a></td>
<td>perezhilton.com</td>
</tr>
<tr>
<td>5</td>
<td><a href="http://www.nola.com">www.nola.com</a></td>
<td><a href="http://www.theguardian.com">www.theguardian.com</a></td>
<td><a href="http://www.theguardian.com">www.theguardian.com</a></td>
</tr>
<tr>
<td>7</td>
<td>weather.com</td>
<td><a href="http://www.huffingtonpost.com">www.huffingtonpost.com</a></td>
<td><a href="http://www.reuters.com">www.reuters.com</a></td>
</tr>
<tr>
<td>9</td>
<td>vine.co</td>
<td><a href="http://www.politicususa.com">www.politicususa.com</a></td>
<td><a href="http://www.youtube.com">www.youtube.com</a></td>
</tr>
<tr>
<td>10</td>
<td><a href="http://www.washingtonexaminer.com">www.washingtonexaminer.com</a></td>
<td><a href="http://www.rollingstone.com">www.rollingstone.com</a></td>
<td>patch.com</td>
</tr>
</tbody>
</table>
3.5 Tutorials on use

This section provides a step-by-step tutorial on how to use our system.

1. Unzip the project.

2. Install all the required packages as mentioned in Section 3.1.

3. Put the collection you want to run into tweet_collection folder.

4. Go to the root directory of the project.

5. Upload the raw tweet file to the Hadoop cluster and start jobs on each VM.

   $ ./URL_push.sh <tweet collection id>

6. Check the VMs’ status occasionally till all nodes’ status become [FINISHED]; see the discussion in Section 7.2.1.

   $ ./URL_checker.sh

7. When all nodes’ status become [FINISHED], pull and merge the split long URLs files to the local machine.

   $ ./URL_pull.sh <tweet collection id>

8. Run statistic analysis.

   $ ./URL_Statistics.sh Dataset_z_<tweet collection id>_tweets_urls.tsv
4 Testing

In this section, we will discuss the testing procedure, results, and corresponding interpretation.

4.1 Approach

For the testing part, we planned to manually create a collection with a small number of tweets, so that we could control the results. By running the test collection, we checked the correctness of the result by comparing the test result with the ideal result. For some fixed values like the number of URLs and the number of unique URLs, we compared the exact values. For the unstable results like the number of URLs with different status codes, the results were acceptable if they were in the correct range. When we constantly access a web server, the response times vary from time to time. When the response time exceeds the threshold, we will stop accessing that web server. Therefore, some statistics fluctuated.

4.2 Introduction of Testing Collection

For the testing collection, we created a collection of 200 tweets from the Connecticut school shooting collection. This testing collection contains 100 tweets whose URL status code are 200, and 100 tweets whose URL status code are 404.

4.3 Results

The test results can be separated into three parts which are shown as tables below.

Table 14: The Fixed Test Results

<table>
<thead>
<tr>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Tweets</td>
</tr>
<tr>
<td>Number of Tweets with URL(s)</td>
</tr>
<tr>
<td>Number of Tweets with 1 URL</td>
</tr>
<tr>
<td>Number of Tweets with 2 URLs</td>
</tr>
<tr>
<td>Number of URLs</td>
</tr>
<tr>
<td>Number of Unique URLs</td>
</tr>
</tbody>
</table>

Table 15: The Fluctuating Test Results

<table>
<thead>
<tr>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of URLs with Code 0</td>
</tr>
<tr>
<td>Number of URLs with Code 200</td>
</tr>
<tr>
<td>Number of URLs with Code 403</td>
</tr>
<tr>
<td>Number of URLs with Code 400</td>
</tr>
</tbody>
</table>
Table 16: Top 10 Domains

<table>
<thead>
<tr>
<th>Domain</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>apne.ws</td>
<td>30</td>
</tr>
<tr>
<td><a href="http://www.youtube.com">www.youtube.com</a></td>
<td>28</td>
</tr>
<tr>
<td>survcast.com</td>
<td>15</td>
</tr>
<tr>
<td><a href="http://www.cnn.com">www.cnn.com</a></td>
<td>12</td>
</tr>
<tr>
<td><a href="http://www.thestar.com">www.thestar.com</a></td>
<td>9</td>
</tr>
<tr>
<td>curry.virginia.edu</td>
<td>7</td>
</tr>
<tr>
<td><a href="http://www.lifeofacatholicteen.com">www.lifeofacatholicteen.com</a></td>
<td>6</td>
</tr>
<tr>
<td><a href="http://www.ibosocial.com">www.ibosocial.com</a></td>
<td>5</td>
</tr>
<tr>
<td>feeds.feedburner.com</td>
<td>4</td>
</tr>
<tr>
<td>ictmax.org</td>
<td>4</td>
</tr>
</tbody>
</table>

4.4 Interpretations of Results

For the fixed test result, the testing results matches the expected results. Therefore, we passed the test.

For the fluctuating test results, since the web servers would be unstable when they were constantly accessed, the expected result for URLs with status code of 200 was around 50%, and the number of URLs with status code of 404 was around 35%. The result of number of URLs with status code of 200 was 55%, and the result of number of URLs with status of 404 was 33%. Both of them passed the test.

For the top 10 domain test, there are a series of top-10 values for different categories. Here we only picked top 10 domains as an example, and the results matched with our expected values. Therefore, this also passed the test.
5 Developer Manual

This section aims to help developers to continue working upon this project.

5.1 Inventory of all program files

The following table explains the inventory of all program files.

