

Digital Library Assignment 2

Q1)

WARC can be created using following four tools wget, WARCreate, Heritrix and webrecorder.io.

wget was the simplest to implement. It was just one line of code as below :-

...

```
os.system("wget '%s' --warc-file='%s' --no-warc-compression" % (line, name))
```

...

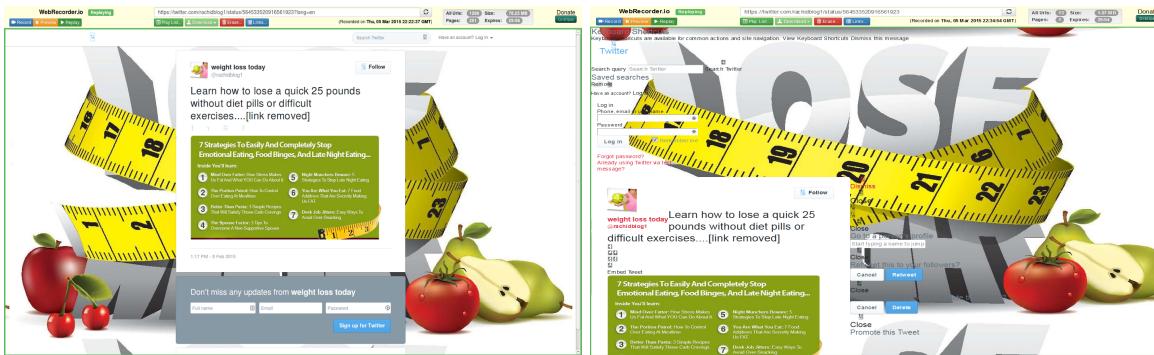
WARCreate is a chrome plugin to create a warc of current file. It was not creating WARC's for certain urls and from what I observed it was not doing it for pages that have deep links like imdb.

Heritrix warc was done using WAIL. Setting up wail took a lot of time. Wail configuration was also not straightforward. It was taking time to create WARC for certain urls as they had many sub pages. So later on a configuration was added to heritrix to restrict by time size and documents in heritrix job config file.

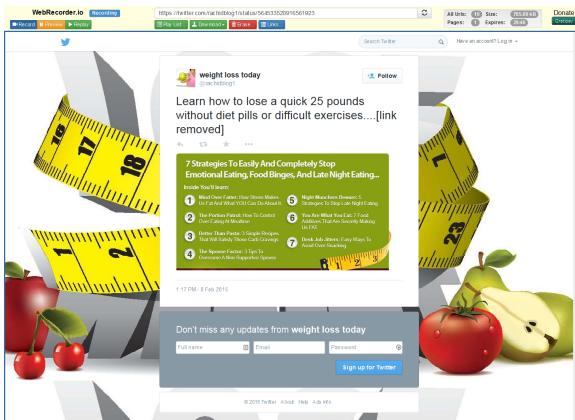
Webrecorder.io is an online tool for creating and replaying WARC file. It was simple to use. In Webrecorder.io was pretty useful in replaying WARC files created from all four methods.

Below are the screenshot for replay of warc generated by using Heritrix, WARCreate, webrecorder.io and wget in the same order. Also after every example I am adding description and comparison for warc files. All screenshot can be found in the following path (
<https://github.com/aag1091/cs851-s15/tree/master/assignment2/outputs>)

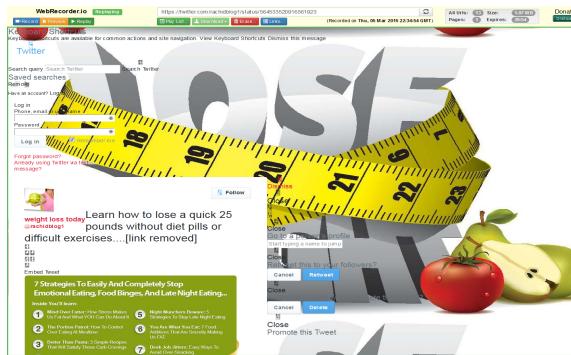
1) <https://twitter.com/rachidblog1/status/564533520916561923>



