Khaja Masroor Ahmed

Assignment 2

CS 751: Introduction to Digital Libraries
Dr. Michael Nelson
Spring 2015

March 6, 2015

Contents

1

Question 1

1.1 Question

- Choose 100 URIs from A1.
- Generate WARC files of those URIs using:
  - wget
  - WARCreate
  - Heritrix (stand-alone or via WAIL)
  - webrecorder.io
- Describe the resulting WARC files: quantitatively compare contrast the results of the WARC files of the same URI as generated by different tools.
  - choose interesting examples.
- Demonstrate playback of 2-3 WARCs in the (Wayback Machine (via WAIL or stand-alone) or pywb) and (webrecorder.io)
  - https://github.com/iipc/openwayback
  - https://github.com/ikreymer/pywb

1.2 Solution

- Fetching the 100 URIs was an arduous task, as a lot of the URIs had explicit content or were not accessible anymore. I had to open most of the links manually and as I progressed through the links I noticed a pattern and started avoiding those links to avoid additional time fetching the URIs.
- I saved the URIs into a uri.txt file.
- Soon after I installed the plug-in for WARCreate from Chrome Web Store and started manually visiting these URIs and clicking on the Warccreate button in the address bar for chrome web browser for downloading the WARC files.
- After WARCreate, I moved onto the next tool webrecorder.io. I manually opened each of the URIs on the website and started downloading the

WARC files. While fetching them manually I realized this activity could be automated but I decided that performing this task manually would re-assure me that the WARC files have been created properly.

- The third tool WAIL consumed most of my time as the set-up wasn't as simple as it had multiple dependencies, required configuration changes and lacked proper documentation. As the thought of setting up stand-alone heritrix creeped into my mind, WAIL set-up issues were addressed by an email from Alexander Nwala and after following his instructions I was able to get the WAIL tool working.
- I used the WAIL tool to create the archive but it took too long for most of the links so I put a time limit of approximately 10 minutes for each of the URIs and then manually stopped execution and moved to the next URI.
- "wget" was the simplest of all the tools for WARC creation. I used a simple unix command with the template "wget URI –warc-file=FILENAME –no-warc-compression" for each of the 100 URIs.

1.3 Comparison

- From the 10000 URIs retrieved from the first assignment we took a sample of 100 URIs. To make a meaningful comparison of the sample I calculated the mean of the size of WARC files from each of the tools.

- Below is the graph to represent the average size of the WARC files for each of the tools.

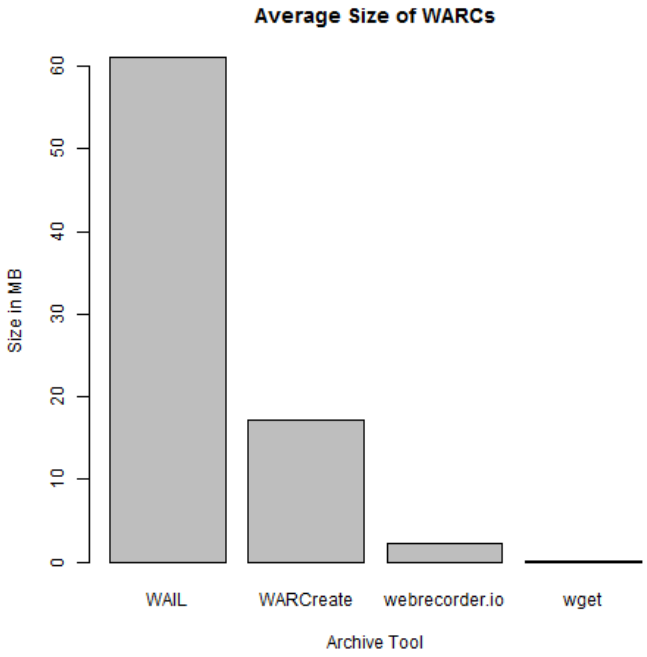


Fig. 1.1. Average Size of WARC file for sample size=100

- While downloading the WARC files for each of the above tools I noticed a trend that WAIL was able to crawl the deepest and it had to be manually stopped. The size of the files generated by WAIL were significantly larger than the ones created by the other tools.
- With this information and the above graph I concluded that WAIL creates the largest of the files. Upon further analysis I found that WAIL crawls dereferences each of the URIs in the webpage. WAIL does not stop until it crawls to the end of the URI.
- As it is a configurable tool, we can modify the specify the time spent to crawl each URI, maximum size of the WARC file and/or maximum documents downloaded in the crawler-beans.cxml created for each of the jobs.
- Below are the WARC size plots for three randomly chosen URIs as specified below, which I've used for the rest of the assignment for playback demonstration.
  - URI#1: http://edition.cnn.com/videos/us/2015/02/04/lead-dnt-johns-plane-crash-taiwan.cnn
  - URI#2: http://www.dccomics.com/blog/2015/02/04/breaking-news-milo-ventimiglia-is-coming-to-gotham

- URI#3: http://www.theguardian.com/football/2015/feb/04/jonny-evans-manchester-united-louis-van-gaal