<table>
<thead>
<tr>
<th>File</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>add_key.sh</td>
<td>contains IP addresses of 22 VMS</td>
</tr>
<tr>
<td>server.list</td>
<td>shell script to add ssh keys</td>
</tr>
<tr>
<td>dis_in folder</td>
<td>contains split files, will be uploaded to 22 VMs for processing</td>
</tr>
<tr>
<td>dis_out folder</td>
<td>contains split files, downloaded from 22 VMs</td>
</tr>
<tr>
<td>src folder</td>
<td>contains the wayback_tweet_url.java file</td>
</tr>
<tr>
<td>dlrl-lib-latest.jar</td>
<td>framework for cleaning tweets on Hadoop</td>
</tr>
<tr>
<td>File_Helper.py</td>
<td>splitter and combiner for 22VMs</td>
</tr>
<tr>
<td>tweet_collection folder</td>
<td>contains raw tweets</td>
</tr>
<tr>
<td>tweet_s_url_collection folder</td>
<td>contains URLs [twee_id, RT, data, url_list]</td>
</tr>
<tr>
<td>tweet_l_url_collection folder</td>
<td>contains file used for reporting</td>
</tr>
<tr>
<td>tweet_report folder</td>
<td>contains final reports</td>
</tr>
<tr>
<td>URL_Compare.py</td>
<td>used to compare the similarity between two web-page contents</td>
</tr>
<tr>
<td>URL_Crawler.scala</td>
<td>runs with dlrl-*jar to extract URLs from tweets</td>
</tr>
<tr>
<td>URL_pipeline.sh</td>
<td>script for the automatic process</td>
</tr>
<tr>
<td>URL_push.sh</td>
<td>script used to upload file and start jobs on VMs</td>
</tr>
<tr>
<td>URL_pull.sh</td>
<td>script used to pull and merge distributed long_URLs to local machine</td>
</tr>
<tr>
<td>URL_VM_checker.sh</td>
<td>script used to heck VM status when expanding short URLs</td>
</tr>
<tr>
<td>URL_Statistics.py</td>
<td>used to create reports, using files in tweet_l_url_collection folder</td>
</tr>
<tr>
<td>wayback_tweet_url.jar</td>
<td>generates long URLs and Wayback Machine URLs</td>
</tr>
<tr>
<td>test_result.csv</td>
<td>the testing result</td>
</tr>
<tr>
<td>Report_Visualization.ipynb</td>
<td>Jupyter Notebook file used to construct data frame for visualization</td>
</tr>
</tbody>
</table>

5.2 Tutorials on installing software to rebuild or makes changes

5.2.1 Python packages installation

(1) Install BeautifulSoup

   $ pip install BeautifulSoup

(2) Install readability

   $ pip install readability-lxml
(3) Install articleDateExtractor (option 1)
   $ pip install articleDateExtractor

(4) Install articleDateExtractor (option 2)
   $ git clone https://github.com/Webhose/article-date-extractor
   $ cd article-date-extractor
   $ python2 setup.py install

(5) Install Numpy
   $ sudo pip install -U numpy

(6) Install all packages in NLTK
   $ python2 -m pip -H install -U nltk
   $ nltk.download("punkt")

(7) Install sciki-learn:
   $ pip2 install sciki-learn

5.2.2 Useful commands

(1) Change the access permission of a directory
   $ sudo chown -R $USER /absolute/path/to/directory

(2) List installed packages
   $ pip2 show <package_name>

(3) When pipeline crashed, try the following
   (a) ssh to the first node
   (b) Go to the project directory
       $ cd 2017s_tweet_url
   (c) Check the length of long.tsv and short.tsv for two times
       $ wc *.tsv
       $ wc *.tsv
       If the first column, which indicates the number of lines in the file, shows that long.tsv and short.tsv share the same number of lines, we can conclude that the job on this node has finished. Else, we can check if the results are changing. If two results are the same and the two files have different numbers of lines, we can spot a hang of job on this node; go to (d). If everything looks good, we can go to the next node and start from (b).
   (d) Delete the URL that causes the problem. Use the line number shown above in long.tsv to locate the harmful URL in the short.tsv.
   (e) Restart the job on the node
       $ nohup java -Xmx1024m -jar wayback_tweet_url.jar &
(4) Some tips on modifying `wayback_tweet_url.jar` file

The **jar** file we used in this project was compiled in Java 1.7. If the Java on your computer is not 1.7, you should change the compile environment. One option is changing the compile environment in the IDE. After the compiling, we get the corresponding **class** file. To get the new jar file, go to the **src folder**, where folders **vt** and **MATA-INF** are located, then go to `.\vt\dlrl`, substitute the old java file with the new **class** file you just created. Go back to the **src folder**, and run the following command.

```
$ jar cmvf META-INF/MANIFEST.MF wayback_tweet_url.jar  vt
```

(5) Kill processes on a node

(a) ssh to the node
(b) go to project directory
   ```
   $ cd 2017s_tweet_url/
   ```
(c) list all processes on the node, find the pids to kill
   ```
   $ ps aux | grep java
   ```
(d) kill the processes using their pids
   ```
   $ kill -9 pid
   ```

(6) Check VM status when expanding short URLs

After uploading the files and starting the jobs on the VMs, you can use the script **URL_VM_checker** to check the job on each VM.
```
$ ./URL_VM_checker.sh
```

This checker will return results as shown in Figure 11. On the Status column, [OK] means the node is running, [ERROR] means that the node is halted, and [FINISHED] means that the job is finished on that node.

```
Checking VM Status... This may take a minute.

<table>
<thead>
<tr>
<th>Node #</th>
<th># of URLs converted:</th>
<th>percentage finished</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>398</td>
<td>0.8890 %</td>
<td>[OK]</td>
</tr>
<tr>
<td>2</td>
<td>354</td>
<td>0.7907 %</td>
<td>[ERROR]</td>
</tr>
<tr>
<td>3</td>
<td>44770</td>
<td>100.0000 %</td>
<td>[FINISHED]</td>
</tr>
</tbody>
</table>

...  ...

Short URLs per node ~ 44770

Done! Time Cost: 1525148296 sec
```

Figure 11: A typical checker result
6 Reflections

In this section, we will discuss the lessons we learned from this project. It includes the schedule, difficulties that we encountered, and the corresponding solutions we applied and the future work.

6.1 Schedule

6.1.1 Role assignment

For this project, each member has different tasks. The detailed role assignment is listed below.