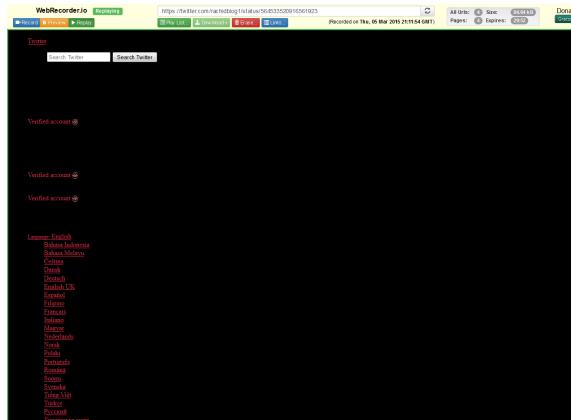
Heritrix



Webrecorder.io



WARCreate



wget

All above screenshots are from webrecorder.io and below is one screenshot using pywb

This is an archived page from Thu, 05 Mar 2015 22:21:19 GMT

weight loss today (@rachidblog1)

Learn how to lose a quick 25 pounds without diet pills or difficult exercises....[link removed]

7 Strategies To Easily And Completely Stop Emotional Eating, Food Binges, And Late Night Eating...

Inside You'll learn:

- 1 Mind Over Fatter: How Stress Makes Us Fat And What You Can Do About It.
- 2 The Portion Patrol: How To Control Over Eating At Mealtimes
- 3 Better Than Pasta: 3 Simple Recipes That Will Satisfy Those Carb Cravings
- 4 The Spouse Factor: 3 Tips To Overcome A Non-Supportive Spouse
- 5 Night Munchers Beware: 5 Strategies To Stop Late Night Eating
- 6 You Are What You Eat: 7 Food Additives That Are Secretly Making Us FAT.
- 7 Desk Job Jitters: Easy Ways To Avoid Over-Snacking

2) <http://willbrooks.deviantart.com/art/Doctor-Who-Titles-GIF-491765997>

The screenshot shows the Heritrix web crawler interface. The left panel displays a search results page for "Doctor Who Titles" on DeviantArt. The results include various fan art pieces, such as a large circular collage featuring the Doctor and his companions, and several smaller images of the TARDIS and Doctor Who characters. The right panel shows the raw HTML code of one of the search results, specifically a DeviantArt page for a fan art piece titled "DOCTOR WHO". The HTML includes standard web page elements like the title, meta tags, and a large black watermark for "DEVANT ART".

Webrecorder.io

wget

All above screenshots are from webrecorder.io and below is one screenshot using pywb

This is an archived page from Thu, 05 Mar 2015 22:20:47 GMT

deviantART
Browse Art Prints Shop Groups Deviants

Find More Art
Daily
Deviations Critiques Critiqueable Wallpaper Random
Random Deviant Random Deviation Random Group

Sta.shdA muro Chat Journals Today Forum

Buy Art Buy adCast
Buy Premium
Membership
Shop

Groups Forum Chat Journals
Wallpaper Portfolio
Critiques Critiqueable Random
Deviant Random Deviation
Random Group Help & FAQ
More

Mobile

Search

Submit Join

Username or Email
Password Stay logged in
Login

Join DeviantArt for FREE Take the Tour Forgot Password or Username? Login

3) <http://www.f1technical.net/features/19903>

Technical analysis: Mercedes F1 W06

Do you think that Abu Dhabi's testing rules were reported correctly and fully?

Poll

Todays events
2017 - Test days Event calendar

Want to be a friend
Improve this page
Knowledge base
About us
Feedback
Sponsors
Amazon.co.uk

Technical analysis: Mercedes F1 W06

Do you think that Abu Dhabi's testing rules were reported correctly and fully?

Poll

Todays events
2017 - Test days Event calendar

Want to be a friend
Improve this page
Knowledge base
About us
Feedback
Sponsors
Amazon.co.uk

Heritrix

Technical analysis: Mercedes F1 W06

Do you think that Abu Dhabi's testing rules were reported correctly and fully?

