# Introduction to Web Graphs

Sebastian Nagel sebastian@commoncrawl.org
Pedro Ortiz Suarez pedro@commoncrawl.org
Thom Vaughan thom@commoncrawl.org
Greg Lindahl greg@commoncrawl.org

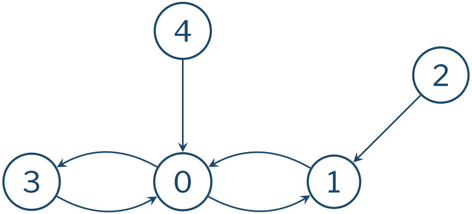
IIPC Web Archiving Conference 2025, 8 – 10 April 2025, Oslo, Norway

# The Webgraph or Hyperlink Graph

- The webgraph describes the link structure between pages of the World Wide Web [1]
- Web pages correspond to the nodes (or “vertices”) of the graph
- The hyperlinks connecting the web pages are the edges (or “arcs”)
- The webgraph is a directed graph because hyperlinks are unidirectional
- Web pages are (usually) represented by URLs

# Example Webgraph

A sample graph based on five Wikipedia pages:




| | |
|---|---|
| 0 | `https://en.wikipedia.org/wiki`/Webgraph |
| 1 | `https://en.wikipedia.org/wiki`/PageRank |
| 2 | `https://en.wikipedia.org/wiki`/Popularity |
| 3 | `https://en.wikipedia.org/wiki`/World_Wide_Web |
| 4 | `https://en.wikipedia.org/wiki`/Citation_graph |

# Aggregation Levels

- web pages / URL
- host part of the URL
- pay-level domain, registered domain, one level below the registry suffix
- top-level domain (TLD): `org`, `uk`
- example 1:

| | |
|---|---|
| `https://en.wikipedia.org/wiki/Webgraph` | page / URL |
| `https://en.wikipedia.org/wiki/Webgraph` | host |
| `en.wikipedia.org` | domain |
| `wikipedia.org` | TLD |

- example 2: `https://libguides.ials.sas.ac.uk/az/uk-web-archive`

  `libguides.ials.sas.ac.uk`

  `libguides.ial.sas.ac.uk`

# Aggregation Levels – Host and Domain

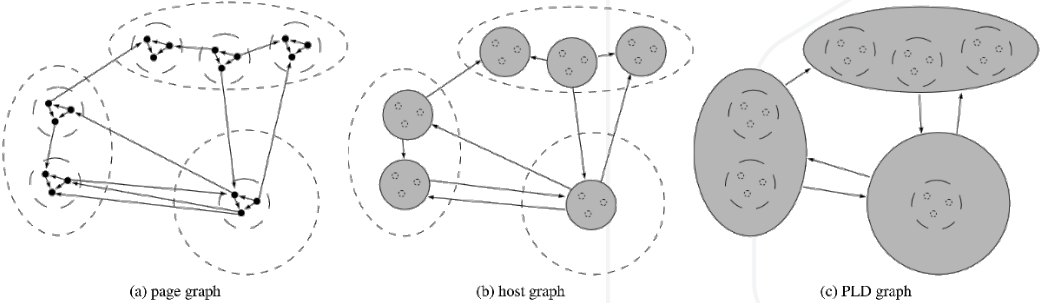
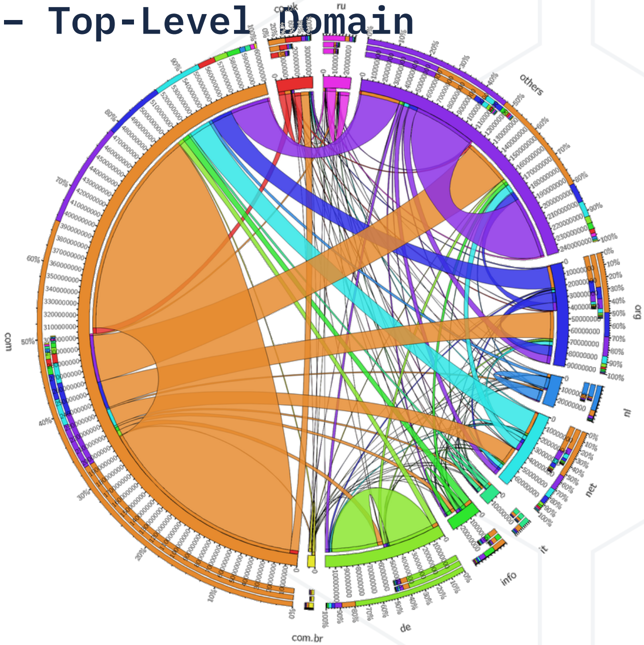