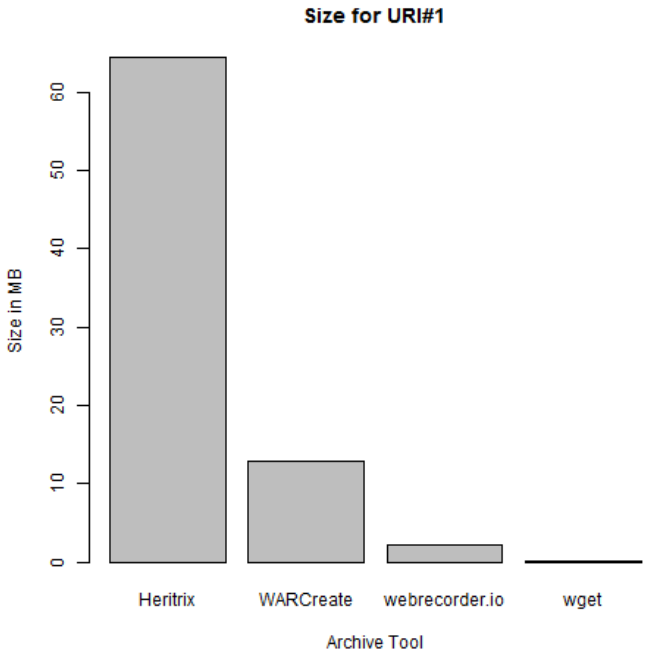


Fig. 1.2. Archive Tools vs. Size for URI#1

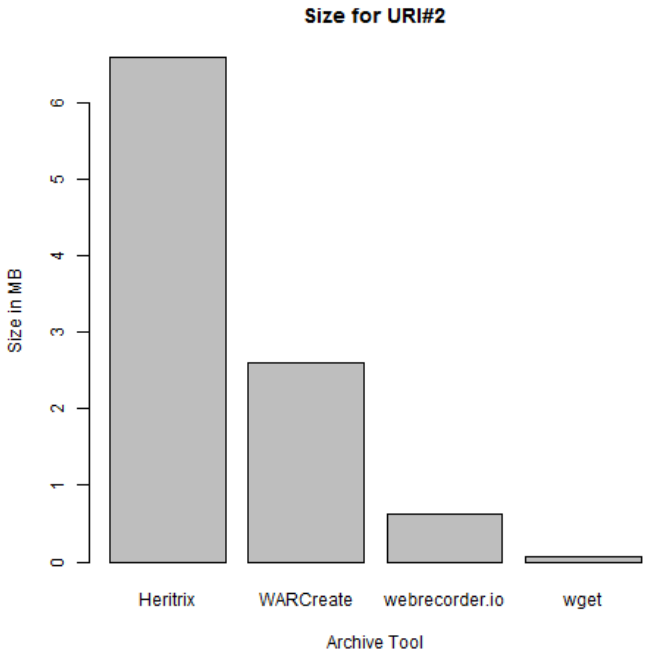


Fig. 1.3. Archive Tools vs. Size for URI#2

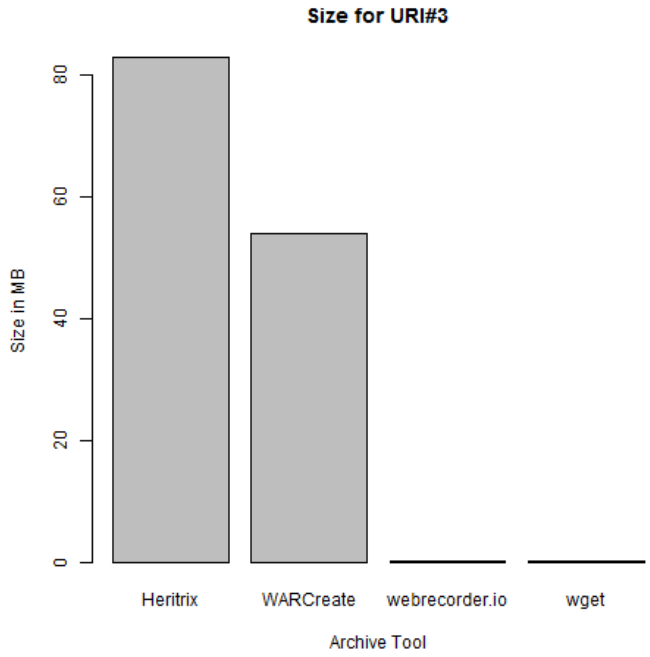


Fig. 1.4. Archive Tools vs. Size for URI#3

1.4 Playback

1.4.1 Webrecorder.io

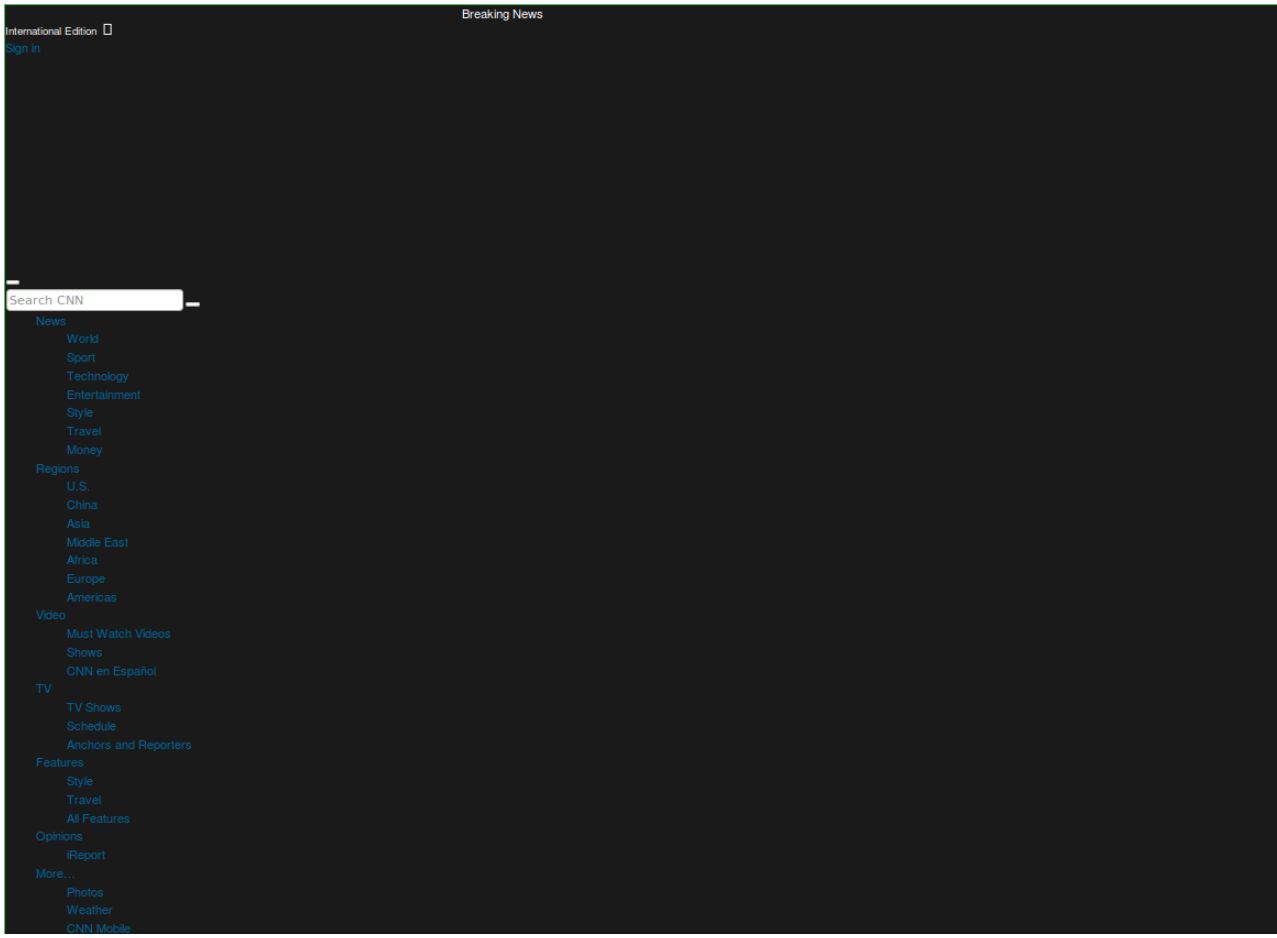


Fig. 1.5. Web Recorder - WAIL generated WARC for URI#1

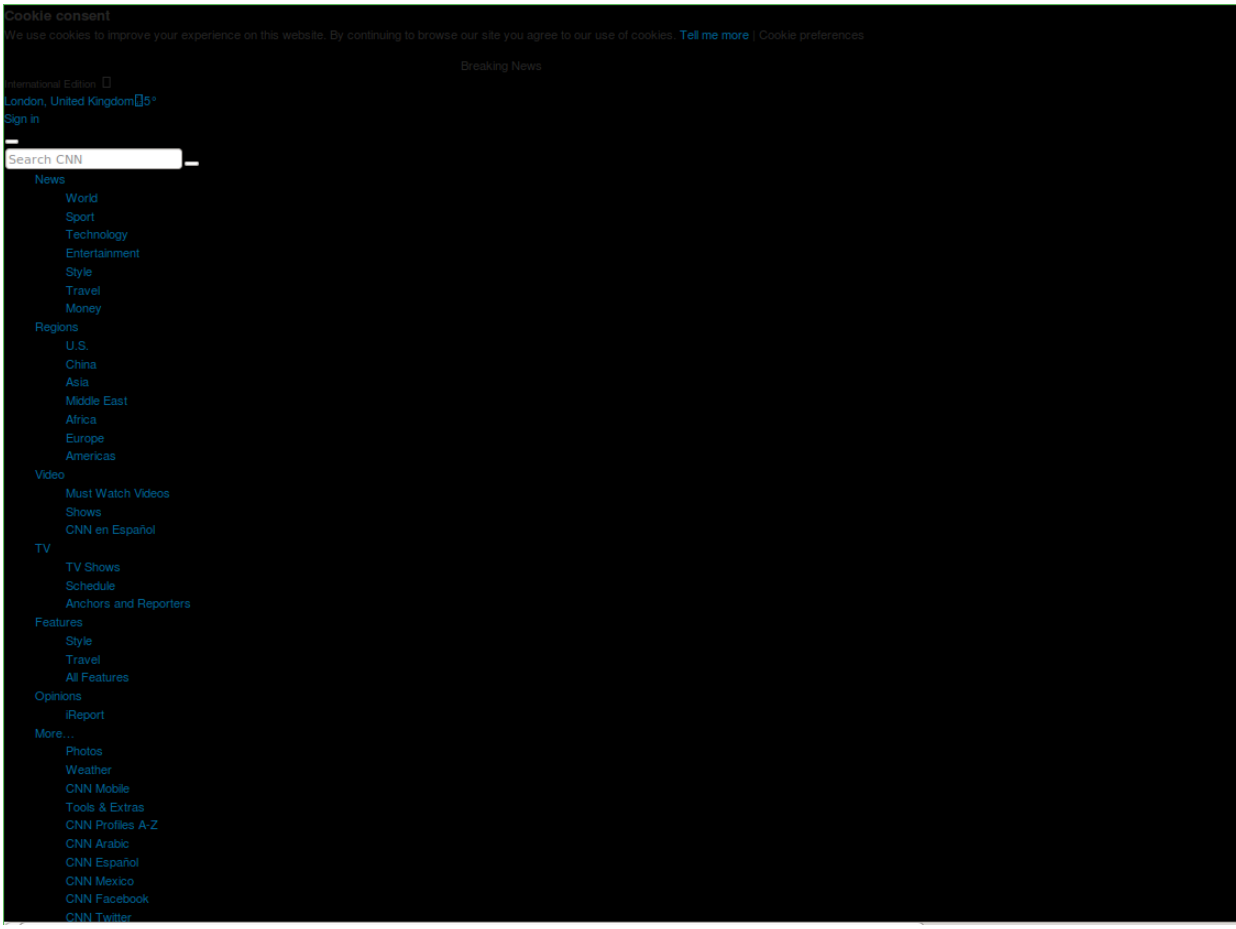


Fig. 1.6. Web Recorder - WARCreate generated WARC for URI#1

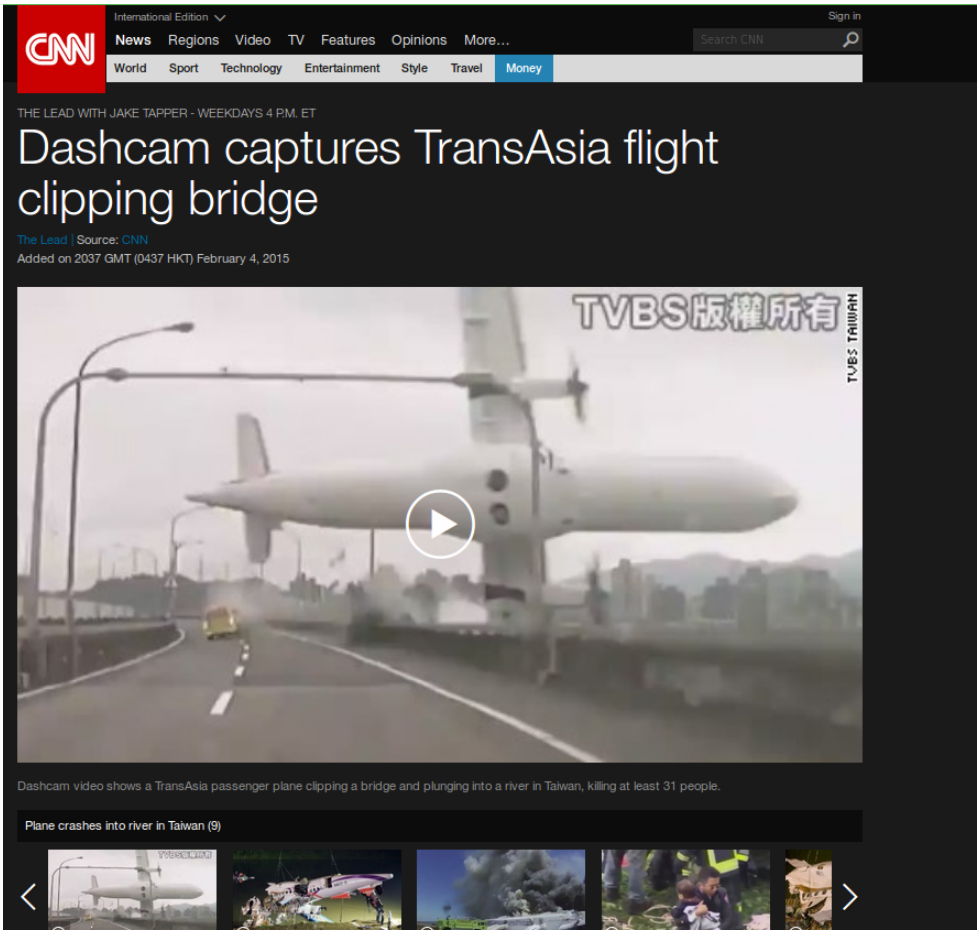
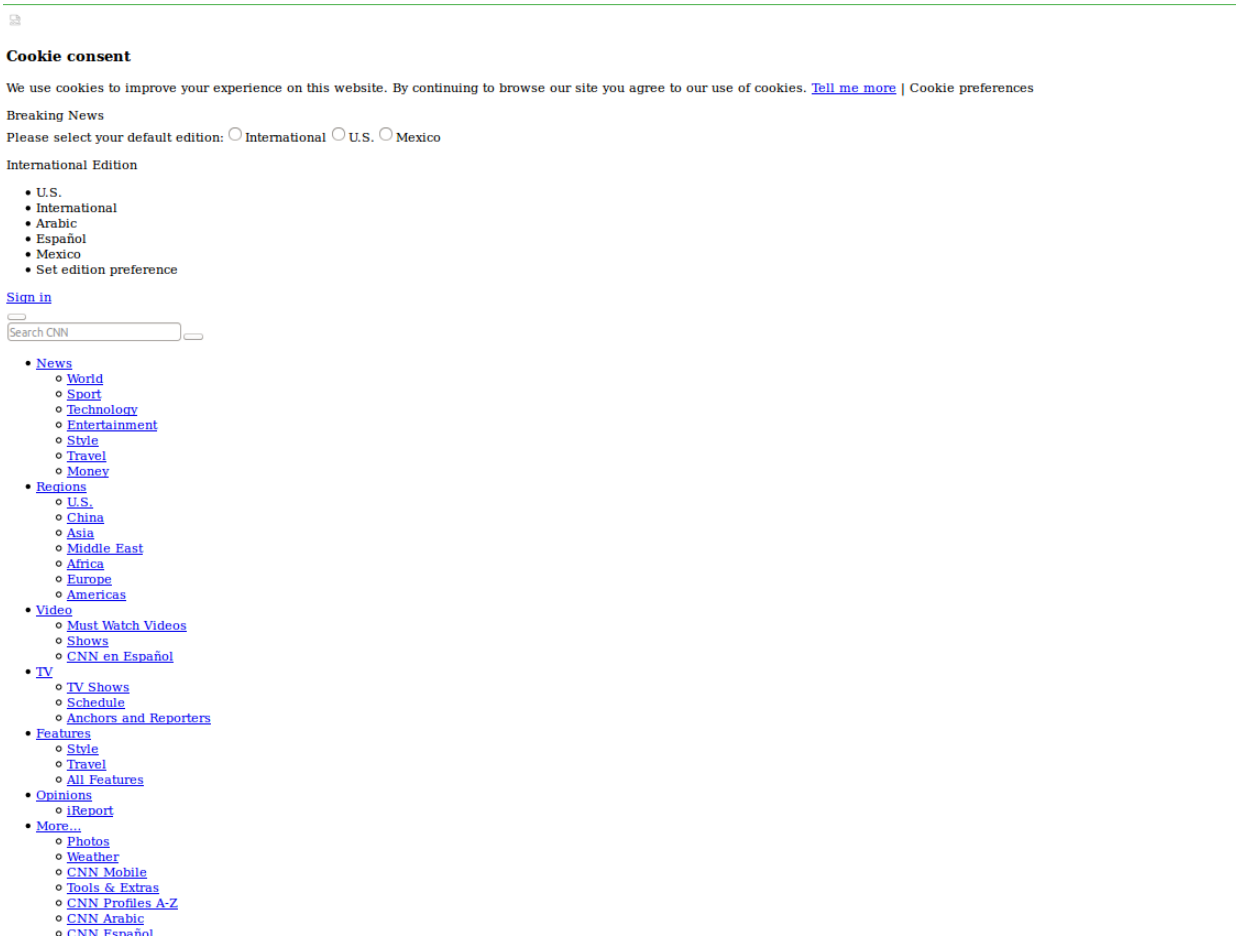


Fig. 1.7. Web Recorder - webrecorder.io generated WARC for URI#1



Cookie consent

We use cookies to improve your experience on this website. By continuing to browse our site you agree to our use of cookies. Tell me more | Cookie preferences

Breaking News

Please select your default edition: International U.S. Mexico

International Edition

- U.S.
- International
- Arabic
- Español
- Mexico
- Set edition preference

Sign in

Search CNN

- News
  - World
  - Sport
  - Technology
  - Entertainment
  - Style
  - Travel
  - Money
- Regions
  - U.S.
  - China
  - Asia
  - Middle East
  - Africa
  - Europe
  - Americas
- Video
  - Must Watch Videos
  - Shows
  - CNN en Español
- TV
  - TV Shows
  - Schedule
  - Anchors and Reporters
- Features
  - Style
  - Travel
  - All Features
- Opinions
  - iReport
- More...
  - Photos
  - Weather
  - CNN Mobile
  - Tools & Extras
  - CNN Profiles A-Z
  - CNN Arabic
  - CNN Español