Poll

Todays events
2017 - Test days Event calendar

Want to be a friend
Improve this page
Knowledge base
About us
Feedback
Sponsors
Amazon.co.uk

Technical analysis: Formula One uncovered

Log in or Register
Remember me Log in Forget your password

- News
 - Headlines
 - Grand Prix calendar
 - Event calendar
- Technical
 - Development
 - Drivers
 - Glossary
- Database
 - All races ever
 - Rules & Regulations
 - Statistics
- Photo
 - Categories
 - View new photo
 - View active photo
- More
 - Chat
 - Polls
 - Links

Home - Archive - Technical analysis: Mercedes F1 W06

All times are UTC + 1 hour (DST)

Technical analysis: Mercedes F1 W06

Do you think that Abu Dhabi's testing rules were reported correctly and fully?

Poll

Todays events
2017 - Test days Event calendar

Want to be a friend
Improve this page
Knowledge base
About us
Feedback
Sponsors
Amazon.co.uk

Webrecorder.io

wget

All above screenshots are from webrecorder.io and below is one screenshot using pywb

F1 Technical: Formula One uncovered

Login or Register

Username: *
Password: *

Remember me [Forgot your password?](#)

- News
 - [Headlines](#)
 - [Grand Prix calendar](#)
 - [Event calendar](#)
- Technology
 - [Development blog](#)
 - [Features](#)
 - [Glossary](#)
- Database
 - [All cars ever](#)
 - [All teams ever](#)
 - [Rules & Regulations](#)
 - [Statistics](#)
- Photos
- Forum
 - [Categories](#)
 - [View new posts](#)
 - [View active topics](#)
- More
 - [Chat](#)
 - [Polls](#)
 - [Links](#)

Home » Articles » Technical analysis: Mercedes F1 W06

All times are UTC + 1 hour [DST]

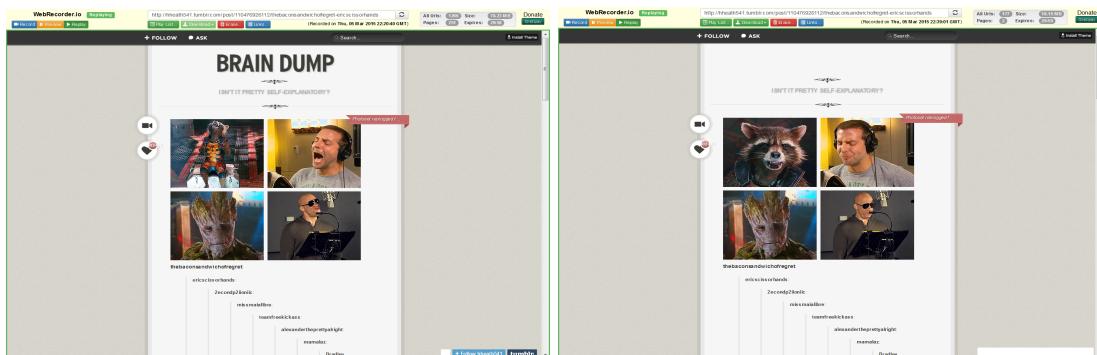
Technical analysis: Mercedes F1 W06

By Steven De Groot on 01 Feb 2015, 17:15

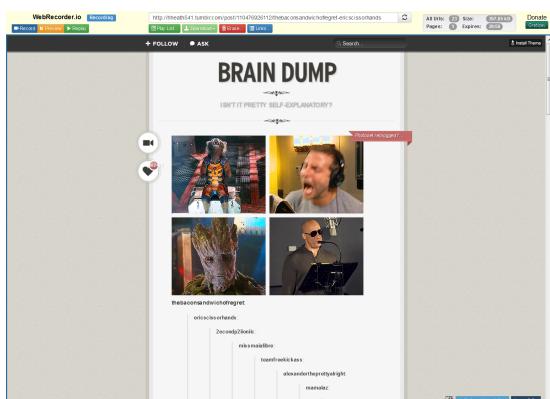
Having utterly dominated the 2014 season, Mercedes had a big task over the winter to design a new car that could replicate that success. Still, rivals have reiterated over the

