THE UNIVERSITY of EDINBURGH
informatics

# Improved methodology for longitudinal Web analytics using Common Crawl

Henry S. Thompson
WebSci24, Universität Stuttgart, 22 May 2024

# Background

Generative AI (GAI) systems are only as good as the data they are trained on

By far the largest and most comprehensive source of training data for Natural Language GAI is Common Crawl (CC)

- CC has been collecting very large samples of the Web on a monthly or bi-monthly basis since 2008
- Making them freely available courtesy of Amazon since 2013
- In total it now contains more than 100 billion successful retrievals, more than 99% of which are HTML

The most high-profile use of CC is ChatGPT

- 80% of the training data for GPT-3 was taken from 4 years of CC
- $410 \times 10^9$ tokens out of $500 \times 10^9$

Despite its name, CC is not, in the original sense of the phrase, a web crawler

- That is, it doesn't collect data by fetching from some seed URIs and then following links to find more
- It starts each 'crawl' with a curated set of over $5 \times 10^9$ URIs
- And fetchs from this set until it reaches the desired size
- Latterly somewhere around $3.5 \times 10^9$ successful retrievals

# Are CC datasets *representative* of the Web?

After my students and I had been using CC datasets to research changes in Web usage over time

- Persistent identifier success
- Cookie usage
- Natural language use

We realised we needed to justify any claims we wanted to make

- That is, do results based on a sequence of CC datasets reliably reflect changes in the Web?

The methodology reported in the published paper describes in detail the approach we've developed

- It provides some good evidence in support of a 'yes' answer
- But actually *proving* this is made difficult by an obvious but often overlooked consequence of the shift to "Web 2.0" early in this century:
  - Less and less Web content is directly human-authored
  - Indeed it no longer makes sense to ask how big the Web is
  - A complete snapshot is not only infeasable in practice
  - It's in-principle impossible
  - Because much of the Web is being synthesised on-the-fly by servers, in response to properties of requests received

So *proving* a claim that *any* finite sample is representative of the Web is not amenable to simple statistical tests of significance

# CC segments and index

Each CC dataset is divided into 100 approximately equal **segments**

- The division is random, so each segment is meant to be representative of the whole

Assessing the representativeness of the segmentation is a logical first step to assessing the representativeness of the dataset itself

- But a CC dataset is *very* large
- The most recent dataset we used for this study was from August 2023
  - It's 98TB compressed in size
    - consisting of 90,000 WARC-format files
    - containing nearly $3.5 \times 10^9$ successful request/response records in total

So computing some plausible measure for each segment (900 files, $35 \times 10^6$ records) for comparison against the whole is a very substantial amount of computation

Fortunately since 2013 every CC dataset has an **index**

- Containing one JSON-array for each retrieval
- And it's only 223GB (compressed)

Explaining how it functions as an index would take too long

- What matters for our purposes is that it contains a dozen or so pieces of metadata about every retrieval, including
  - HTTP request URI, response code and body length
  - charset and media type (from `Content-type response header`)
  - Media type (as sniffed by Apache Tika)
  - In HTML responses, up to three languages as detected by CLD2.

# Our methodology: Efficient validation of CC's segmentation

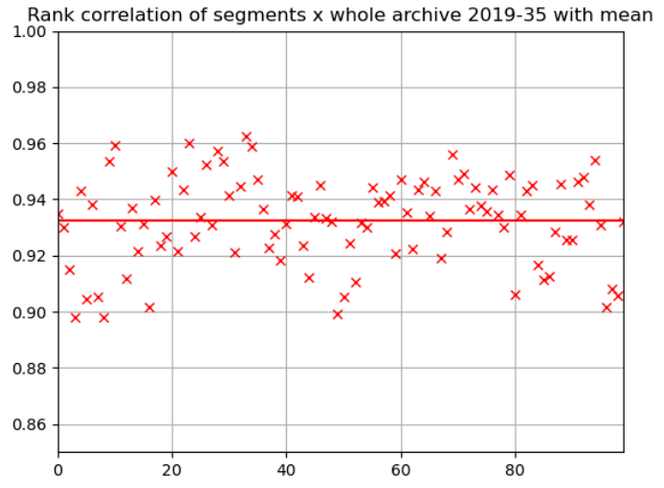
Using the index, we tabulated the distribution of media types for each segment each of four successive August datasets

