COMMON CRAWL

FIELD GUIDE

TRAINING DATA IS THE NEW RANKING FACTOR

The AI Visibility Audit.

How to check whether your site reaches AI systems, and how to stay visible in the crawl that trains them. A practical playbook for SEOs and GEOs.

Stephen Burns

Web Intelligence Lead, Common Crawl Foundation

commoncrawl.org

WHY THIS GUIDE EXISTS

The crawl decides what AI knows.

> There is a layer of search work that sits upstream of everything most of us audit. Before on-page optimisation, before technical SEO, before link building, a website has to be reachable by the crawlers that feed AI training data. If it is not, a page can rank beautifully in Google and still be invisible to ChatGPT, Gemini, Claude, and Perplexity.

This guide turns that idea into a repeatable deliverable. It explains how AI systems discover content, where SEO intervention actually happens, and how to run a five-check AI Visibility Audit using free tools. It is written for SEOs and GEOs (generative engine optimisation practitioners) who want a concrete framework rather than theory. Three gut-checks to set the scene: has a client asked why their content isn't in AI answers, have you audited a site's AI crawler access in the last six months, and could you explain right now why a high-ranking page might be invisible to a model?

TWO LINES THAT DECIDE A LOT

A website's reach into AI can come down to its `robots.txt`. The lines below either open the door to Common Crawl's bot, the dataset behind a great deal of LLM training, or quietly close it. Naming `CCBot` explicitly is clearest, but a permissive wildcard works too: a website with no AI-crawler disallow rules is open by default, since the absence of a matching rule means access is allowed.

```
# Name the bot explicitly…
User-agent: CCBot
Allow: /
# …or use a permissive wildcard (any UA without its own rule):
User-agent: *
Allow: /
```

What's inside

- **01 · What controls AI training data.** The infrastructure layer most SEOs have never looked at, and why training-data inclusion behaves like a ranking factor.
- **02 · How the Web Graph sets crawl priority.** Harmonic Centrality, the timing lag between publishing and appearing in answers, and the two layers you live in once you're in.
- **03 · The invisible blocking problem.** How CDN and WAF defaults silently disallow AI crawlers, why some publishers legitimately want out, and why AI leans towards English.
- **04 · The AI Visibility Audit.** Five checks, how to verify a request really is CCBot, the free toolkit, and where to start.

SECTION 01

What controls AI training data.

The upstream infrastructure layer most SEOs have never seen, and the nonprofit that quietly became part of it.

Meet the Common Crawl Foundation

Common Crawl is a nonprofit, founded in 2007 by Gil Elbaz, whose mission is to democratise access to web information by producing and maintaining an open repository of web crawl data that is universally accessible. In practice, it runs a bot called **CCBot** that crawls the open web every month and publishes the results as free, downloadable archives on Amazon S3.

That archive matters to anyone working in search because it became a primary training source for large language models. When OpenAI and others began building modern LLMs, Common Crawl was one of the datasets they reached for. So a decision about whether CCBot can crawl your site is, indirectly, a decision about whether your content can enter the data these models learn from.

10+ PB
Open archive accumulated since 2008, in the petabyte range

~2–2.5B
Web pages captured per monthly crawl

300B+
Total pages captured across the full corpus

A single monthly snapshot runs to roughly 350–400 TiB of uncompressed data, and the dataset is cited in thousands of research papers. The exact per-crawl page count fluctuates month to month because of seed lists, revisit scheduling, and crawler operations, so treat any single figure as approximate.

Sources: *Common Crawl Foundation corpus documentation and cc-crawl-statistics (commoncrawl.github.io). Mozilla Foundation, "Training Data for the Price of a Sandwich" (~9.5 PB as of mid-2023). Per-snapshot page and size figures vary by crawl and over time; ranges shown reflect that variance.*

How AI systems actually discover content.

There are four steps between the open web and a model knowing your content. SEO intervention happens at step one, and most SEOs have never looked at it.

1 **CCBot (and other crawlers) crawl the web**

Bots fetch publicly accessible pages, following links and respecting `robots.txt`. This is the only step you directly control through site configuration, and it is where access is won or lost.