**Figure 1:** Page-level webgraph and aggregations on host and domain level [2]

# Aggregation Levels – Top-Level Domain




Top-level domain graph [3]

# Related Types of Graphs

- Citation graph
- Social network
  - Directed: Twitter, BlueSky, Mastodon, Instagram
  - Undirected: Facebook
- Software dependencies

# The WebGraph Framework

- Paolo Boldi, Sebastiano Vigna, Laboratory of Web Algorithms (LAW), University of Milano
- Framework for graph compression [4] and graph algorithms
- Java, developed over 20 years
- (in progress) Reimplementation in Rust [5]

# `LAW Libraries`

- `WebGraph` – efficiently store (compress) and work with “immutable” graphs, includes “HyperBall” to compute Harmonic Centrality
- `Sux4J` – map strings to integers
- `fastutil` – type-specific Java collections (small memory footprint) including big arrays (more than 2 billion items)
- `dsiutils` – various utils
- `law` – includes classes to compute PageRank, but also utility classes for WARC and crawling

# Overview WebGraph Classes

`BVGraph` – binary, compressed graph representation

- `basename.graph` – the graph itself
- `basename.properties` – text files with graph properties, including the class name
- `basename.offsets` – required for non-sequential access
- used as pair: graph and its “transpose” (inverted direction of arcs): `basename-t.*`

`ArcListASCIIGraph` – read/write textual graph representations

- nodes are integers from 0 to $n-1$
- one line for every arc: $\langle$source$\rangle$ $\langle$target$\rangle$
- numerically sorted by source and target

# BVGraph Intro

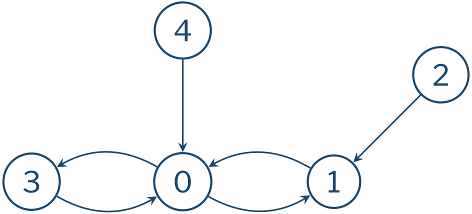
```
java it.unimi.dsi.webgraph.BVGraph -g ArcListASCIIGraph edges.txt exmpl

# Load exmpl.graph and convert it back to text (written to stdout)
java it.unimi.dsi.webgraph.ArcListASCIIGraph exmpl /dev/stdout

# Transpose of the graph
java it.unimi.dsi.webgraph.Transform transposeOffline exmpl exmpl-t

# Statistics
java it.unimi.dsi.webgraph.Stats --save-degrees exmpl
```

- instructions: `https://github.com/commoncrawl/wac2025-webgraph-workshop`
- working directory: `wac2025-webgraph-workshop/data/example-graph/`
- Java `CLASSPATH` set
- commands listed in process-example.sh)

# The Wikipedia Graph – Interactive Session

```
$> jshell --class-path "$CC_WEBGRAPH_JAR"

jshell> import org.commoncrawl.webgraph.explore.GraphExplorer

jshell> GraphExplorer e = new GraphExplorer("enwiki-2024")

jshell> e.ls("Webgraph")
jshell> e.sl("Webgraph")

jshell> /exit

# A JShell script loading the graph before starting the interactive session:
$> jshell \
     --class-path "$CC_WEBGRAPH_JAR" \
     -R-Dgraph="enwiki-2024" \
     "$CC_WEBGRAPH"/src/script/webgraph_ranking/graph_explore_load_graph.jsh
```