- And merged them to get the complete distribution for each of the four:

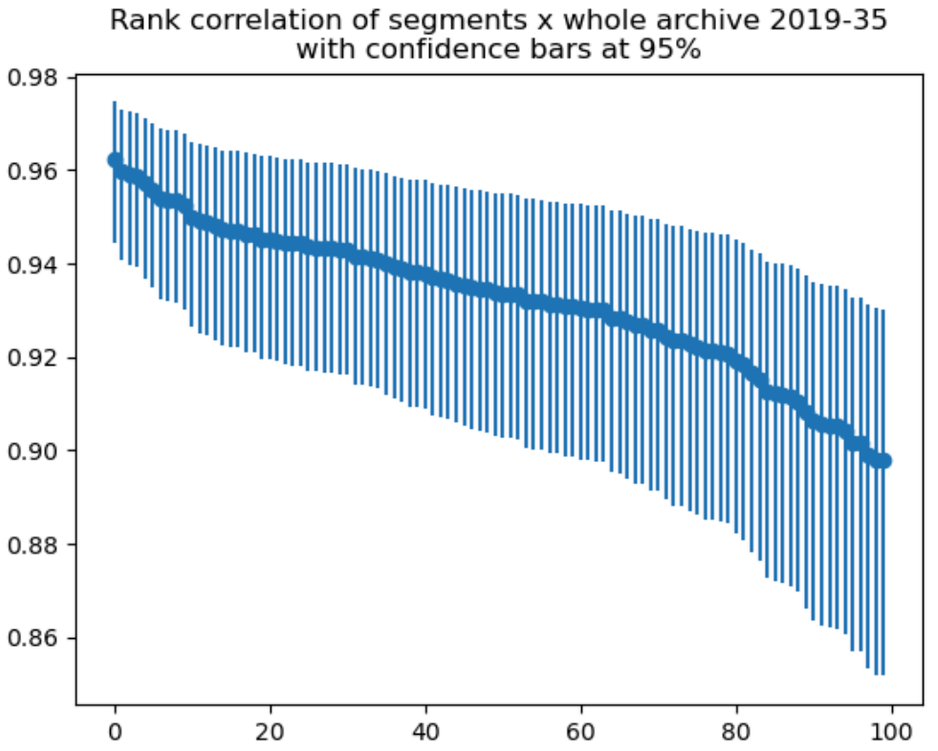
**Table 3: Sample from whole-archive mime tabulation from 2019-35**

| frequency | mime | mime-detected |
|---|---|---|
| 2,232,464,436 | text/html | ditto |
| 650,577,285 | text/html | application/xhtml+xml |
| 40,022,222 | unk | text/html |
| 3,985,789 | application/atom+xml | ditto |
| 3,879,977 | application/pdf | ditto |
| 3,741,189 | image/jpeg | ditto |
| 2,741,054 | unk | application/xhtml+xml |
| 2,488,581 | application/rss+xml | ditto |
| 1,565,481 | text/xml | application/rss+xml |
| 1,229,831 | text/plain | ditto |

We could then compare the (rank) correlation of each segment's distribution with that of the whole:




Re-ordering by correlation and adding 95% confidence bars:




QED: All the segments correlate well

- The best is significantly better than the bottom third
- The worst is signifantly worse than the top 10%

# Our methodology: Principled subsetting to reduce resource load

There are of course many things that can't be explored using only the index

- When only the actual response headers and/or body are relevant

But we can use the index to choose a representative subset of segments to reduce processing load

- Using the approach described just now

See the published paper for more details

# Improving temporal resolution using Last-Modified header data

For detailed study of Web evolution, we need something better than the date a page was retrieved