Guoxin Sun
- Team Leader
- Developer
- Completer Finisher
- URL Handling Lead

Kehan Lyu
- Resource Investigator
- Developer
- Presentation Lead
- Hadoop Cluster Lead

Liyan Li
- Coordinator
- Developer
- Report Lead
- Wayback Machine Lead

6.1.2 Team meeting

We normally held a meeting with the client every Thursday afternoon from 4:00 PM to 5:00 PM. We stuck to the plan, and finished our milestones on time. However, we did not have a decent estimation about the running time for large Tweet Collections. As the number of tweets increased, the running time increased to several days. Also, as we modified the system, we had to re-run the program multiple times which wasted a lot of time. Both reasons made it difficult to finish processing all 12 collections. We learned that, before we ran the whole dataset, we could create a test data file to test the correctness of the code. Unit testing could save us lots of time.
7 Conclusions and Future Plans

7.1 Conclusions

1. People were more interested in embedding URLs in tweets from 2013-2015, and the interest faded away from 2015-2017.

2. People usually only embedded one URL in a tweet, and it is rare that a tweet embedded three or more URLs.

3. The percentage of unique URLs is fairly low, which is around 20%. For some collections, the percentage is even below 10%.

4. The percentage of URLs with status code 200 is very high, which means most of the URLs are still hosted healthily.

5. The URLs in newer tweets have lower chance to be retrieved by Wayback Machine.

6. Wayback Machine is most likely to archive webpages within five days of the tweet post dates.

7. Wayback Machine also most likely to archive webpages within five days of the webpage post dates, but also likely to archive them after a year.

8. From Top-K domain analysis, we found that for all kinds of Tweet Collections, popular video sharing websites, news websites, and social media websites dominate the list. Only the top domains for none event-driven collection ”Obesity” also contain some keyword specific domains.
7.2 Future Plans/ Possible Improvement

7.2.1 Utilizing idle machines

As the architecture we discussed in section 3.3.1, we split the raw tweet file among 22 Virtual Machines. Each VM will have their own job to run. Thus, it makes sense that each VM/node will have their own progress; see Figure 12.

<table>
<thead>
<tr>
<th>Node #</th>
<th># of URLs converted:</th>
<th>percentage finished</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>241148</td>
<td>99.6600 %</td>
<td>[OK]</td>
</tr>
<tr>
<td>2</td>
<td>223191</td>
<td>92.2400 %</td>
<td>[OK]</td>
</tr>
<tr>
<td>3</td>
<td>241960</td>
<td>100.0000 %</td>
<td>[FINISHED]</td>
</tr>
<tr>
<td>4</td>
<td>241960</td>
<td>100.0000 %</td>
<td>[FINISHED]</td>
</tr>
<tr>
<td>5</td>
<td>241960</td>
<td>100.0000 %</td>
<td>[FINISHED]</td>
</tr>
<tr>
<td>6</td>
<td>241961</td>
<td>100.0000 %</td>
<td>[FINISHED]</td>
</tr>
<tr>
<td>7</td>
<td>241960</td>
<td>100.0000 %</td>
<td>[FINISHED]</td>
</tr>
<tr>
<td>8</td>
<td>241960</td>
<td>100.0000 %</td>
<td>[FINISHED]</td>
</tr>
<tr>
<td>9</td>
<td>241960</td>
<td>100.0000 %</td>
<td>[FINISHED]</td>
</tr>
<tr>
<td>10</td>
<td>241960</td>
<td>100.0000 %</td>
<td>[FINISHED]</td>
</tr>
<tr>
<td>11</td>
<td>241960</td>
<td>100.0000 %</td>
<td>[FINISHED]</td>
</tr>
<tr>
<td>12</td>
<td>241960</td>
<td>100.0000 %</td>
<td>[FINISHED]</td>
</tr>
<tr>
<td>13</td>
<td>241960</td>
<td>100.0000 %</td>
<td>[FINISHED]</td>
</tr>
<tr>
<td>14</td>
<td>240828</td>
<td>99.5300 %</td>
<td>[OK]</td>
</tr>
<tr>
<td>15</td>
<td>241960</td>
<td>100.0000 %</td>
<td>[FINISHED]</td>
</tr>
<tr>
<td>16</td>
<td>241960</td>
<td>100.0000 %</td>
<td>[FINISHED]</td>
</tr>
<tr>
<td>17</td>
<td>241961</td>
<td>100.0000 %</td>
<td>[FINISHED]</td>
</tr>
<tr>
<td>18</td>
<td>241960</td>
<td>100.0000 %</td>
<td>[FINISHED]</td>
</tr>
<tr>
<td>19</td>
<td>241960</td>
<td>100.0000 %</td>
<td>[FINISHED]</td>
</tr>
<tr>
<td>20</td>
<td>241960</td>
<td>100.0000 %</td>
<td>[FINISHED]</td>
</tr>
<tr>
<td>21</td>
<td>241960</td>
<td>100.0000 %</td>
<td>[FINISHED]</td>
</tr>
<tr>
<td>22</td>
<td>241960</td>
<td>100.0000 %</td>
<td>[FINISHED]</td>
</tr>
</tbody>
</table>

Figure 12: A snapshot of progress on each node

From the above figure, we can see that most of the VMs/nodes have finished their work and they are waiting for nodes 1, 2, and 14 to finish. Before all nodes finish their jobs, these idle nodes will not be able to handle new jobs, which is definitely a waste of resources, especially for a project that requires a significant number of data processing. One potential improvement would be pipelining the jobs and checking the status of the VMs constantly.
7.2.2 Solutions for current issues

7.2.2.1 Sustained Internet Connection

When the program is converting the short URLs to long URLs, the computer must also connect to the Internet to get the latest processing progress of each node. As the size of collection grows, it takes more than one day to finish one. However, there was no idle computer given to us to do this.

Solution

We did some tests of the program, and realized that when we close the computer, we only lost the connection between our local machines and nodes, but the node would still keep running the program and save the result to files. Once the file is on the Hadoop server, we can close the connection between the server and our local machine. This finding saved us time because we did not need to close and rerun the program when the computer was disconnected with the Internet.

7.2.2.2 Dirty URLs

When we were using Wayback Machine to retrieve the website backups, we found out that there was a very low percentage of URLs that could be retrieved from it when we were using the long URLs converted directly from short URLs.