2 **Raw data enters the archive**

Captured pages are written to monthly snapshots (WARC, WAT, WET, and CDX index files) and published openly. If you weren't crawled, you simply aren't in the snapshot.

3 **AI labs filter and train**

Model builders pull from the archive (and other sources), filter for quality, and train. Inclusion here is downstream of being present and being judged worth keeping.

4 **The model knows your content**

Once trained, the model can surface, paraphrase, and recommend what it learned. That is the visibility payoff, and it traces all the way back to step one.

THE CORE CLAIM

- **Training-data inclusion behaves like a ranking factor.** It operates upstream of on-page optimisation, technical SEO, and link building.
- **It is set by infrastructure decisions** (CDN configuration, robots.txt, server-side rendering), not by content strategy alone.
- **Most of the industry has no framework for it yet.** That gap is the opportunity this guide is built around.

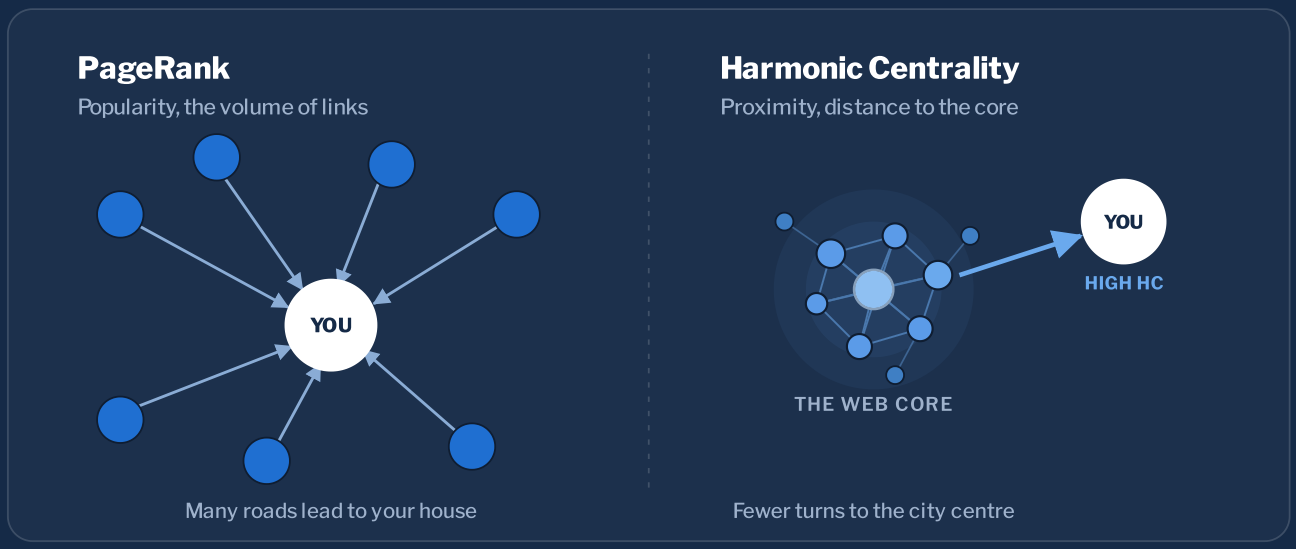
SECTION 02

How the Web Graph sets crawl priority.

Harmonic Centrality and timing. Why some websites get crawled deeply and often, and how to earn that priority.

Harmonic Centrality drives crawl priority

Common Crawl publishes a Web Graph that maps the link structure of the web at host and domain level, and derives a centrality measure from it. The intuition: a domain near the core of the web's link topology is more "central" than one stranded at the edge, and central domains get prioritised for crawling. This is a different question from classic PageRank-style popularity.



PageRank asks how many important sites link to you. Harmonic Centrality asks how close you sit to the web's core. A single link from a core site can lift centrality more than dozens from isolated ones, which is why connectivity, not raw link volume, drives crawl priority.