4)

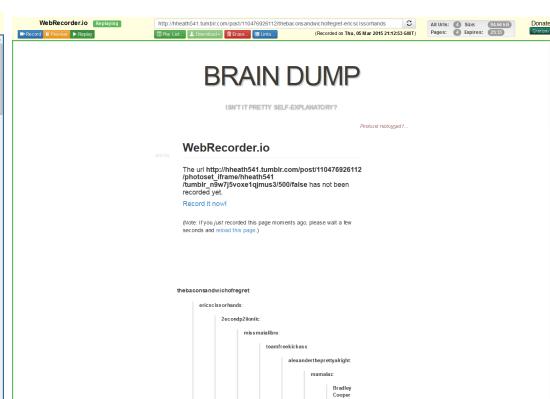
<http://hheath541.tumblr.com/post/110476926112/thebaconsandwichofregret-ericscissorhands>



Heritrix



WARCreate

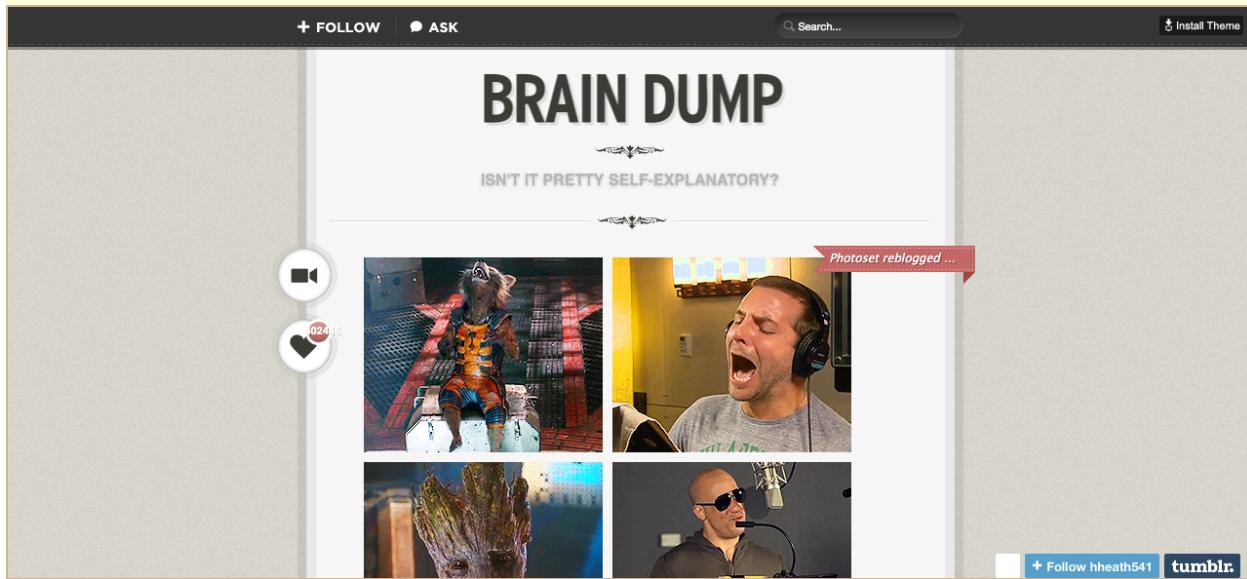


Webrecorder.io

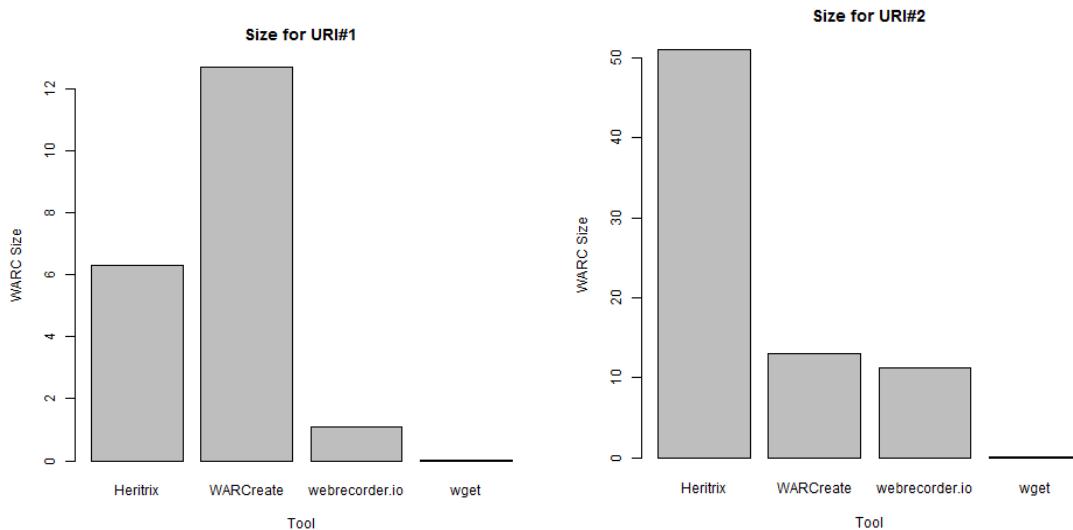
wget

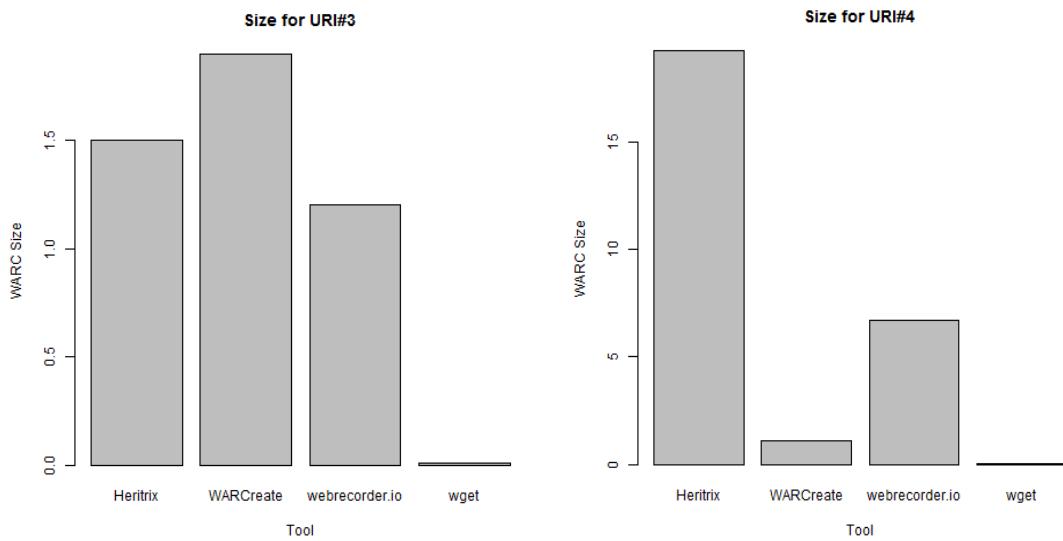
All above screenshots are from webrecorder.io and below is one screenshot using pywb

This is an **archived** page from **Thu, 05 Mar 2015 22:20:40 GMT**



Here are the graphs comparing sizes for warc files created by different tools.





The sizes of warc files do depend on the method that we use to generate the warc file. Wget has the least size for all four URI. Heritrix would always have bigger size because it always crawls into sub-pages for a URI. The size of WARC also depends on url we are trying to call. If it has more media and deep pages it will be of bigger size.

Q2)

Solr has configurations to specify things like what we can index from warc files and what shouldn't be indexed. We can specify if images are to be extracted or not.

For running queries I kept the default configuration given in webarchive-discovery.

The below screenshot shows a query to return urls for domain twitter. This would give links from twitter.