# Ranking Webgraphs – Harmonic Centrality




$$
\begin{aligned}
0:&\quad 3.5 = 3 + \tfrac{1}{2}\\
1:&\quad 3.0 = 2 + 2 \times \tfrac{1}{2}\\
2:&\quad 0\\
3:&\quad 2.33 = 1 + 2 \times \tfrac{1}{2} + \tfrac{1}{3}\\
4:&\quad 0
\end{aligned}
$$

# Ranking Webgraphs – PageRank

Paolo Boldi's explanation [8]:
`https://youtu.be/cnGJtGP4gL4?t=2044`

# WebGraph Classes For Ranking

```
# PageRank
java it.unimi.dsi.law.rank.PageRankParallelGaussSeidel \
     --alpha .85 --threads 2 --mapped exmpl-t exmpl-pr
java it.unimi.dsi.law.io.tool.DataInput2Text --type double exmpl-pr.ranks

# Harmonic Centrality via HyperBall
java it.unimi.dsi.webgraph.algo.HyperBall --threads 2 --offline --log2m 12 \
     --harmonic-centrality exmpl-hc.bin exmpl-t exmpl
java it.unimi.dsi.law.io.tool.DataInput2Text --type float exmpl-hc.bin
```

# Webgraphs Based on Common Crawl Data

2013—2015 Web Data Commons, University of Mannheim: hyperlink graphs and rankings [10, 11, 3, 2]

- Page/host/domain-level hyper-link graphs
- Host-level site ranking by harmonic centrality, pagerank, indegree centrality, Katz centrality [12]

2016 Common Search: host-level webgraph and pagerank [13, 14]

2017— “In-house” host/domain-level webgraph datasets by CCF [15, 16]

# Why the WebGraph Framework?

- Proven to work for ranking the Web Data Commons hyperlink graphs
- Main goal of the CCF webgraphs: graph-based rankings as relevance signal for the web crawls
- Frank McSherry [17, 18]: “throwing more machines at a problem isn’t necessarily the best approach. A laptop can outperform clusters when used effectively.”
- Same experience while evaluation and comparing Spark’s GraphX and the WebGraph framework

# Common Crawl Webgraph Datasets

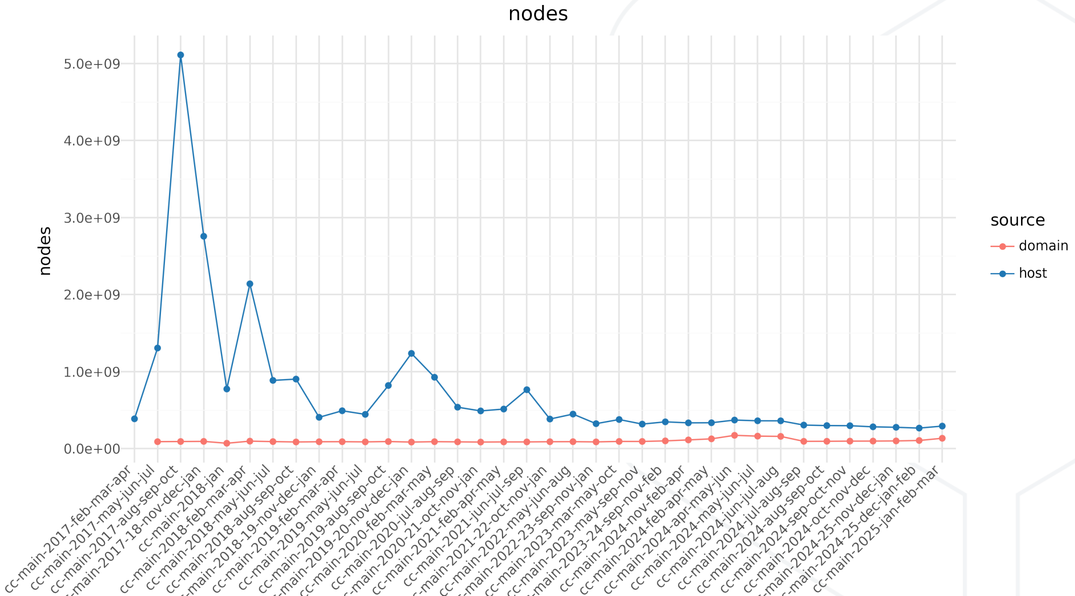
- One graph dataset combines three “monthly” crawl datasets
- Initially released quarterly
- Since monthly using a sliding window of the three latest crawls