The practical consequence: link quality compounds. Higher centrality means crawled more often, which means more pages in archives, more training data, and more chance models know and recommend you. For an SEO this reframes link building around connectivity to the core of the graph the crawler prioritises, not just referral authority.

Sources: *Common Crawl Web Graph documentation (commoncrawl.org/web-graphs). Harmonic Centrality is the centrality measure CCF publishes for its host- and domain-level graphs. Crawl-prioritisation behaviour per CCF release notes and Sebastian Nagel (CCF) guidance.*

The timing problem, and the two layers.

Publishing a page and appearing in an AI answer can be months apart. Understanding the lag, and where your content lives once it's in, changes how you set client expectations.

The lag between publishing and appearing

There is a pipeline delay that has no equivalent in classic SEO, where a recrawl can surface a change in days. For training-data visibility the chain is longer:

- **Page is published** on your site.
- **CCBot discovers it** on a future crawl pass, which depends on your centrality and link discovery.
- **A monthly crawl captures it** into a snapshot.
- **The archive is published** openly.
- **AI training consumes it** in a later model build or retraining cycle.

Each hop adds latency. The takeaway for client work: parametric AI visibility is a slow-moving asset. Get the access and connectivity right early, because the payoff arrives on the model's schedule, not yours.

Once you're in, you live in two layers

PARAMETRIC MEMORY

Pre-cutoff content, baked into the weights. This is what the model "knows" without looking anything up. It depends on having been crawled and trained on before the model's training cutoff date.

RETRIEVAL (RAG)

Post-cutoff content, fetched live at query time. This depends on being accessible to retrieval/search bots right now, which is a separate access question from training crawlers.

The two layers have different access requirements. A website can be in parametric memory but blocked from live retrieval, or vice versa. A complete audit checks both: were you reachable when the model trained, and are you reachable for live fetch today?

Framework: *Duane Forrester, "When the Training Data Cutoff Becomes a Ranking Factor," March 2026. The parametric-vs-retrieval distinction reflects how current models combine trained knowledge with live retrieval; specifics vary by model and provider.*

SECTION 02 · CONTINUED

Why training cutoffs matter.

Each model has a training cutoff. Content published after it lives only in retrieval, not in the model's baked-in memory, until the next training cycle.

Cutoff dates are published by the model providers and move with each release. The pattern that matters for visibility work: the further back a model's cutoff, the more it relies on retrieval for anything recent, and the more your live crawler access (not just training access) determines whether you show up. Treat the dates below as illustrative of the pattern; always confirm the current figure in the provider's own documentation before quoting it to a client.

MODEL FAMILY	KNOWLEDGE SOURCE PATTERN	LAYER EMPHASIS
GPT family	Trained knowledge to a stated cutoff, plus live search for newer queries	Parametric + RAG
Gemini	Trained knowledge plus Google retrieval integration	Parametric + RAG
Claude	Trained knowledge to a stated cutoff, with retrieval where tools are available	Parametric + RAG
Perplexity	Retrieval-native: answers are built primarily from live fetches	RAG-first

WHAT TO TELL CLIENTS

"To appear in what the model already knows, you needed to be crawlable before its cutoff. To appear in answers about anything recent, you need to be crawlable for live retrieval today. Both require that your infrastructure isn't quietly blocking the relevant bots."

On verification: *Specific cutoff dates change frequently and differ by model version. This guide deliberately does not pin exact dates, since they cannot be guaranteed accurate at the time of reading. Confirm against OpenAI, Google, Anthropic, and Perplexity documentation directly. Framework attribution: Duane Forrester, March 2026.*

SECTION 03

The invisible blocking problem.

The most common way to vanish from AI is not a deliberate choice. It is a default setting in a CDN or WAF that no one reviewed.

How websites block AI crawlers without meaning to

A page can rank well in Google and still be unreachable by AI crawlers, because the block lives at the edge, not in the content. Two mechanisms account for most of it:

- **Managed robots.txt.** Some CDNs can inject AI-crawler disallow rules into your robots.txt automatically. A managed-robots.txt feature typically creates the file if you don't have one, or prepends its own rules to the top of your existing file, disallowing known AI user agents such as GPTBot, ClaudeBot, Google-Extended, and CCBot.
- **WAF / bot-management blocks.** A "block AI bots" toggle or bot-fight rule rejects requests by user agent or behaviour at the firewall, before they ever reach your server. The site owner sees nothing in their own files; the bot sees a 403.