Fig. 1.8. Web Recorder - wget generated WARC for URI#1

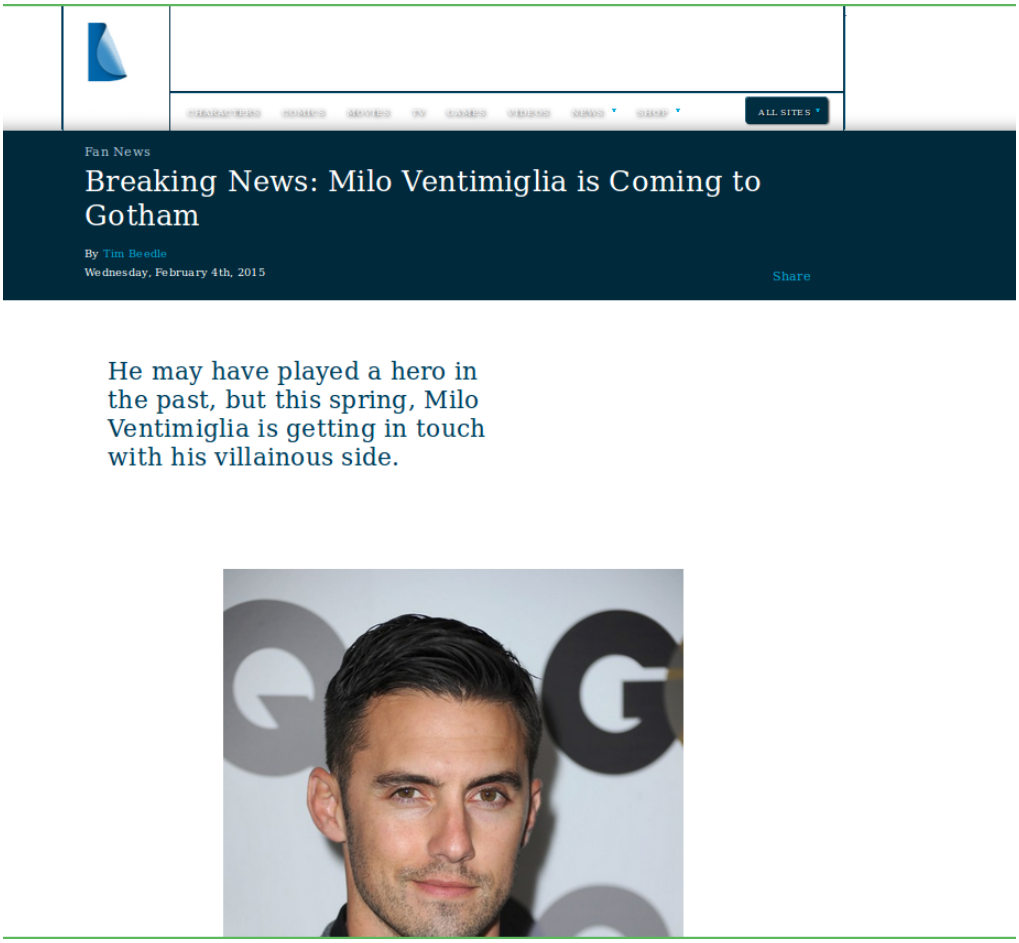


Fig. 1.9. Web Recorder - WAIL generated WARC for URI#1

Fig. 1.10. Web Recorder - WARCreate generated WARC for URI#1

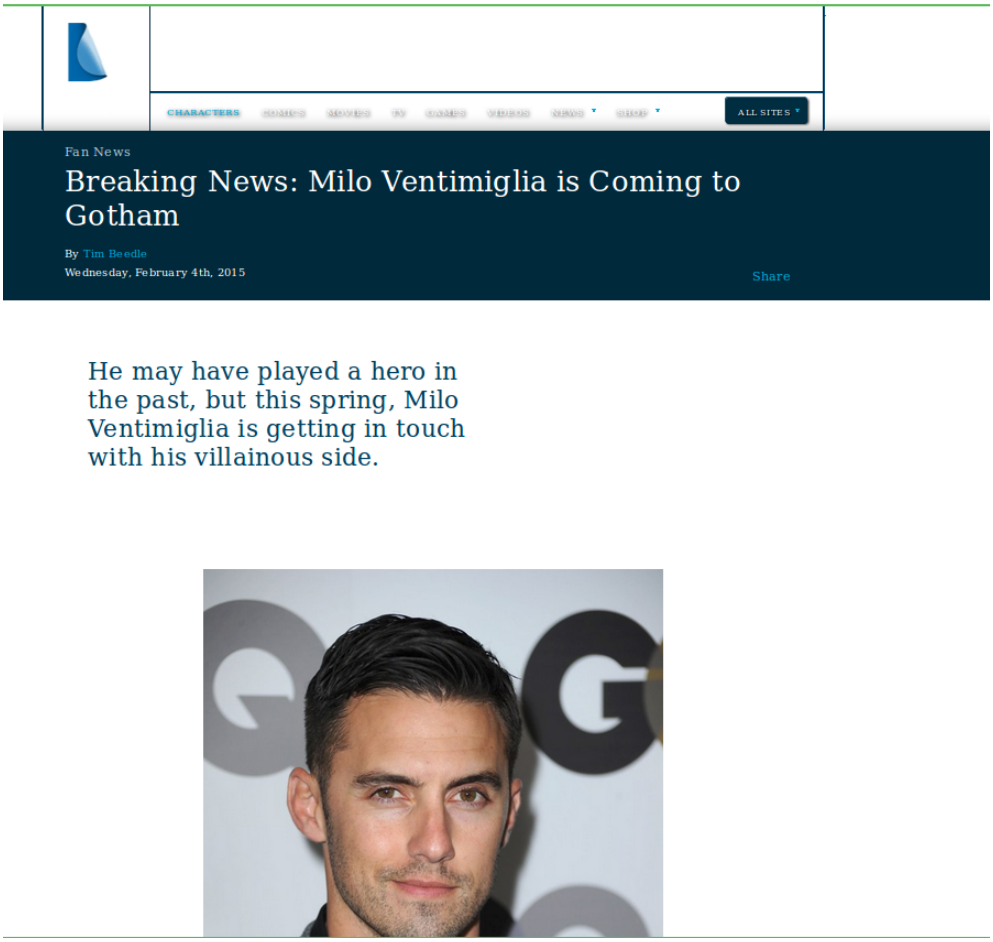


Fig. 1.11. Web Recorder - webrecorder.io generated WARC for URI#1

- DC ENTERTAINMENT
- DC Comics
- Vertigo
- MAD

DC Entertainment

- About DC Entertainment
- Jobs
- Contact Us
- Subscriptions
- Advertising
- Privacy Policy
- Terms of Use
- Ad Choices
- Submissions
- CPSC Certificates
- Ratings
- Site Map
- Shop Help

Search form

Search
Search

DC Comics SOCIAL

Fan News

Breaking News: Milo Ventimiglia is Coming to Gotham

Breaking News: Milo Ventimiglia is Coming to Gotham

By Tim Beedle Wednesday, February 4th, 2015
Share

He may have played a hero in the past, but this spring, Milo Ventimiglia is getting in touch with his villainous side.

He may have played a hero in the past, but this spring, Milo Ventimiglia is getting in touch with his villainous side.

Gotham's already seen its fair share of gangsters, hit men and full on psychopaths, but Detective Jim Gordon will find himself going toe-to-toe with a new kind of serial killer later this season p
multiple-episode recurring arc.