- Only host and domain-level aggregations

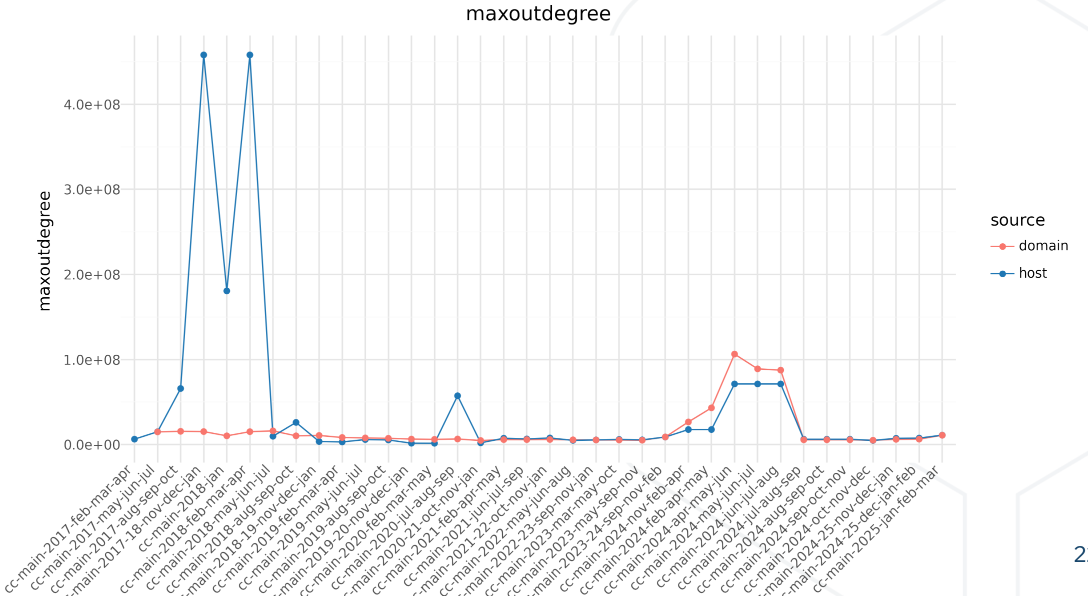
! A page-level graph would be too large and costly to build and rank

- A small dataset (only Gigabytes) but a good representation of the sample crawled by CCBot

# CCF Webgraph Datasets: Number of Nodes



# CCF Webgraph Datasets: Max Outdegree



# Common Crawl Webgraphs – Construction

Host-level graph (PySpark)

- extract links from WAT and redirects from WARC
- every link saved as pair ⟨from-host, to-host⟩ using reverse domain names
- sort and enumerate the host names – that's the vertices file(s)
- replace host names by numbers in from-to pairs – the edges file(s)

Domain-level graph (custom Java using WebGraph classes)

- clip the subdomains off the host name, based on the public suffix list
- hosts of one domain are in one block (because of rev. domain name sorting)
- assign a new consecutive domain ID and store it in an array
- convert and fold edges using the mapping of host-domain IDs

# The Need for Sampling

Why sampling and prioritization are necessary? Why not just follow links?

- An average “monthly” crawl includes 3 billion page captures with
  500+ billion links
  25+ billion unique URLs linked
- Up to 2.5 billion URLs listed in a single sitemap (sitemap index) [19]

Need to select a diverse and representative sample given

- Limited resources
- Requirements for crawler politeness: do not overload a single web site
- It’s easy to get lost in the wrong corner of the web!