Solution

We looked through the long URLs and find out that most of the URLs were dirty URLs, which are URLs contain the question mark, followed by parameters. When we removed the question mark and the followed parameters, it became retrievable. However, it is not true for all URLs. For example, YouTube only distinguishes the URLs by the parameters after question mark, which means if we simply removed all parameters in URLs, all of the YouTube URLs would be the same, which is wrong. If the WayBack Machine could retrieve an URL, we would keep it. If WayBack Machine could not retrieve an URL, we would clean the URL by removing the question mark and all of the characters followed by the question mark.

7.2.2.3 Bad separator

When creating the long URL file, the original code used double pipe (||) as the separator. However, when we parsed the URLs, we found out that the URLs sometimes can also contain double pipe, which will cause the error.

Solution

The solution we discussed with client is using JSON format. In this way, every section of information is formatted with name/value pair. So there is no way to cause unexpected error. But the drawback of this solution is to rewrite the statistics Python script and jar file which append the WayBack Machine URLs.
7.2.2.4 Halt caused by using *articleDateExtractor* library

When we run the statistics Python script to find the time interval between raw tweet and the publish date of the website the long URLs pointed to, we were using an open source Python module called *articleDateExtractor*. When we ran the program, sometimes it would cause a halt.

**Solution**

To avoid this problem, we set up the time limit of five seconds to raise the SIGALRM. In this way, once the function is spending more than five seconds for one URL, the signal handler will raise an exception, and the program will catch this exception and continue to the next URL.

7.2.3 Analyzing more collections

We had spent a big chunk of our time debugging the pipeline structure and fixing the intermediate data. With efforts made, we are now able to finish analyzing a tweet collection in a reasonable time. Using the current version of our system, we are able to handle various lengths of tweet collections in an efficient manner.
8 Acknowledgements

Thanks go to US NSF for its support of the Global Event and Trend Archive Research (GETAR) project, through grant IIS-1619028.

This project is supported by Liuqing Li, who is a graduate research assistant in DLRL (Digital Library Research Laboratory) at Virginia Tech.

We would also like to thank Liuqing for all of his help, insight and guidance, without which we would not have been able to accomplish the work we completed this semester.
References


A Appendix

A.1 Project Milestones

Milestone 1 - Complete Contract - 2/1
Milestone 2 - Environment setup - 2/2
Implement two basic functions - 2/12

(1) URL existence
(2) URL amount

Presentation 1 - 2/13

Milestone 4 - Implement a basic function - 2/19

(1) Top Domain URLs

Milestone 5 - Test and improve basic functions, and start writing report - 3/2

(1) Percentage of tweets with URLs
(2) Percentage of tweets with different number of URLs
(2) Percentage of unique URLs in all URLs
(3) Percentage of unique URLs with different status codes
(4) Percentage of URLs with code 200 per year
(5) Top-K domain names in all URLs
(6) Top-K domain names in unique URLs
(7) Top-K domain names in retweets

Milestone 6 - Learn Wayback Machine - 3/9
Milestone 7 - Brainstorm on advanced topics - 3/9

Presentation 2 - 3/20

Milestone 8 - Implement two advanced functions - 3/23

(1) Percentage of unique URLs that can be retrieved (200 vs. others)
(2) Percentage of unique URLs that can be retrieved per year

Presentation 3 - 4/3
Milestone 9 - Implement other advanced functions - 4/13

(1) Distribution of time interval between tweet posted date and Wayback Machine nearest date
(2) Distribution of time interval between tweet posted date and long URL date
(3) Distribution of similarity between tweet text and long URL content with code 200

Milestone 10 - Test and improve advanced functions, and write report - 4/20

(1) Percentage of unique URLs that can be retrieved (200 vs. others)
(2) Percentage of unique URLs that can be retrieved per year
(3) Distribution of time interval between tweet posted date and Wayback Machine nearest date
(4) Distribution of time interval between tweet posted date and long URL date
(5) Distribution of similarity between tweet text and long URL content with code 200

Milestone 11 - Project Wrap-up - 4/27

(1) Finish report
(2) Write comment and clean up the code

Final Presentation - 5/1

(1) Testing and assessment
(2) Deliverables and accomplishments
(3) Lessons learned and ideas for the future

A.2 A Tweet Report File Example

We got all of our visualizations from analyzing the final reports we generated. The following content shows an example of a final report. Every final report contains two parts. The first part is the overall statistics, and the second part is a collection of statistics for each year covered by the Tweet Collection. Each part of the report has its corresponding title which briefly introduces the meaning of the data.
Total  
# of Tweets 1525518  