Fig. 1.12. Web Recorder - wget generated WARC for URI#1

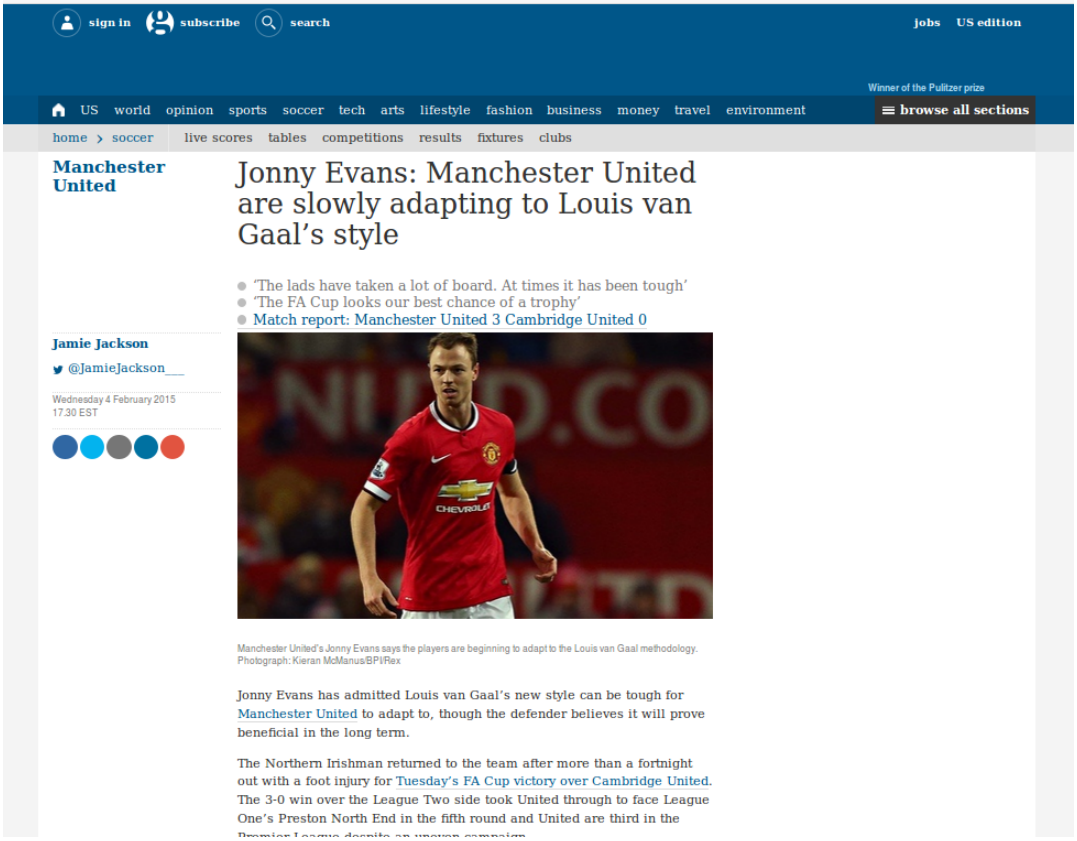


Fig. 1.13. Web Recorder - WAIL generated WARC for URI#1

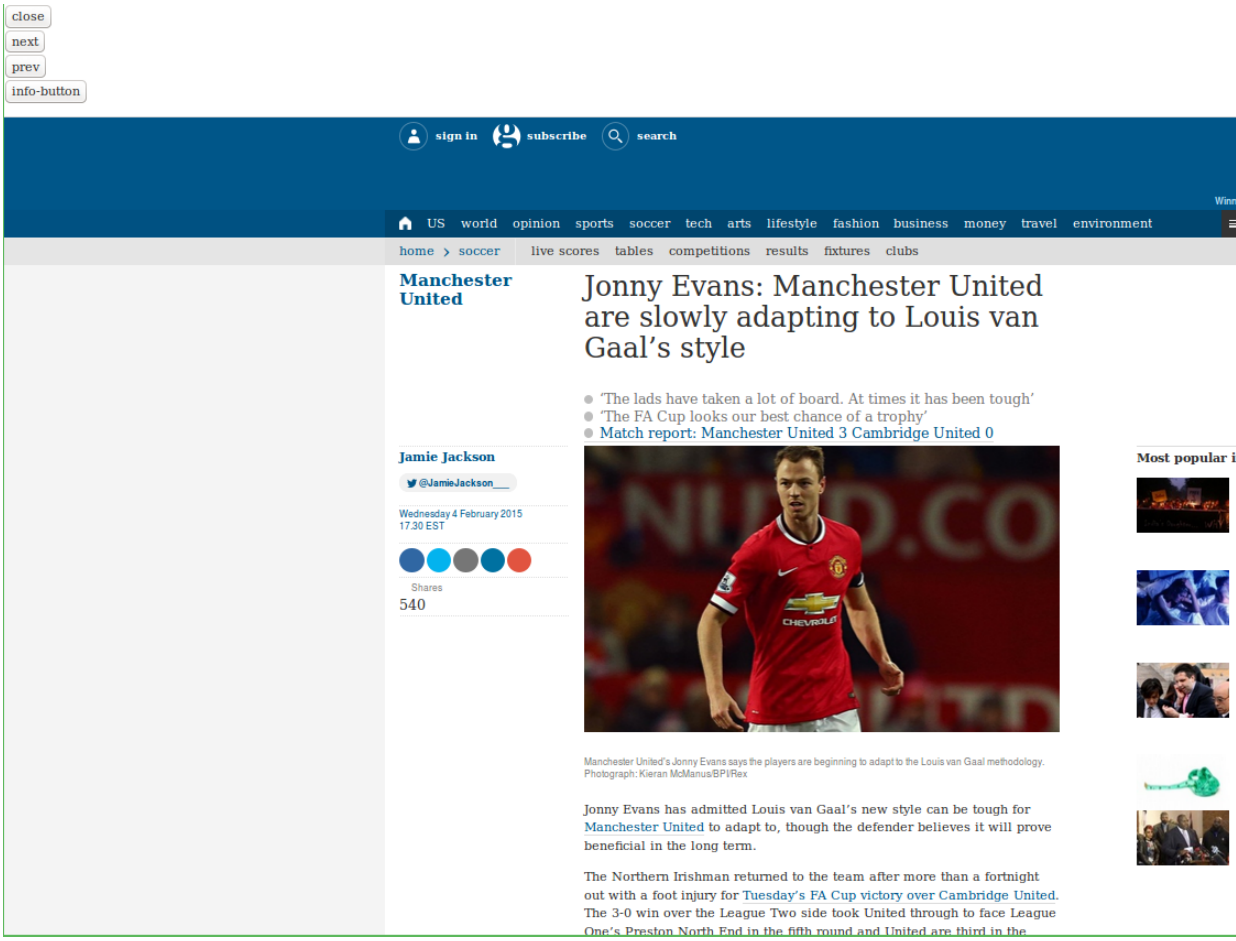


Fig. 1.14. Web Recorder - WARCreate generated WARC for URI#1

Fig. 1.15. Web Recorder - webrecorder.io generated WARC for URI#1

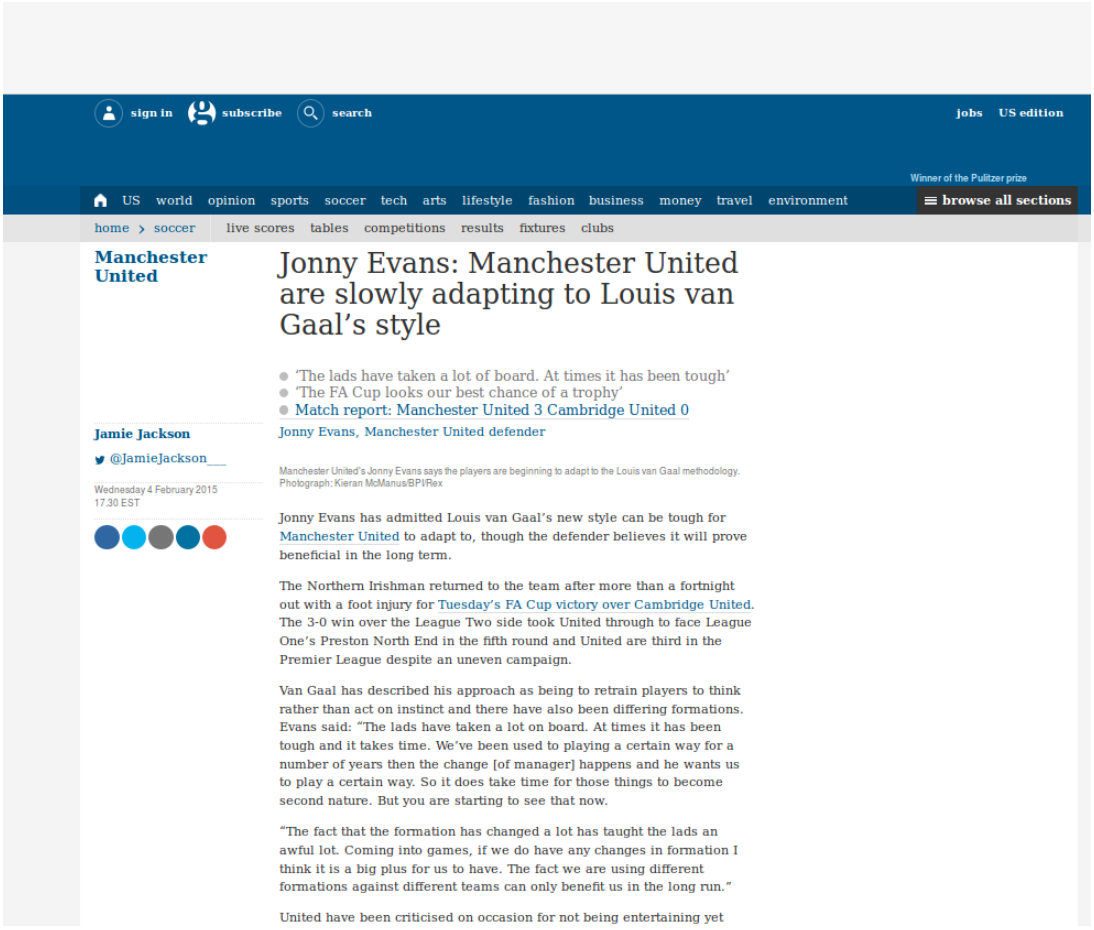


Fig. 1.16. Web Recorder - wget generated WARC for URI#1

1.4.2 pywb

- For setting pywb I followed the instructions at https://github.com/ikreymer/pywb.
- I created the cdx files for the WARC files and then launched pywb using the command wayback after going to the root folder of pywb in the terminal.
- I used the web browser and went to "localhost:8080" for accessing the web interface for pywb. Below is the search interface for pywb.

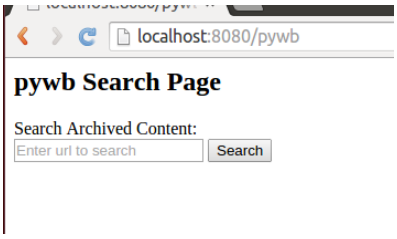


Fig. 1.17. Web Search Interface pywb

- After creating the cdx files for the associated URIs, enter the URI in the search field.
- I was facing an issue with the WAIL results not being displayed for pywb inspite of the cdx files being generated properly and no errors in the command prompt. I tried using different WAIL generated WARC files but I was still not able to view it in the list of search results and hence wasn't able to display the results.