# Stratified Domain-Level Sampling

Domain-level harmonic centrality ranks

- Define a “budget” [20] per registered domain
  - How many URLs/pages are sampled per domain
  - Domain: one level below the registry suffix, e.g. `w.org`, `data.gov.uk`)
- Are used during URL discovery to sample sitemaps or home pages (top-ranking domains: always, decreasing likelihood for lower ranks)
- Are “projected” to the page-level by inlink count or OPIC [21]
  - Rank the pages within a domain
  - ! We have no absolute “page quality metrics” comparing two pages from two different domains

# Domain-Level Graph-Based Ranking Example

- Top-N `.edu` domains ranked by harmonic centrality (or pagerank) calculated on CC's domain-level hyperlink graphs [22]
- Reverse domain name notation [23]
- Order by harmonic centrality ("hc") [7, 8]
  - ranks are shown not scores
  - PageRank rank [24], too
  - global ranks over domains below all top-level domains, not only `.edu`
- Includes not only universities (*)
- Compared with university rankings by QS World [25] and Forbes [26]

# Domain-Level Graph-Based Ranking Example

| pos | hc | pr | rev. domain |
|---|---|---|---|
| 1 | 71 | 297 | edu.stanford |
| 2 | 78 | 285 | edu.harvard |
| 3 | 90 | 392 | edu.mit |
| 4 | 135 | 588 | edu.berkeley |
| 5 | 157 | 757 | edu.psu |
| 6 | 167 | 515 | edu.cornell |
| 7 | 203 | 522 | edu.cmu |
| 8 | 213 | 978 | edu.princeton |
| 9 | 228 | 998 | edu.utexas |
| 10 | 236 | 818 | edu.columbia |
| 11 | 239 | 1011 | edu.yale |
| 12 | 249 | 1063 | edu.wisc |
| 13 | 268 | 1050 | edu.washington |
| 14 | 292 | 1358 | edu.brookings* |
| 15 | 300 | 1405 | edu.usc |
| 16 | 349 | 2076 | edu.ncsu |
| 17 | 352 | 1243 | edu.si* |
| 18 | 391 | 1824 | edu.georgetown |
| 19 | 397 | 1248 | edu.academia* |
| 20 | 398 | 1010 | edu.uchicago |

| rank | QS World [25] |
|---|---|
| 1 | MIT |
| 4 | Harvard |
| 6 | Stanford |
| 10 | Caltech |
| 11 | U. Pennsylvania |
| 12 | Berkeley (UCB) |
| 16 | Cornell |
| 21 | Chicago |
| 22 | Princeton |
| 23 | Yale |
| 32 | Johns Hopkins |
| 34 | Columbia |
| 42 | UCLA |
| 43 | NYU |
| 44 | Michigan-Ann Arbor |
| 50 | Northwestern |
| 58 | Carnegie Mellon |
| 61 | Duke |
| 66 | Texas at Austin |
| 69 | Illinois |

| rank | Forbes [26] |
|---|---|
| 1 | Princeton |
| 2 | Stanford |
| 3 | MIT |
| 4 | Yale |
| 5 | Berkeley |
| 6 | Columbia |
| 7 | U. Pennsylvania |
| 8 | Harvard |
| 9 | Rice |
| 10 | Cornell |
| 11 | Northwestern |
| 12 | Johns Hopkins |
| 13 | UCLA |
| 14 | Chicago |
| 15 | Vanderbilt |
| 16 | Dartmouth College |
| 17 | Williams College |
| 18 | Brown |
| 19 | Claremont McKenna |
| 20 | Duke |

# CCF Domain-Level Graph – Interactive Session

```
$> jshell \
     --class-path ”$CC_WEBGRAPH_JAR” \
     -R-Dgraph=”cc-main-2025-jan-feb-mar-domain” \
     ”$CC_WEBGRAPH”/src/script/webgraph_ranking/graph_explore_load_graph.jsh
```

- instructions: `https://github.com/commoncrawl/wac2025-webgraph-workshop`
- see also: `https://github.com/commoncrawl/cc-webgraph/blob/main/graph-exploration-README.md`

# Link Spam – Challenging the Crawler

- Spam is part of the web, it's ok if some is contained in the Common Crawl archives
- October 2017: the crawler hit a spam cluster
  - crawled: 56 million pages (1.5% of the crawl), 70,000 domains
  - known from links: 320,000 domains, 2.5 billion subdomains
- highly branching spam clusters are expensive for a crawler: every subdomain requires a DNS look-up and robots.txt fetch/caching
- measures: set limit of crawled subdomains per domain and try to detect and block the worst link spam clusters