The percentage of the URLs with keywords 'hurricane sandy' is 0.306680317843%

<table>
<thead>
<tr>
<th># of Tweets with URLs</th>
<th>772583</th>
<th>50.6%</th>
</tr>
</thead>
<tbody>
<tr>
<td># of Tweets with 1 URL(s)</td>
<td>687093</td>
<td>88.9%</td>
</tr>
<tr>
<td># of Tweets with 2 URL(s)</td>
<td>84101</td>
<td>10.9%</td>
</tr>
<tr>
<td># of Tweets with 3 URL(s)</td>
<td>1324</td>
<td>0.2%</td>
</tr>
<tr>
<td># of Tweets with 4 URL(s)</td>
<td>43</td>
<td>0.0%</td>
</tr>
<tr>
<td># of Tweets with 5 URL(s)</td>
<td>11</td>
<td>0.0%</td>
</tr>
<tr>
<td># of Tweets with 6 URL(s)</td>
<td>11</td>
<td>0.0%</td>
</tr>
<tr>
<td># of URLs</td>
<td>859560</td>
<td></td>
</tr>
<tr>
<td># of Unique URLs</td>
<td>83466</td>
<td>9.7%</td>
</tr>
<tr>
<td># of URLs with Code -1</td>
<td>2</td>
<td>0.0%</td>
</tr>
<tr>
<td>retrieved</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td># of URLs with Code 0</td>
<td>1</td>
<td>0.0%</td>
</tr>
<tr>
<td>retrieved</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td># of URLs with Code 200</td>
<td>8319</td>
<td>14.2%</td>
</tr>
<tr>
<td>retrieved</td>
<td>58657</td>
<td>70.3%</td>
</tr>
<tr>
<td># of URLs with Code 203</td>
<td>11</td>
<td>0.0%</td>
</tr>
<tr>
<td>retrieved</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td># of URLs with Code 300</td>
<td>1</td>
<td>0.0%</td>
</tr>
<tr>
<td>retrieved</td>
<td>1</td>
<td>100.0%</td>
</tr>
<tr>
<td># of URLs with Code 301</td>
<td>16</td>
<td>0.0%</td>
</tr>
<tr>
<td>retrieved</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td># of URLs with Code 302</td>
<td>15</td>
<td>0.0%</td>
</tr>
<tr>
<td>retrieved</td>
<td>5</td>
<td>33.3%</td>
</tr>
<tr>
<td># of URLs with Code 307</td>
<td>1</td>
<td>0.0%</td>
</tr>
<tr>
<td>retrieved</td>
<td>1</td>
<td>100.0%</td>
</tr>
<tr>
<td># of URLs with Code 400</td>
<td>664</td>
<td>0.8%</td>
</tr>
<tr>
<td>retrieved</td>
<td>68</td>
<td>10.2%</td>
</tr>
<tr>
<td># of URLs with Code 401</td>
<td>20</td>
<td>0.0%</td>
</tr>
<tr>
<td>retrieved</td>
<td>5</td>
<td>25.0%</td>
</tr>
<tr>
<td># of URLs with Code 402</td>
<td>4</td>
<td>0.0%</td>
</tr>
<tr>
<td>retrieved</td>
<td>2</td>
<td>50.0%</td>
</tr>
<tr>
<td># of URLs with Code 403</td>
<td>12098</td>
<td>14.5%</td>
</tr>
<tr>
<td>retrieved</td>
<td>2689</td>
<td>22.2%</td>
</tr>
<tr>
<td># of URLs with Code 404</td>
<td>10554</td>
<td>12.6%</td>
</tr>
<tr>
<td>retrieved</td>
<td>734</td>
<td>7.0%</td>
</tr>
<tr>
<td># of URLs with Code 405</td>
<td>124</td>
<td>0.1%</td>
</tr>
<tr>
<td>retrieved</td>
<td>28</td>
<td>22.6%</td>
</tr>
<tr>
<td># of URLs with Code 406</td>
<td>24</td>
<td>0.0%</td>
</tr>
<tr>
<td>retrieved</td>
<td>14</td>
<td>58.3%</td>
</tr>
<tr>
<td># of URLs with Code 410</td>
<td>216</td>
<td>0.3%</td>
</tr>
<tr>
<td>retrieved</td>
<td>20</td>
<td>9.3%</td>
</tr>
<tr>
<td># of URLs with Code 416</td>
<td>97</td>
<td>0.1%</td>
</tr>
<tr>
<td>retrieved</td>
<td>21</td>
<td>21.6%</td>
</tr>
<tr>
<td># of URLs with Code 429</td>
<td>137</td>
<td>0.2%</td>
</tr>
<tr>
<td>retrieved</td>
<td>1</td>
<td>100.0%</td>
</tr>
</tbody>
</table>
retrieved 4 2.9%
# of URLs with Code 430 retrieved 1 50.0%
# of URLs with Code 479 retrieved 0 0%
# of URLs with Code 500 retrieved 23 7.4%
# of URLs with Code 502 retrieved 21 42.9%
# of URLs with Code 503 retrieved 25 5.7%
# of URLs with Code 504 retrieved 0 0%
# of URLs with Code 505 retrieved 0 0%
# of URLs with Code 999 retrieved 0 0%
# of URLs with Code 999 retrieved 0 0%

Time between Raw Tweet and Wayback URL <= 1 Days: 26.6%
Time between Raw Tweet and Wayback URL <= 5 Days: 21.2%
Time between Raw Tweet and Wayback URL <= 10 Days: 5.8%
Time between Raw Tweet and Wayback URL <= 30 Days: 10.1%
Time between Raw Tweet and Wayback URL <= 90 Days: 9.6%
Time between Raw Tweet and Wayback URL <= 180 Days: 7.1%
Time between Raw Tweet and Wayback URL <= 365 Days: 6.9%
Time between Raw Tweet and Wayback URL > 365 Days: 12.8%

Time between long URL and Wayback URL <= 1 Days: 25.0%
Time between long URL and Wayback URL <= 5 Days: 24.6%
Time between long URL and Wayback URL <= 10 Days: 5.0%
Time between long URL and Wayback URL <= 30 Days: 7.5%
Time between long URL and Wayback URL <= 90 Days: 7.1%
Time between long URL and Wayback URL <= 180 Days: 5.0%
Time between long URL and Wayback URL <= 365 Days: 6.0%
Time between long URL and Wayback URL > 365 Days: 19.8%

Percentage of unique URLs that can be retrieved (200 vs. others) 14.35%

Top 10 URLs
Top 10 Domains
1. twitter.com 138376
2. www.nhc.noaa.gov 44846
3. www.redgage.com 39842
4. www.youtube.com 39817
5. www.nytimes.com 29731
6. www.theguardian.com 9853
7. www.gofundme.com 9348
8. www.huffingtonpost.com 9300
9. www.nj.com 8147
10. www.facebook.com 7792

Top 10 Domains in retweets
1. twitter.com 38195
2. www.redgage.com 18116
4. www.youtube.com 5010
5. www.theguardian.com 4467
6. www.nhc.noaa.gov 4337
7. www.huffingtonpost.com 3481
8. www.nydailynews.com 3428
9. www.politicususa.com 3256
10. www.rollingstone.com 2847

Top 10 Wayback URLs
http://www.redgage.com/photos/Kinderhook/hurricane-sandy-utility-trucks-head-to-baltimore.html 8594
http://www.redgage.com/photos/Kinderhook/waiting-for-hurricane-sandy.html 8131
http://www.redgage.com/photos/Kinderhook/did-you-need-to-buy-some-bread.html 7236
http://ysear.ch/11M 6549
https://twitter.com/ericawerner/status/824408284711047168 5405
http://streaming.radionomy.com/JamendoLounge 3035
http://www.rollingstone.com:80/culture/features/rockaway-beach-
surfing-rebels-restore-after-hurricane-sandy-w478999 ...