The screenshot shows the Apache Solr admin interface. On the left, there's a sidebar with various navigation links like Dashboard, Logging, Core Admin, Java Properties, Thread Dump, discovery, Query, Replication, and Schema Browser. The 'Query' link is currently selected. In the main panel, there's a 'Request-Handler (qt)' section with a dropdown set to '/select'. Below it are several input fields: 'common' (with 'domain:twitter.com' entered), 'fq' (empty), 'sort' (empty), 'start', 'rows' (set to 0 and 10), 'fl' (empty), and 'df' (empty). Under 'Raw Query Parameters', there are two key-value pairs: 'key1=val1&key2=val2'. The 'wt' field is set to 'json'. There are also two checkboxes: 'indent' (checked) and 'debugQuery' (unchecked). To the right of the interface is a browser window displaying the JSON response of the search request. The URL is <http://localhost:8080/discovery/select?q=domain%3Atwitter.com&wt=json&indent=true>. The response shows a single document with the following details:

```

{
  "responseHeader": {
    "status": 0,
    "QTime": 7,
    "params": {
      "q": "domain:twitter.com",
      "indent": "true",
      "wt": "json",
      "_": "1425615792663"
    }
  },
  "response": {
    "numFound": 2,
    "start": 0,
    "docs": [
      {
        "source_file_s": "WARCmerge20150305003147090023.warc@2229123",
        "url": "https://twitter.com/rachidblog1/status/564533520916561923?utm_source=twitterfeed&utm_medium=twit",
        "host": "twitter.com",
        "domain": "twitter.com",
        "public_suffix": ".com",
        "content_length": 82943,
        "id": "sha1:UYOFJEOEO2W5ROB2M2VP53EXSKETCIN/3NUSPhFjh/Q08xlesNzCSw==",
        "hash": [
          "sha1:UYOFJEOEO2W5ROB2M2VP53EXSKETCIN"
        ],
        "crawl_date": "2015-03-05T00:30:31Z",
        "crawl_year": "2015",
        "workbook_data": "20150305003147090023"
      }
    ]
  }
}

```

The below screenshot shows a query to return urls that org as public suffix. This would give me all org domain urls.

This screenshot is similar to the one above, showing the Apache Solr admin interface with the 'Query' link selected. The 'Request-Handler (qt)' dropdown is set to '/select'. The 'common' field contains 'public_suffix:org'. All other fields ('fq', 'sort', 'start', 'rows', 'fl', 'df', 'Raw Query Parameters', 'wt', 'indent', and 'debugQuery') are empty or at their default values. The browser window to the right shows the JSON response for this query. The URL is http://localhost:8080/discovery/select?q=public_suffix%3Aorg&wt=json&indent=true. The response shows multiple documents, with the first one being:

```

{
  "responseHeader": {
    "status": 0,
    "QTime": 9,
    "params": {
      "q": "public_suffix:org",
      "indent": "true",
      "wt": "json",
      "_": "1425615853748"
    }
  },
  "response": {
    "numFound": 40,
    "start": 0,
    "docs": [
      {
        "source_file_s": "flashfrozen-jwat-recompressed.warc.gz#849",
        "url": "http://bits.wikimedia.org/en.wikipedia.org/load.php?debug=false&lang=en&modules=site&only=script",
        "content_type_ext": "php",
        "host": "bits.wikimedia.org",
        "domain": "wikimedia.org",
        "public_suffix": "org",
        "server": [
          "Apache",
          "PHP/5.3.10-lubuntu3.4+wmf1"
        ],
        "content_type_served": "text/javascript; charset=utf-8",
        "content_length": 8963,
        "id": "sha1:1CABD7D6A77E9E080C9D0148380777B9D7B0000"
      }
    ]
  }
}

```

For processing documents in solr I executed following command.

...

```
java -jar warc-indexer-2.0.1-20150116.110435-2-jar-with-dependencies.jar -s
http://localhost:8080/discovery -t
/Users/avinashgosavi/projects/cs851-s15/assignment2/wget-combined-warc
```

...