The reason this is widespread rather than rare: a single major CDN can sit in front of a large share of the web, and some providers began defaulting new domains toward blocking AI crawlers. So the block can be the out-of-the-box state, not something anyone opted into.

```
## A managed-CDN robots.txt block, prepended automatically:
User-agent: CCBot
Disallow: /

User-agent: GPTBot
Disallow: /
```

Sources: *Major CDN provider documentation on managed-robots.txt and AI-bot blocking features. Industry reporting that some CDNs began default-blocking AI crawlers on new domains (e.g. Mersel AI, Mar 2026; IndexLab, Oct 2025). Provider docs note that an AI-bot block rule can take precedence over other bot-management rules.*

How big is the blocking problem?

Large and growing fast. The exact share depends on what you measure, but every credible dataset points the same direction: AI crawlers are now the most-blocked agents on the web.

Headline figures vary because studies measure different things: different crawler lists, different site samples (top-1k vs news vs all-hosted), and different methods (robots.txt parsing vs observed traffic). Here are the load-bearing, checkable numbers, each tied to its source.

WHAT THE RESEARCH SHOWS

A peer-reviewed arXiv study found AI-blocking by reputable news sites rose from **23% in September 2023 to nearly 60% by May 2025**, with those sites disallowing an average of 15.5 AI user agents.

Originality.AI's analysis of the top 1,000 websites found GPTBot blocked by **35.7%** as of August 2024, up roughly sevenfold from about 5% a year earlier. CCBot was blocked by 22.1% in the same dataset.

The effect concentrates at the top: network-level analysis by a major CDN found **40% of the top 10 properties block AI bots**, scaling down to 8.8% of the top 1,000 and 2.98% of the top one million. The bigger and more visible the site, the likelier it blocks.

THE COMMON CRAWL ANGLE

In the news vertical, a January 2026 BuzzStream study of 100 top US and UK news sites found CCBot was the single most-blocked training bot at **75%**, ahead of Anthropic-ai (72%), ClaudeBot (69%), and GPTBot (62%). For a news or publisher client, assume CCBot access is the default casualty and check it first.

The good news: getting back in is straightforward

Because well-behaved crawlers like CCBot respect `robots.txt`, the fix is usually as simple as the block was. Allow the AI user agents at the edge and in your robots.txt, and crawlers resume on their normal schedule. The sooner a website is open, the sooner it starts accumulating presence in monthly archives, and that presence compounds over time. Opening access today is the highest-leverage, lowest-effort move most websites can make toward AI visibility.

Sources: *"Is Misinformation More Open? A Study of robots.txt Gatekeeping on the Web," arXiv 2510.10315 (Oct 2025). Originality.AI top-1,000 robots.txt analysis (Aug 2024). Network-level AI-bot access and blocking data by site rank from a major CDN's public traffic analysis. BuzzStream, "Which News Sites Block AI Crawlers" (Jan 2026), via Search Engine Journal.*

The honest other side: some will want out.

This guide is about helping websites that want to be included. It would be dishonest to pretend that is everyone.

Plenty of publishers have good reasons to keep their work out of AI training, whether on principle, on commercial grounds, or because they are actively opposed to how their content would be used. That is a legitimate choice, and it deserves tooling that actually does what the publisher thinks it does. The same access checks in this guide work in reverse: they let you confirm you are genuinely excluded, not just assuming you are.

A STRUCTURAL PROBLEM WORTH KNOWING

The harder truth is that opting out is not as reliable as most people believe. Researchers (Liao et al., ACM CCS 2025) identified several thousand domains configured so that the publisher believes they have opted out, when in fact they have not. The mechanisms are inconsistent across crawlers, and a rule that stops one bot may do nothing to another. Intent and outcome can quietly diverge.