- It may have been created well before that

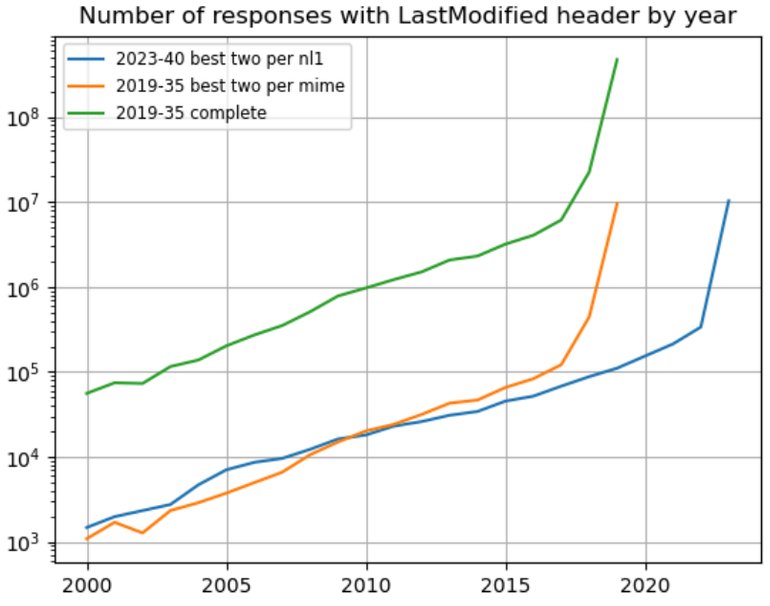
The Last-Modified HTTP response header offers more differentiated timing information.

- However it is neither required not guaranteed to be accurate
- For example in the complete August 2019 dataset we found only around 17% of the successful retrievals had a useable, credible, Last-Modified header
- But that's still $600 \times 10^6$, which is plenty for useful analyses

We have created and published an augmented version of the index for that August 2019 dataset which includes a canonicalised Last-Modified value where one is available

This allows us to look further back, and in more detail

- And, using the technique described earlier, we selected two representive segments from the August 2023 dataset to compare with data from the 2019-35 dataset, both complete and for two segments only



# Is CC's utility at risk?

Tim Berners Lee used to say that we need to study the Web so we don't break it by mistake

- If we don't figure out how to stop ChatGPT and its like from destroying Common Crawl's utility
- We may well end up breaking the Web as a source of useful information for the rest of us

Yet CC's value lies at least in part that it is *very* lightly curated

- And because it does *not* run Javascript, it may be somewhat biased, in a good way, towards human-authored text

GPT-3 was trained on CC datasets from before 2020

- But if generative AI systems start encorporating more recent CC releases in their training
- They will be training in part on their own output

Despite this, the major players in the space continue to resist attempts to get them to watermark or otherwise ineradicably identify their outputs

- Leading to a vicious circle of degrading informativeness

# One interim possibility to extend CC's utility

I've been using 'human-authored' as if it meant 'more useful/reliable/...'

- Which is naive at best

But there is one source of relatively reliable information on the Web which is largely absent from CC

- Almost by accident, scholarly papers from refereed journals are under-represented

CC is made freely available courtesy of Amazon

- Respecting their generosity means not wasting space
- In fact CC truncates all retrievals at 1MB
- The impact of this on HTML data is proportional
- But it's catastrophic for PDF data

The PDF format uses data dictionaries to reduce size

- And the dictionaries are serialised at the *end* of a file
- So truncation makes a PDF unreadable

PDFs make up less than 1% of successful retrievals

- And more than 25% of them are truncated and therefore useless

Furthermore, many scholarly journals are pay-walled, and so not available to CC at all

So it would make a big difference to the amount of high-quality information in CC to get rid of the 1MB limit for PDFs

# A more speculative possibility

[Thanks to Fernando Pereira of Google for a discussion which stimulated my thinking in this area]

ChatGPT doesn't plagiarise

- That is, it doesn't copy material word-for-word
- Rather, it synthesises paraphrases of one or more relevant sources

It has been suggested that this produces a recognisable ChatGPT *style*

- A sort of *non-distinctive* or *anonymous* style
- Might it be possible to automatically *detect* that style?

Stylistics has been a part of NLP since the very beginning

- Mosteller and Wallace *Inference and Disputed Authorship 1964*!

It seeems at least possible that more modern NLP techniques could be used to detect ChatGPT output?

- And at least annotate CC content going forward with an automaticity rating

# In conclusion

You can find the augmented index referred to above online, along with some introductory material, a worked example and these slides:

- https://markup.co.uk/ccrawl/