Fig. 1.18. Search results page on pywb

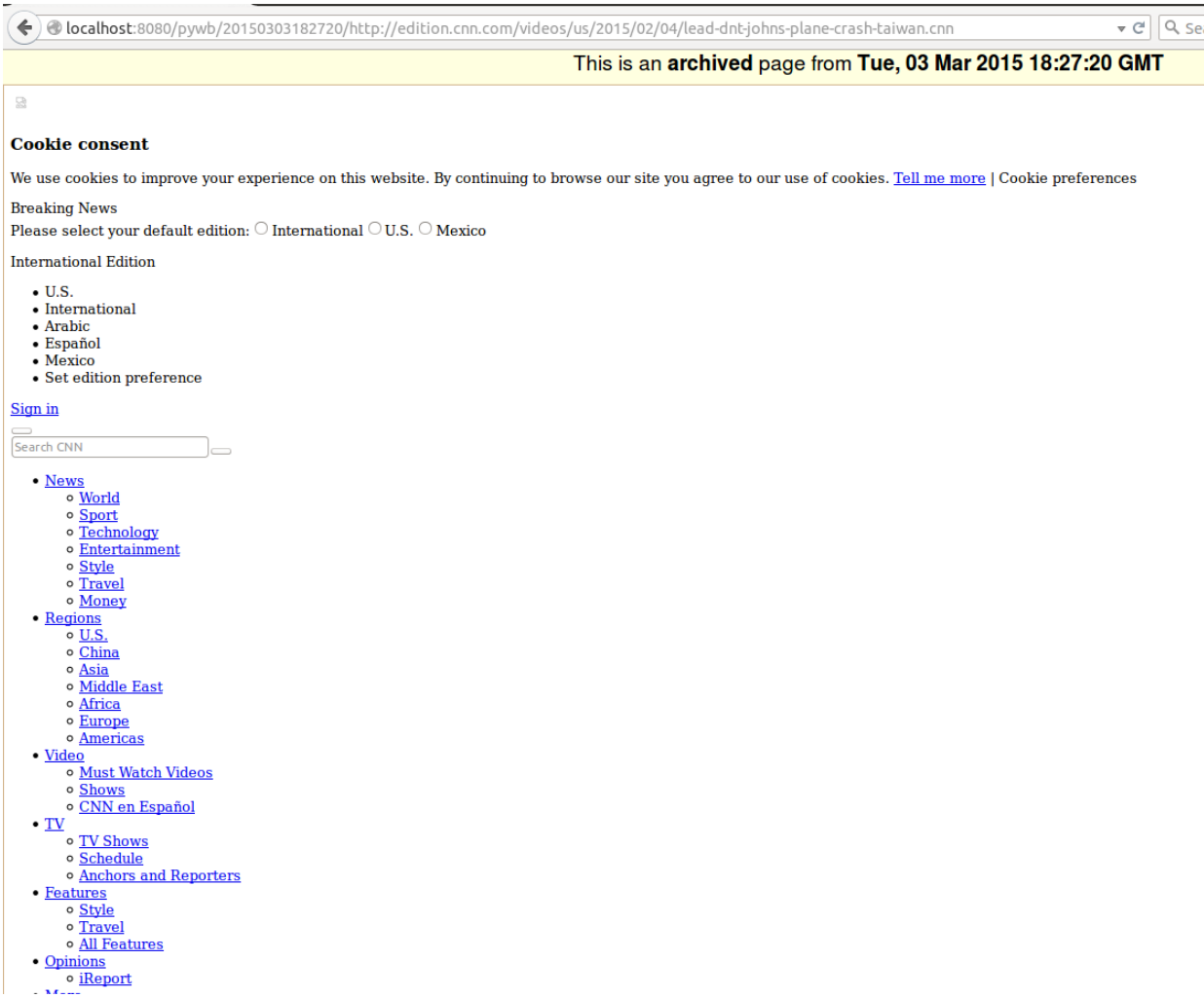


Fig. 1.19. pywb - wget generated WARC for URI#1

Fig. 1.20. pywb - WARCreate generated WARC URI#2

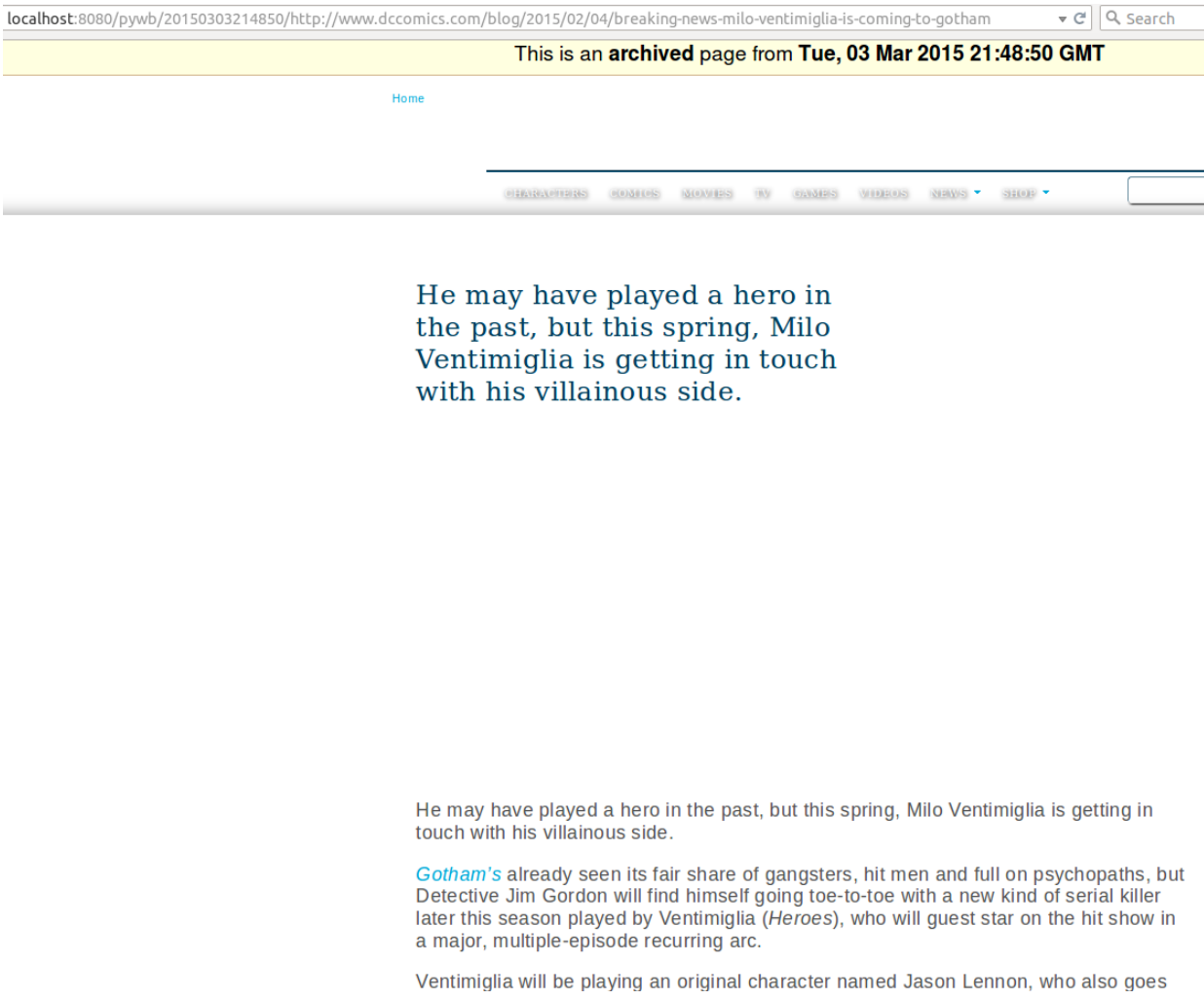


Fig. 1.21. pywb - webrecorder.io generated WARC URI#2

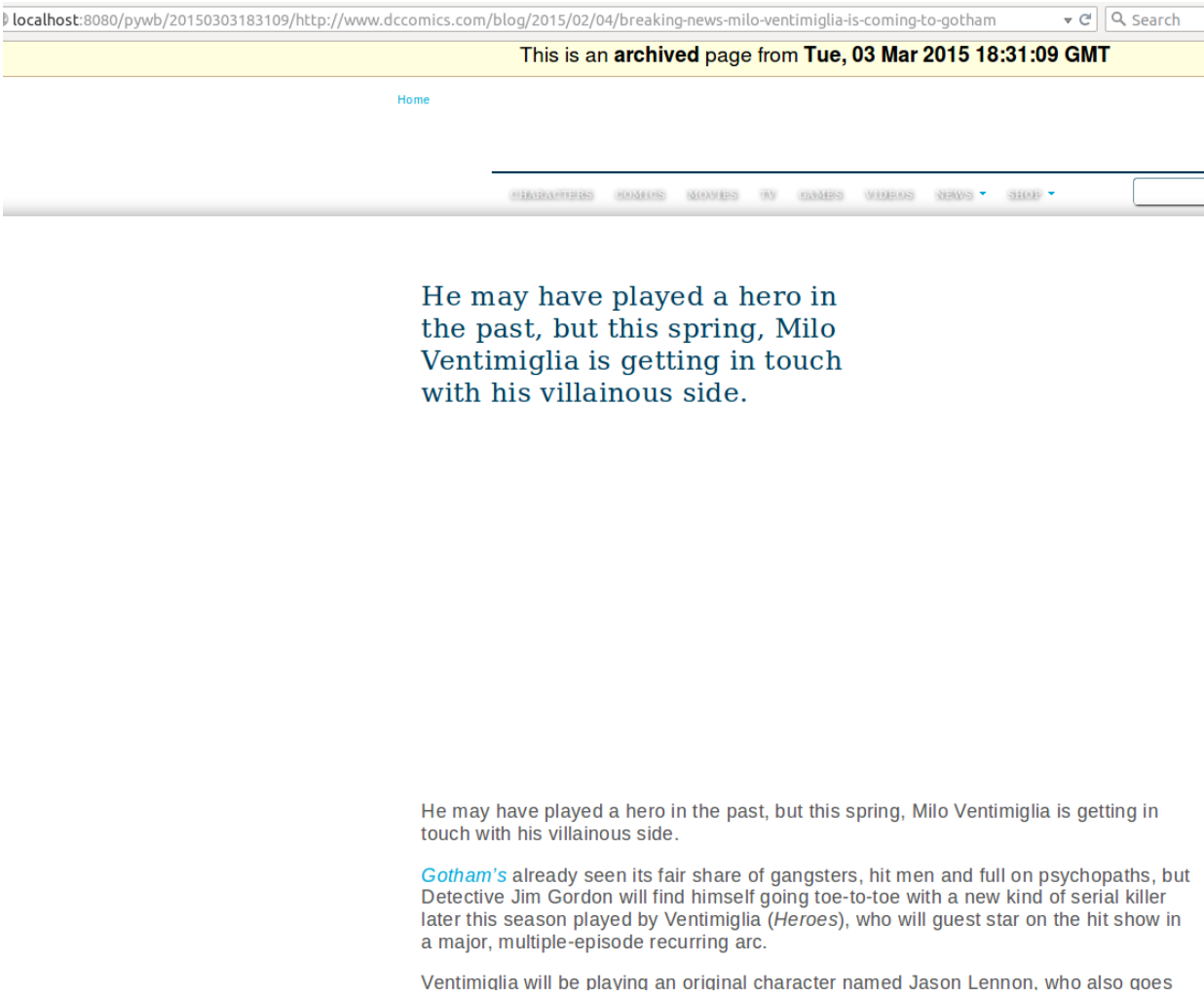


Fig. 1.22. pywb - wget generated WARC URI#2

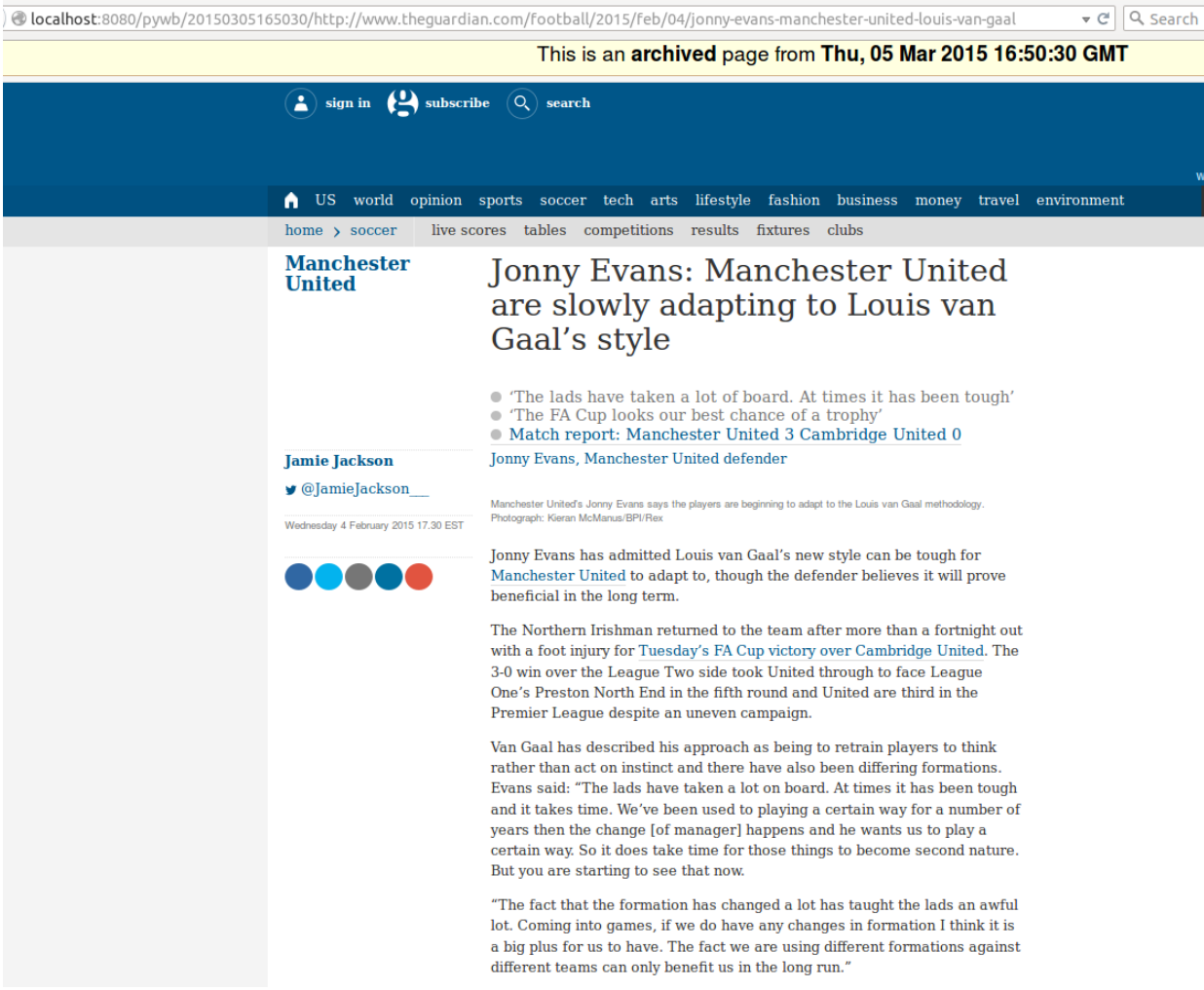


Fig. 1.23. pywb - webrecorder.io generated WARC URI#3

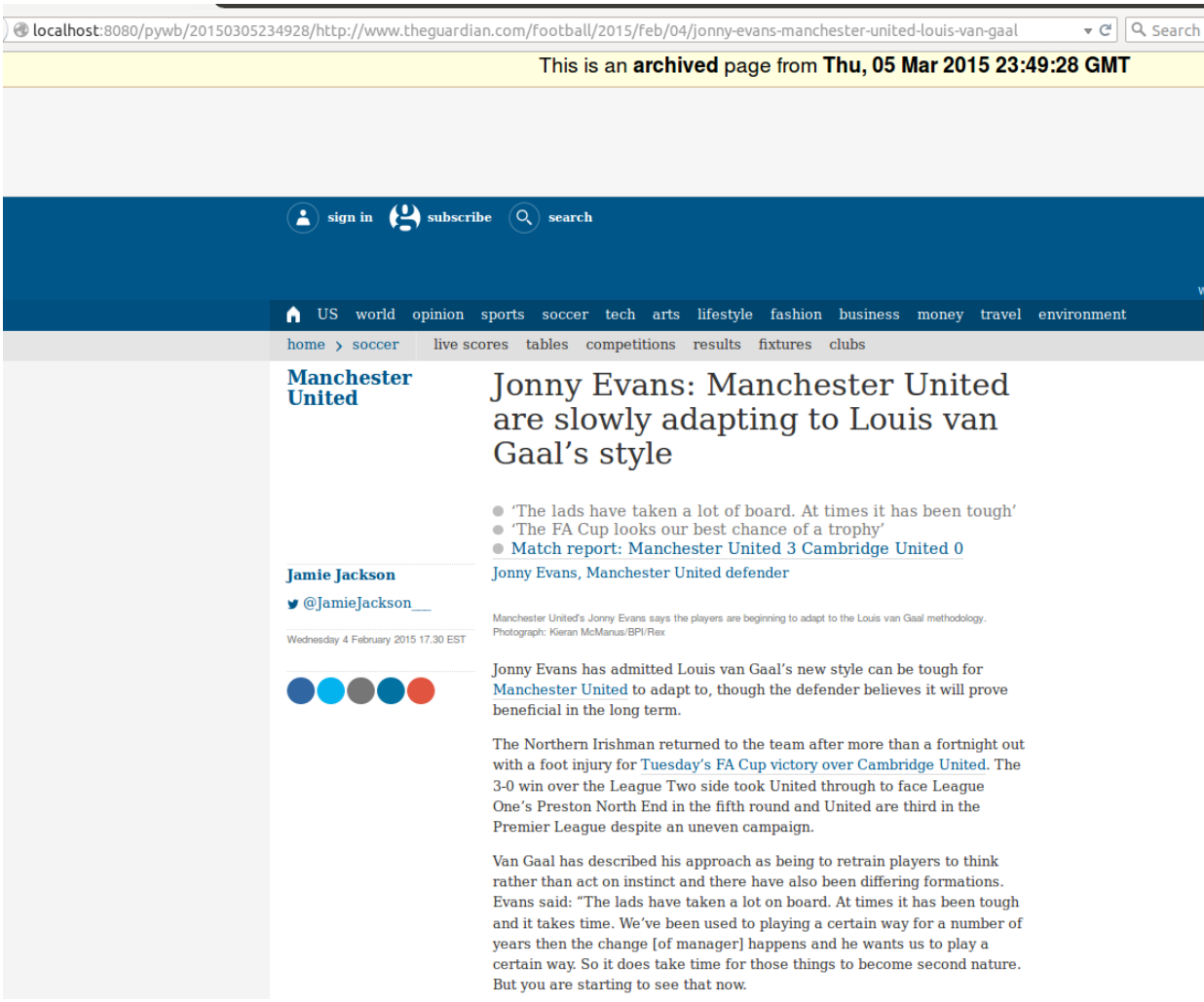


Fig. 1.24. pywb - wget generated WARC URI#3

1.5 Code Listing

```
1 B <- c(83.1,54,0.1537,0.240)
2 png("C:/Users/kahmed/Desktop/uri3.png")
3 barplot(B, main="Size for URI#3", xlab="Archive Tool", ylab=
      "Size in MB", names.arg = c("Heritrix", "WARCreate", "
      webrecorder.io", "wget"))
```

Listing 1.1. R program for generating the bar plot for WARC Tools vs. Average Size of WARC files for 100 samples

2

Question 2

2.1 Question

- Ingest the 100 URIs from their resulting WARC files into a SOLR instance.
  - see the code + tutorial at: https://github.com/ukwa/webarchive-discovery
- Demonstrate several functioning queries on the files (a full front-end is not required)
  - describe the configuration choices you made in setting up SOLR and processing the documents

2.2 Solution

- For setting up SOLR I used the link as provided in the question.
- The instructions provided were clear and concise which helped me set-up the necessary tools without any issues.
- Though I faced an issue when the pywb was running and I wouldn't be able to run SOLR. Then I realized that both the applications were trying to run on the same port as it was already in use. Upon closing the pywb instance and then running SOLR everything was working like a well oiled machine.
- I merged all the WARC files that I had retrieved using wget, using the tool WARCMerge located at https://github.com/maturban/WARCMerge.
- After merging the WARC file I ran the command for indexing as provided in the user instructions by the author of the tool.
- Following this I ran the default web interface for SOLR and ran the following queries as displayed below.