Year: 2013
# of Tweets  258435

The percentage of the URLs with keywords 'hurricane sandy' is
0.289185767158%

# of Tweets with URLs  123208  47.7%
# of Tweets with 1 URL(s)  118500  96.2%
# of Tweets with 2 URL(s)  4604  3.7%
# of Tweets with 3 URL(s)  99  0.1%
# of Tweets with 4 URL(s)  2  0.0%
# of Tweets with 5 URL(s)  3  0.0%
# of URLs  128028
# of Unique URLs  30491  23.8%
# of URLs with Code -1  2  0.0% # of URLs retrieved  0  0%
# of URLs with Code  0  1  0.0% # of URLs retrieved  0  0%
# of URLs with Code 200  17123  56.2% # of URLs retrieved  3803  22.2%
# of URLs with Code 203  3  0.0% # of URLs retrieved  0  0%
# of URLs with Code 300  1  0.0% # of URLs retrieved  1  100.0%
# of URLs with Code 301  16  0.1% # of URLs retrieved  0  0%
# of URLs with Code 302  6  0.0% # of URLs retrieved  1  16.7%
# of URLs with Code 307  1  0.0% # of URLs retrieved  0  0%
# of URLs with Code 400  372  1.2% # of URLs retrieved  32  8.6%
# of URLs with Code 401  11  0.0% # of URLs retrieved  3  27.3%
# of URLs with Code 403  6618  21.7% # of URLs retrieved  1232  18.6%
# of URLs with Code 404  5756  18.9% # of URLs retrieved  425  7.4%
# of URLs with Code 405  39  0.1% # of URLs retrieved  14  35.9%
# of URLs with Code 406  8  0.0% # of URLs retrieved  2  25.0%
# of URLs with Code 410  89  0.3% # of URLs retrieved  12  13.5%
# of URLs with Code 416  73  0.2% # of URLs retrieved  17  23.3%
# of URLs with Code 429          56 0.2% # of URLs retrieved 1 1.8%
# of URLs with Code 430           2 0.0% # of URLs retrieved 1 50.0%
# of URLs with Code 479           3 0.0% # of URLs retrieved 0 0%
# of URLs with Code 500         113 0.4% # of URLs retrieved 14 12.4%
# of URLs with Code 502          14 0.0% # of URLs retrieved 0 0%
# of URLs with Code 503         163 0.5% # of URLs retrieved 11 6.7%
# of URLs with Code 504           3 0.0% # of URLs retrieved 0 0%
# of URLs with Code 505          17 0.1% # of URLs retrieved 0 0%
# of URLs with Code 999           1 0.0% # of URLs retrieved 0 0%

Percentage of unique URLs that can be retrieved (200 vs. others) 18.29%

Top 10 URLs
1. https://www.youtube.com/watch?v=CC6UU9BwM3Y 832
3. https://twitter.com/MilitaryPorn/status/371708653579272192/photo/1 599
4. http://www.stumbleupon.com 552

Top 10 Domains
1. www.youtube.com 6697
2. twitter.com 5845
3. www.nytimes.com 3369
4. www.huffingtonpost.com 3272
5. www.facebook.com 2753
6. threadsphere.bid 2632
7. mashable.com 2073
8. www.instagram.com 1778
9. www.nj.com 1680
10. www.nhc.noaa.gov 1671
Top 10 Domains in retweets
1. twitter.com 2469
2. www.youtube.com 1182
3. www.huffingtonpost.com 1141
4. t.co 1105
5. www.nytimes.com 926
6. twitpic.com 655
7. www.nbcphiladelphia.com 354
8. www.instagram.com 339
9. www.theatlantic.com 300
10. www.nj.com 278

Top 10 Wayback URLs
http://mashable.com/2012/10/27/emergency-apps/ 560
http://www.youtube.com/watch?v=CC6UU9BwM3Y 510
http://twitpic.com/80/bqmevu 465
http://www.stumbleupon.com/ 408
https://twitter.com/OMGFacts/status/262955515401863168/photo/1 251
http://www.nj.com/ 239
https://www.youtube.com/watch?v=k3RCMZqZ5uE 230
http://www.engadget.com/2013/10/30/google-donates-17000-nexus-7-tablets/?ncid=rss_truncated 208

-------------------
Year: 2014
# of Tweets 117177

The percentage of the URLs with keywords 'hurricane sandy' is 0.322305811255%

<table>
<thead>
<tr>
<th># of Tweets with URLs</th>
<th>64203</th>
<th>54.8%</th>
</tr>
</thead>
<tbody>
<tr>
<td># of Tweets with 1 URL(s)</td>
<td>60181</td>
<td>93.7%</td>
</tr>
<tr>
<td># of Tweets with 2 URL(s)</td>
<td>3937</td>
<td>6.1%</td>
</tr>
<tr>
<td># of Tweets with 3 URL(s)</td>
<td>72</td>
<td>0.1%</td>
</tr>
<tr>
<td># of Tweets with 4 URL(s)</td>
<td>1</td>
<td>0.0%</td>
</tr>
<tr>
<td># of Tweets with 5 URL(s)</td>
<td>1</td>
<td>0.0%</td>
</tr>
<tr>
<td># of Tweets with 6 URL(s)</td>
<td>11</td>
<td>0.0%</td>
</tr>
<tr>
<td># of URLs</td>
<td>68346</td>
<td></td>
</tr>
<tr>
<td># of Unique URLs</td>
<td>14420</td>
<td>21.1%</td>
</tr>
<tr>
<td># of URLs with Code 0</td>
<td>1</td>
<td>0.0%</td>
</tr>
<tr>
<td># of URLs retrieved 0</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td># of URLs with Code 200</td>
<td>9836</td>
<td>68.2%</td>
</tr>
<tr>
<td># of URLs retrieved 2059</td>
<td>20.9%</td>
<td></td>
</tr>
<tr>
<td># of URLs with Code 302</td>
<td>3</td>
<td>0.0%</td>
</tr>
<tr>
<td># of URLs retrieved 1</td>
<td>33.3%</td>
<td></td>
</tr>
<tr>
<td># of URLs with Code 400</td>
<td>97</td>
<td>0.7%</td>
</tr>
<tr>
<td># of URLs retrieved 16</td>
<td>16.5%</td>
<td></td>
</tr>
<tr>
<td>Code</td>
<td>Total # of URLs</td>
<td>Retrieved # of URLs</td>
</tr>
<tr>
<td>--------</td>
<td>----------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>401</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>403</td>
<td>2391</td>
<td>679</td>
</tr>
<tr>
<td>404</td>
<td>1737</td>
<td>132</td>
</tr>
<tr>
<td>405</td>
<td>18</td>
<td>7</td>
</tr>
<tr>
<td>406</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>410</td>
<td>54</td>
<td>6</td>
</tr>
<tr>
<td>416</td>
<td>15</td>
<td>5</td>
</tr>
<tr>
<td>429</td>
<td>14</td>
<td>0</td>
</tr>
<tr>
<td>500</td>
<td>163</td>
<td>2</td>
</tr>
<tr>
<td>502</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>503</td>
<td>75</td>
<td>13</td>
</tr>
<tr>
<td>504</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>505</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>505</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Percentage of unique URLs that can be retrieved (200 vs. others) 20.28%