# Link spam detection i

- spam clusters are volatile
- must detect spam with no training data
- simple heuristics proved to work with little supervision based on imbalances between
  - centrality score
  - outgoing and incoming links
  - number of subdomains

  low-ranking domains with too many outlinks or subdomains are suspicious
- once some nodes of a spam cluster are identified, other nodes are easily found by looking for a strongly connected subcluster in the graph

# Link spam detection ii

Example based on the Jan/Feb/March domain-level graph, taking as spam indicator an exceptionally high product of harmonic centrality rank and number of known subdomains

| sort | $log_2(r \cdot n)$ | hc rank $r$ | $n$ subdomains | domain |
|---:|---:|---:|---:|---:|
| 1 | 43.75 | 124417878 | 118695 | 520hlxy.com |
| 2 | 43.63 | 16473451 | 826151 | soukop.cz |
| 3 | 43.28 | 129410034 | 82390 | rxmuju.com |
| 4 | 43.25 | 130995134 | 80058 | ztxd1780.com |
| 5 | 43.25 | 127700773 | 82088 | kswy5288.com |
| 6 | 43.25 | 130026570 | 80477 | syxjwl.com |
| 7 | 43.24 | 128794755 | 80733 | ousendaoju.com |
| 8 | 43.22 | 124097521 | 82628 | lpdida7.cn |
| 9 | 43.21 | 126794407 | 80248 | gjphd.com |
| 10 | 43.21 | 125329529 | 81134 | borjacostasague.com |
| 11 | 43.20 | 130026406 | 77937 | sywlnz.com |
| 12 | 43.17 | 124148699 | 79552 | rzpec.cn |
| | | | | … |
| 3089376 | 27.00 | 30 | 4482266 | blogspot.com |
| | | | | … |

# Link spam detection iii

Imbalance between outdegree and indegree, sorted by $\frac{\text{outdegree}}{(\text{indegree}+1)}$ in descending order

| outdegree | indegree | $n$ subdomains | domain |
|---|---|---|---|
| 5654635 | 2 | 1 | yktsk.top |
| 2616082 | 1 | 1 | indiabacklink.com |
| 690349 | 0 | 1 | websiteprotools.com |
| 687179 | 0 | 1 | packersandmoversdirectory.com |
| 2293088 | 3 | 1 | universalpackersandmovers.com |
| 686732 | 1 | 1 | addondirectory.com |
| 1017892 | 3 | 22 | linksjump.info |
| 428760 | 1 | 1 | livebacklinks.com |
| 592821 | 2 | 1 | zuevwndpl.com |
| 3833680 | 21 | 1 | faithwebsites.net |
| 5270950 | 47 | 1 | fastoq.com |
| 1825600 | 17 | 1 | selfie-battles-are-for-amateurs-tim–kalin-from-seodomains-here.com |
| 1095920 | 10 | 1 | ageokousei.jp |
| 98456 | 0 | 1 | yavatmal.site |
| 98427 | 0 | 1 | dharashiv.site |
| 2243564 | 22 | 2 | moneygame.pro |
| 96386 | 0 | 1 | latur.site |
| 49048 | 0 | 1 | trackdesk.de |
| 41796 | 0 | 1219 | com.0556ms.cn |
| 10968122 | 276 | 1 | sergechel.info |

# Questions?

# References i


[1] ***Webgraph.*** https://en.wikipedia.org/w/index.php?title=Webgraph.

[2] Robert Meusel et al. **"The Graph Structure in the Web – Analyzed on Different Aggregation Levels".** In: *The Journal of Web Science* 1.1 (2015), pp. 33–47. https://pdfs.semanticscholar.org/b5d5/88298e6845b4bfd40ea779ce21e628239ef3.pdf.

[3] Oliver Lehmberg, Robert Meusel, and Christian Bizer. **"Graph structure in the web: aggregated by pay-level domain".** In: *Web Science Conference*. 2014. https://dl.acm.org/doi/10.1145/2615569.2615674.