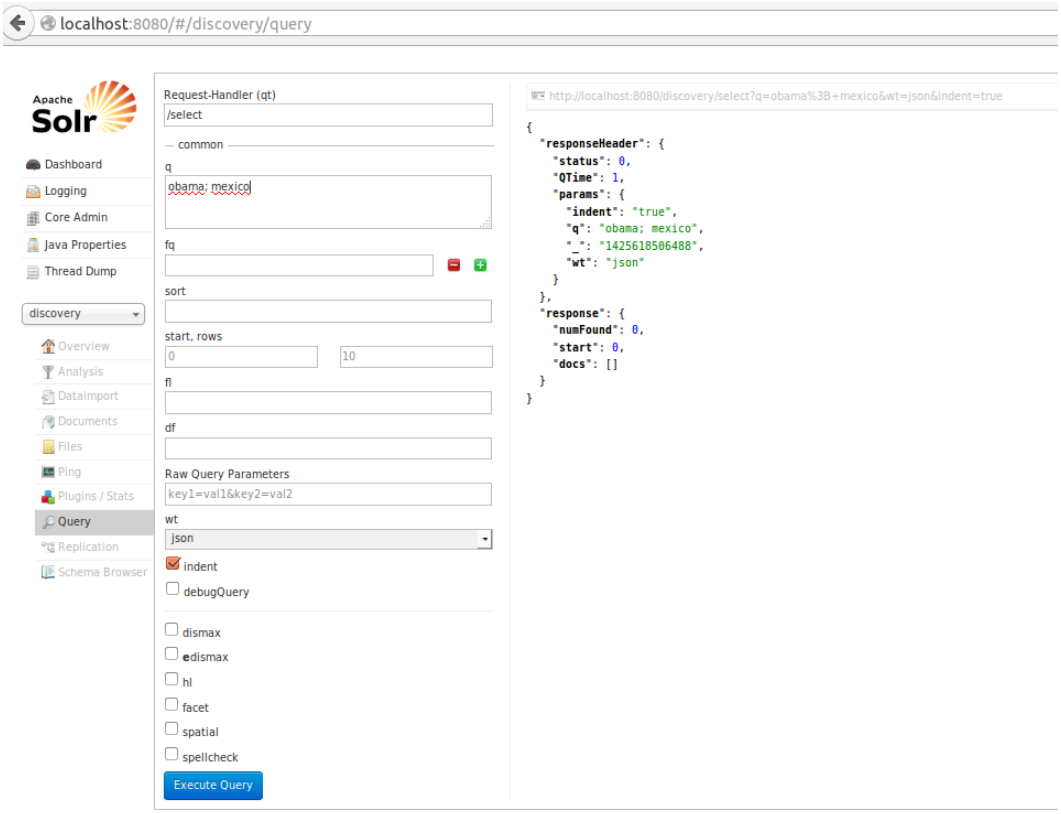


Fig. 2.1. Query: obama; mexico

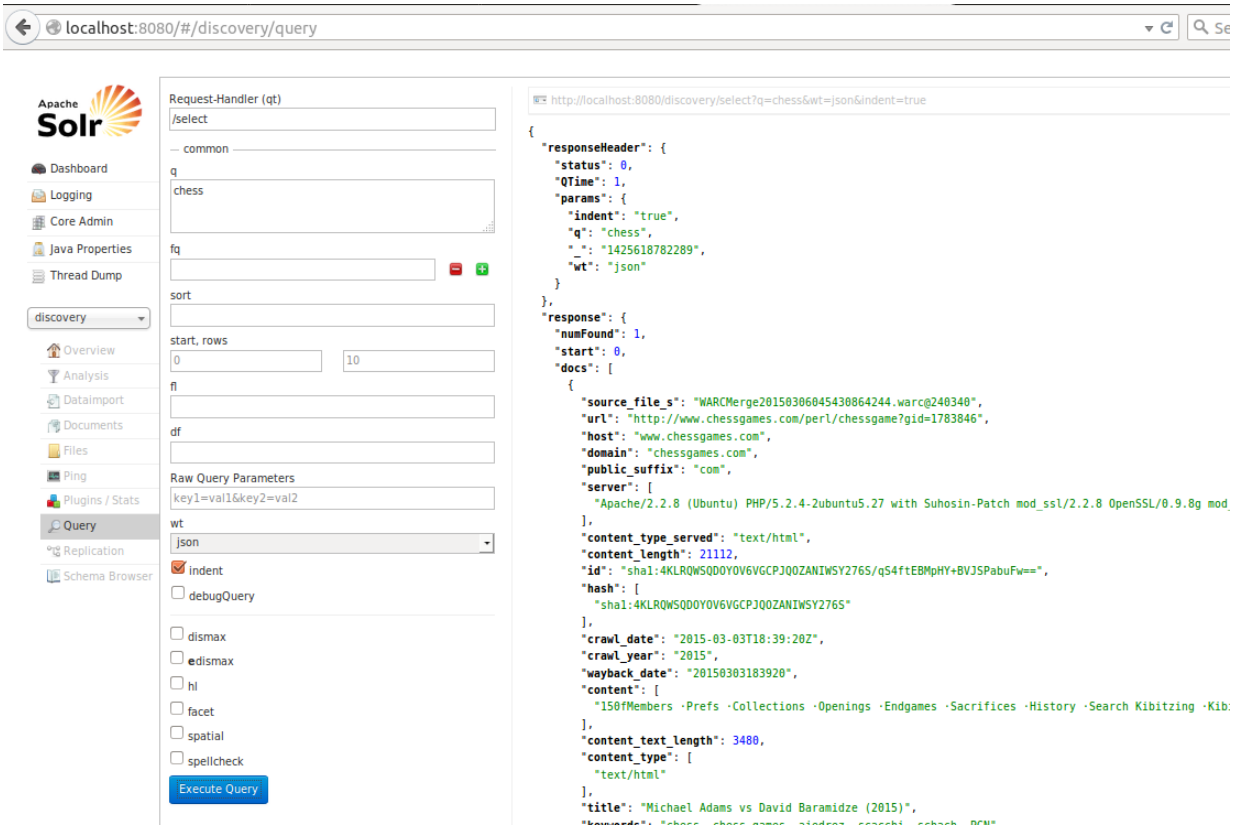


Fig. 2.2. Query: chess

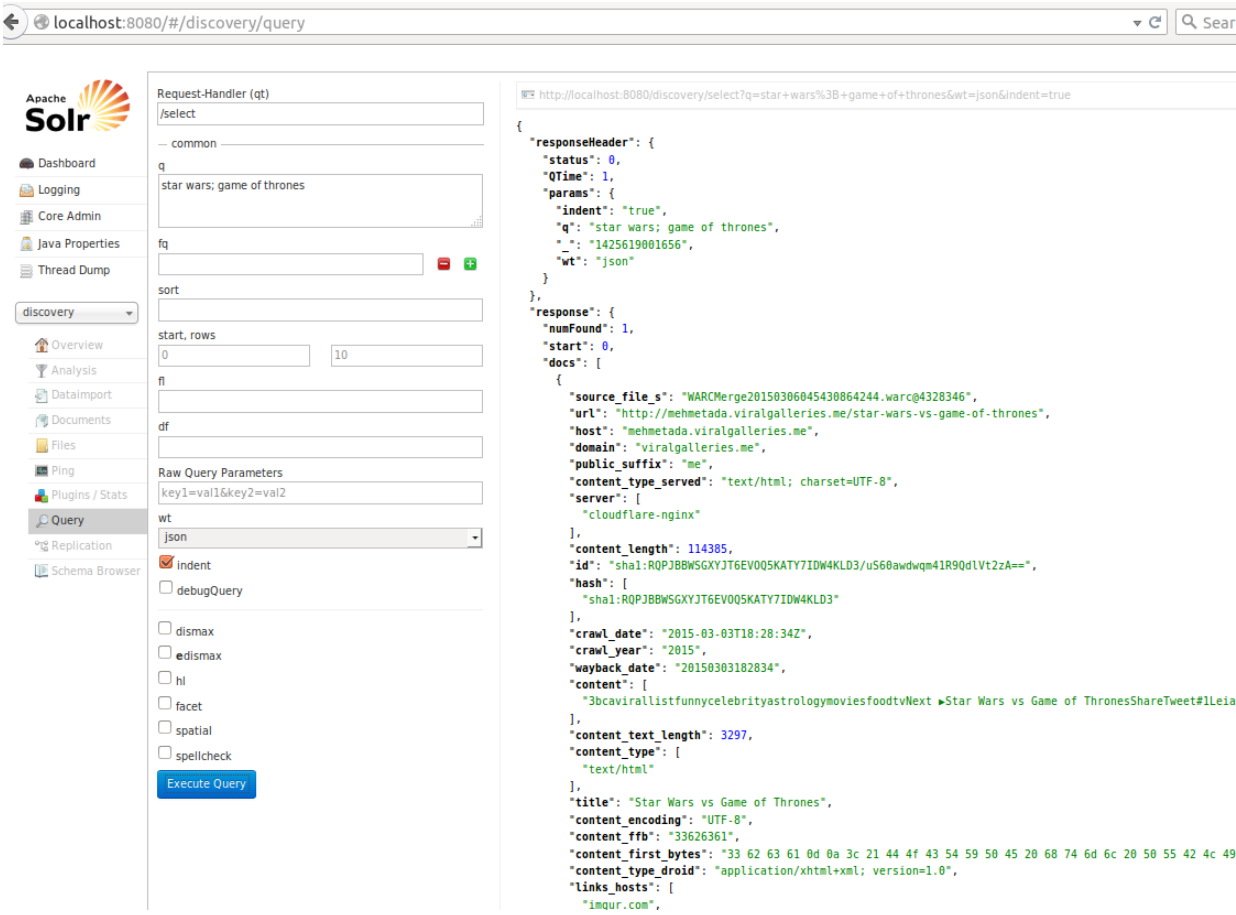


Fig. 2.3. Query: star wars; game of thrones

- From the previous query I toggled the case for the search parameter and the ranking of the results is modified as shown below.

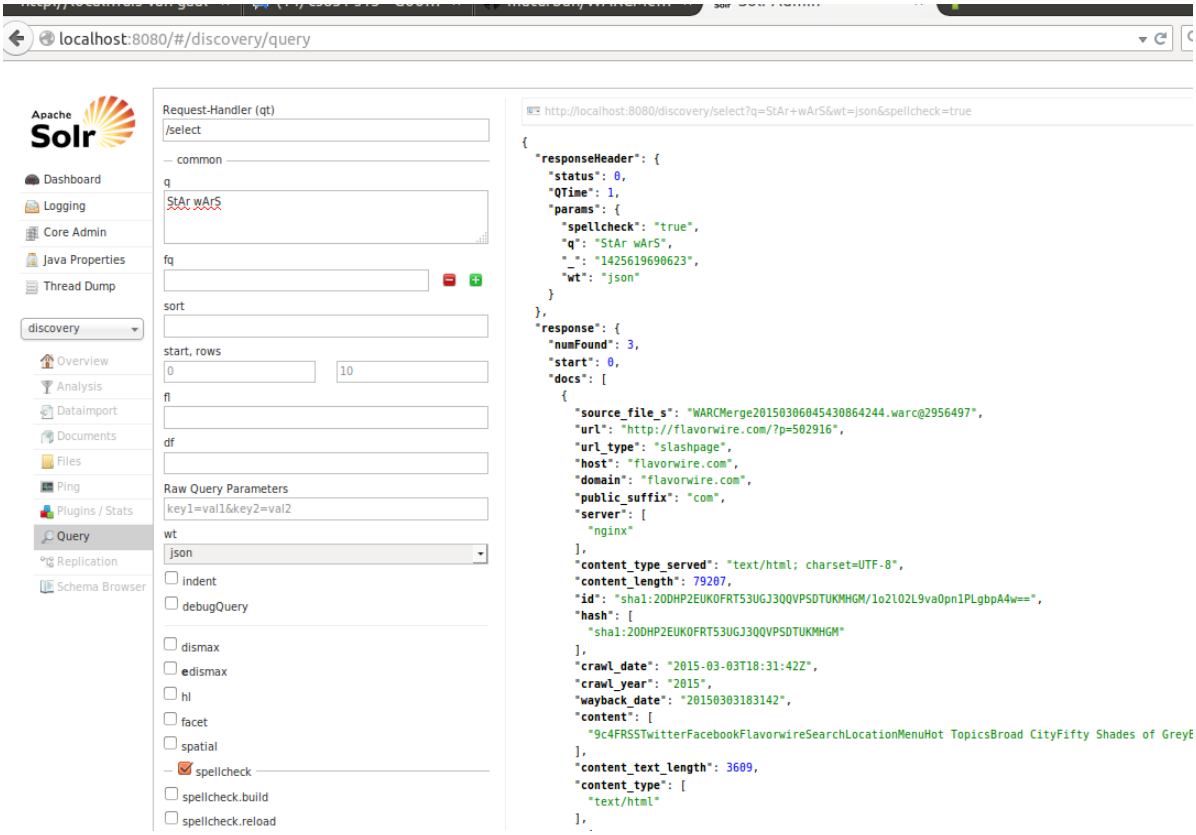


Fig. 2.4. Query: StAr wArS

References

1. R bar plots. http://www.theanalysisfactor.com/r-11-bar-charts/.
2. SOLR front ends. https://github.com/ukwa/webarchive-discovery/wiki/Front-ends.
3. User guide for pywb. https://github.com/ikreymer/pywb.
4. WAIL. http://matkelly.com/wail/.
5. WARC and ARC indexing and discovery tools. https://github.com/ukwa/webarchive-discovery.
6. WARC merge. https://github.com/maturban/WARCMerge.