Top 10 URLs
1. https://twitter.com/StormEffects/status/440955416734752768/photo/1 687
9. https://www.youtube.com/watch?v=zNnSyglMuxc 327
10. https://www.youtube.com/watch?v=BvpAuu548gg 327
Top 10 Domains
1. twitter.com 6580
2. www.youtube.com 3900
3. www.nytimes.com 2624
4. www.nhc.noaa.gov 2411
5. www.huffingtonpost.com 1597
6. www.redgage.com 1436
7. www.nj.com 1394
8. rss.nytimes.com 841
9. www.rawstory.com 840
10. www.nydailynews.com 832

Top 10 Domains in retweets
1. twitter.com 3726
2. www.nytimes.com 1156
3. www.huffingtonpost.com 829
4. www.rawstory.com 674
5. www.youtube.com 546
6. www.nj.com 541
7. www.redgage.com 492
8. www.theguardian.com 363
10. www.axs.com 247

Top 10 Wayback URLs
http://www.rawstory.com/rs/2014/10/hurricane-sandy-survivor-chris-christie-is-sitting-on-800-million-meant-for-disaster-relief/ 730
http://www.redgage.com/photos/Kinderhook/waiting-for-hurricane-sandy.html 355
http://www.axs.com/ 352
http://ysear.ch/11M 344
https://www.youtube.com/watch?v=0lRkDUVlr80 326
http://www.redgage.com/photos/Kinderhook/did-you-need-to-buy-some-bread.html 288

-----------------------------------------------
Year: 2015
# of Tweets 309470

The percentage of the URLs with keywords 'hurricane sandy' is 0.309361910461%
<table>
<thead>
<tr>
<th># of Tweets with URLs</th>
<th></th>
<th>59.4%</th>
</tr>
</thead>
<tbody>
<tr>
<td># of Tweets with 1 URL(s)</td>
<td>153700</td>
<td>83.6%</td>
</tr>
<tr>
<td># of Tweets with 2 URL(s)</td>
<td>29561</td>
<td>16.1%</td>
</tr>
<tr>
<td># of Tweets with 3 URL(s)</td>
<td>551</td>
<td>0.3%</td>
</tr>
<tr>
<td># of Tweets with 4 URL(s)</td>
<td>12</td>
<td>0.0%</td>
</tr>
<tr>
<td># of Tweets with 5 URL(s)</td>
<td>6</td>
<td>0.0%</td>
</tr>
<tr>
<td># of URLs</td>
<td>214553</td>
<td></td>
</tr>
<tr>
<td># of Unique URLs</td>
<td>14157</td>
<td>6.6%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th># of URLs with Code</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0.0%</td>
</tr>
<tr>
<td>200</td>
<td>10925</td>
<td>77.2%</td>
</tr>
<tr>
<td>203</td>
<td>1</td>
<td>0.0%</td>
</tr>
<tr>
<td>302</td>
<td>7</td>
<td>0.0%</td>
</tr>
<tr>
<td>400</td>
<td>98</td>
<td>0.7%</td>
</tr>
<tr>
<td>403</td>
<td>1533</td>
<td>10.8%</td>
</tr>
<tr>
<td>404</td>
<td>1348</td>
<td>9.5%</td>
</tr>
<tr>
<td>405</td>
<td>37</td>
<td>0.3%</td>
</tr>
<tr>
<td>406</td>
<td>1532</td>
<td>10.8%</td>
</tr>
<tr>
<td>410</td>
<td>28</td>
<td>0.2%</td>
</tr>
<tr>
<td>416</td>
<td>2</td>
<td>0.2%</td>
</tr>
<tr>
<td>500</td>
<td>20</td>
<td>0.1%</td>
</tr>
<tr>
<td>502</td>
<td>11</td>
<td>0.1%</td>
</tr>
<tr>
<td>503</td>
<td>37</td>
<td>0.3%</td>
</tr>
<tr>
<td>505</td>
<td>9</td>
<td>0.1%</td>
</tr>
<tr>
<td>506</td>
<td>2</td>
<td>0.2%</td>
</tr>
</tbody>
</table>

Percentage of unique URLs that can be retrieved (200 vs. others): 13.84%

Top 10 URLs
1. https://www.gofundme.com/dgreig    7729

Top 10 Domains
1. twitter.com 34246
2. www.nhc.noaa.gov 11582
3. www.youtube.com 11568
4. www.redgage.com 11198
5. www.gofundme.com 8674
7. www.politicususa.com 4152
8. rss.nytimes.com 2756
9. www.huffingtonpost.com 2589
10. streaming.radionomy.com 2439

Top 10 Domains in retweets
1. twitter.com 9731
2. www.redgage.com 6720
3. www.politicususa.com 3111
4. www.nytimes.com 1887
5. www.youtube.com 1611
6. news.nationalgeographic.com 934
7. www.fema.gov 888
8. www.amazon.com 845
9. www.huffingtonpost.com 761
10. nymag.com 711