[4] Paolo Boldi and Sebastiano Vigna. **"The WebGraph framework I: Compression techniques".** In: WWW '04 (2004), pp. 595–602. https://doi.org/10.1145/988672.988752.

# References ii

[5] Tommaso Fontana, Sebastiano Vigna, and Stefano Zacchiroli. **“WebGraph: The Next Generation (Is in Rust)”.** In: *Companion Proceedings of the ACM on Web Conference 2024*. WWW ’24. Singapore, Singapore: Association for Computing Machinery, 2024, pp. 686–689. `https://doi.org/10.1145/3589335.3651581`.

[6] Ian H. Witten, Marco Gori, and Teresa Numerico. **“Web Dragons: Inside the Myths of Search Engine Technology”.** In: 2006. `https://api.semanticscholar.org/CorpusID:1192963`.

[7] Paolo Boldi and Sebastiano Vigna. **“Axioms for Centrality”.** In: *CoRR* abs/1308.2140 (2013). `https://arxiv.org/abs/1308.2140`.

[8] Paolo Boldi. ***A modern view of centrality measures.*** 2013. `https://www.youtube.com/watch?v=cnGJtGP4gL4`.

# References iii


[9] Paolo Boldi and Sebastiano Vigna. **"In-Core Computation of Geometric Centralities with HyperBall: A Hundred Billion Nodes and Beyond".** In: *2013 IEEE 13th International Conference on Data Mining Workshops* (2013), pp. 621–628. `https://vigna.di.unimi.it/papers.php#BoVHB`.

[10] ***Web Data Commons - Hyperlink Graphs.*** 2013. `https://webdatacommons.org/hyperlinkgraph/index.html`.

[11] Robert Meusel et al. **"Graph Structure in the Web — Revisited".** In: (2014). `http://vigna.di.unimi.it/ftp/papers/GraphStructureRevisited.pdf`.

[12] ***The Common Crawl WWW Ranking.*** `http://wwwranking.webdatacommons.org/`.

# References iv


[13] ***Common Search: Our first public datasets: Host-level WebGraph and PageRank.*** https://web.archive.org/web/20170729110709/https://about.commonsearch.org/2016/07/our-first-public-datasets-host-level-webgraph-and-pagerank/.

[14] https://github.com/commonsearch/cosr-back/blob/master/spark/jobs/pagerank.py.

[15] ***Web Graphs.*** https://commoncrawl.org/web-graphs.

[16] ***Web Graphs Statistics.*** https://commoncrawl.github.io/cc-webgraph-statistics/.

[17] Frank McSherry. ***Scalability! But at what COST?*** 2015. https://www.frankmcsherry.org/graph/scalability/cost/2015/01/15/COST.html.

# References v


[18] Frank McSherry. ***Bigger data; same laptop.*** 2015. `https://www.frankmcsherry.org/graph/scalability/cost/2015/02/04/COST2.html`.

[19] ***sitemaps.org.*** `https://www.sitemaps.org/protocol.html`.

[20] Hsin-Tsang Lee et al. **"IRLbot: Scaling to 6 Billion Pages and Beyond".** In: *ACM Trans. Web* 3.3 (July 2009). ISSN: 1559-1131. `https://doi.org/10.1145/1541822.1541823`.

[21] Serge Abiteboul, Mihai Preda, and Gregory Cobena. **"Adaptive on-line page importance computation".** In: (2003). `https://dx.doi.org/10.1145/775152.775192`.

[22] ***Host- and Domain-Level Web Graphs October, November, December 2024.*** `https://commoncrawl.org/blog/host--and-domain-level-web-graphs-october-november-and-december-2024`.

# References vi


[23] ***Reverse domain name notation.*** https://en.wikipedia.org/wiki/Reverse_domain_name_notation.

[24] ***PageRank.*** https://en.wikipedia.org/wiki/PageRank.

[25] ***QS World University Rankings: The top 100 universities in the USA.*** https://www.topuniversities.com/where-to-study/north-america/united-states/ranked-top-100-us-universities.

[26] ***Forbes America's Top Colleges List 2025 - Best US Universities Ranked.*** https://www.forbes.com/top-colleges/.