Top 10 Wayback URLs
http://www.redgage.com/photos/Kinderhook/hurricane-sandy-utility-trucks-head-to-baltimore.html 3327
http://www.redgage.com/photos/Kinderhook/waiting-for-hurricane-sandy.html 3000
http://streaming.radionomy.com/JamendoLounge 2439
The percentage of the URLs with keywords 'hurricane sandy' is 0.289147827242%

# of Tweets with URLs 210861 52.0%
# of Tweets with 1 URL(s) 184006 87.3%
# of Tweets with 2 URL(s) 26425 12.5%
# of Tweets with 3 URL(s) 402 0.2%
# of Tweets with 4 URL(s) 27 0.0%
# of Tweets with 5 URL(s) 1 0.0%
# of URLs 238175
# of Unique URLs 12387 5.2%
# of URLs with Code 0 retrieved 1 0.0% # of URLs
# of URLs with Code 200 retrieved 10436 84.2% # of URLs
# of URLs with Code 203 retrieved 4 0.0% # of URLs
# of URLs with Code 400 retrieved 86 0.7% # of URLs
# of URLs with Code 401 retrieved 6 7.0%
# of URLs with Code 402 retrieved 25 25.0%
# of URLs with Code 403 retrieved 220 27.4%
# of URLs with Code 404 retrieved 838 6.8% # of URLs
# of URLs with Code 405 retrieved 50 0.4% # of URLs
# of URLs with Code 406 retrieved 3 6.0%
# of URLs with Code 410 retrieved 25 0.2% # of URLs
# of URLs with Code 416 retrieved 2 0.0% # of URLs
# of URLs with Code 429 retrieved 31 0.3% # of URLs
# of URLs with Code 500 retrieved 13 0.1% # of URLs
# of URLs with Code 502          17  0.1% # of URLs retrieved 10  58.8%
# of URLs with Code 503          71  0.6% # of URLs retrieved 1  1.4%
# of URLs with Code 999           1  0.0% # of URLs retrieved 0  0%

Percentage of unique URLs that can be retrieved (200 vs. others) 9.82%

Top 10 URLs
10. https://www.youtube.com/watch?v=zNnSygImuxc  849

Top 10 Domains
1. twitter.com  40858
2. www.nhc.noaa.gov  21006
3. www.redgage.com  16255
4. www.youtube.com  11242
5. www.theguardian.com  4878
7. www.forbes.com  2556
8. www.nydailynews.com  2514
9. ysear.ch  2487
10. www.washingtonpost.com  1877

Top 10 Domains in retweets
1. twitter.com  9185
2. www.redgage.com  7365
3. www.theguardian.com  2390
4. www.nhc.noaa.gov  2140
5. www.nydailynews.com  1961
7. natl.re 1221
8. www.washingtonpost.com 1105
9. www.thedailybeast.com 1096
10. thinkprogress.org 748

Top 10 Wayback URLs
http://www.redgage.com/photos/Kinderhook/hurricane-sandy-utility-trucks-head-to-baltimore.html 4547
http://www.redgage.com/photos/Kinderhook/waiting-for-hurricane-sandy.html 4515
http://www.redgage.com/photos/Kinderhook/did-you-need-to-buy-some-bread.html 2909
http://ysear.ch/11M 2487

-------

Year: 2017
# of Tweets 434790

The percentage of the URLs with keywords 'hurricane sandy' is 0.329549928864%

# of Tweets with URLs 190481 43.8%
# of Tweets with 1 URL(s) 170706 89.6%
# of Tweets with 2 URL(s) 19574 10.3%
# of Tweets with 3 URL(s) 200 0.1%
# of Tweets with 4 URL(s) 1 0.0%
# of URLs 210458
# of Unique URLs 14023 6.7%
# of URLs with Code 0 1 0.0% # of URLs retrieved 0 0%
# of URLs with Code 200 11937 85.1% # of URLs retrieved 883 7.4%
# of URLs with Code 203 3 0.0% # of URLs retrieved 0 0%
# of URLs with Code 400 34 0.2% # of URLs retrieved 12 35.3%
# of URLs with Code 401 1 0.0% # of URLs retrieved 0 0%
# of URLs with Code 402 2 0.0% # of URLs retrieved 0 0%
# of URLs with Code 403 1042 7.4% # of URLs retrieved
retrieved 292 28.0%
# of URLs with Code 404         949  6.8% # of URLs
retrieved  23  2.4%
# of URLs with Code 405          11  0.1% # of URLs
retrieved   4 36.4%
# of URLs with Code 410          13  0.1% # of URLs
retrieved   0  0%
# of URLs with Code 416           1  0.0% # of URLs
retrieved   0  0%
# of URLs with Code 429           8  0.1% # of URLs
retrieved   0  0%
# of URLs with Code 500           3  0.0% # of URLs
retrieved   0  0%
# of URLs with Code 502           2  0.0% # of URLs
retrieved   1 50.0%
# of URLs with Code 503          15  0.1% # of URLs
retrieved   1  6.7%
# of URLs with Code 505          1  0.0% # of URLs
retrieved   0  0%
# of URLs with Code 505           1  0.0% # of URLs
retrieved   0  0%

Percentage of unique URLs that can be retrieved (200 vs. others) 8.67%

Top 10 URLs
1. https://twitter.com/ericawerner/status/824408284711047168 5405

Top 10 Domains
1. twitter.com 50847
2. www.nytimes.com 15554
3. www.redgage.com 10937
4. www.nhc.noaa.gov 8176
1. twitter.com 13084
2. www.nytimes.com 3633
3. www.redgage.com 3537
4. www.rollingstone.com 2640
5. www.nhc.noaa.gov 1985
6. www.snjtoday.com 1731
7. ny.curbed.com 1079
8. www.youtube.com 1064
9. www.theguardian.com 968
10. ref.gl 707

Top 10 Wayback URLs
https://twitter.com/ericawerner/status/824408284711047168 5405
http://ysear.ch/11M 2365
http://www.redgage.com/photos/Kinderhook/did-you-need-to-buy-some-bread.html 2219
https://www.nhc.noaa.gov/gtwo.php?basin=atlc 1676

-------------------------------
--------------