This is the most credible criticism of the current crawling ecosystem: publishers do not yet have reliable tools to express what they actually want. Common Crawl agrees with that criticism and is working on it directly. It runs an Opt-Out Registry, a single place to register preferences including for content already in older crawls, and it is an active participant in the IETF AIPREF working group developing a richer, standardised vocabulary for crawler preferences carried over robots.txt and HTTP headers.

Sources: *Opt-out misconfiguration at scale: Liao et al., ACM CCS 2025. Common Crawl Opt-Out Registry (published September 2025, commoncrawl.org/blog). IETF AIPREF working group (chartered 2025), with participants including major publishers, CDNs, browser vendors, and AI providers.*

AI leans towards English.

For a non-English market, a page can rank well locally and still lose the AI citation to its English equivalent. Two biases stack to cause it.

Why it happens: two biases, one root cause

The root cause is the corpus. English is roughly **41% of pages** in the latest Common Crawl (40.85% of CC-MAIN-2026-21, computed from Common Crawl's official languages.csv), and that share is effectively higher after quality filtering. On top of that base, two separate biases compound:

- **Retrieval bias (the lever you can pull).** AI search retrievers tend to rank English and high-authority pages above equivalent localized ones, an effect that worsens when the query is in or normalised to English. This is documented across comparable multilingual RAG systems, where the retriever, not the generator, is the bottleneck in cross-lingual ranking.
- **Model bias (harder to influence).** Large models appear to compute in a shared concept space that tilts English, so even a parametric answer leans that way. Anthropic reports Claude uses concepts that are partly shared across languages, with that shared circuitry growing as models scale.

HOW TO STATE THIS ACCURATELY

Say models *tilt* toward English, not that they "translate to English." The internal-pivot studies ran on open models (Llama-2, Aya, Gemma), whose internals are inspectable; GPT's and Claude's are not. Anthropic frames Claude's shared space as language-agnostic concepts, not English dominance. And no lab publishes how its production answer engine selects or ranks source URLs, so treat pipeline specifics as inferred, not disclosed.

The lever: publish a strong English edition

Because the retriever is where the bias is most actionable, a high-quality English version of your cornerstone pages is one of the few content-side moves that widens your footprint. Mirror your highest-value pages first, use proper hreflang and distinct, crawlable URLs, keep the English genuinely useful rather than machine-translated filler, and make sure both versions clear the access checks in this guide.

Sources: *English share computed from Common Crawl languages.csv, CC-MAIN-2026-21 (commoncrawl.github.io/cc-crawl-statistics). Retrieval bias and retriever-as-bottleneck: Wu et al. (2024) and related multilingual RAG studies on arXiv (direction varies by setup; some find in-language bias). Shared multilingual concept space: Anthropic, "Tracing the thoughts of a large language model" (2025). Internal English-pivot findings: Wendler et al. (Llama-2) and related work on Aya/Gemma, open models only. Production answer-engine ranking is not publicly disclosed by the labs.*

SECTION 04

The AI Visibility Audit.

Five checks. About 90 minutes. A new category of client deliverable that most agencies don't offer yet.

Everything in the first three sections points to one practical service you can sell now. The checks below move from the most decisive (can the bot even reach you?) to the more strategic (will what it reaches be usable and prioritised?). All five use free tools. Run them in order; an early failure often explains later ones.

WHY NOW

Most agencies don't offer an AI Visibility Audit yet. The frameworks and free tooling already exist, and clients are increasingly asking why they're missing from AI answers. That gap is the window.

The five checks at a glance

- **1. CCBot access check**: can AI crawlers physically reach the site?
- **2. CC Index coverage audit**: is the domain actually in the archive, and how recently?
- **3. Harmonic Centrality check**: is the domain prioritised or deprioritised for crawling?
- **4. Structured data completeness**: is the content easy to represent in training?
- **5. Server-side rendering audit**: does the content exist without JavaScript execution?

THE FIVE CHECKS · 1 OF 5 & 2 OF 5

Access & coverage.

Start with whether the bot can reach you, then whether it actually has.

1 CCBot access check

Confirm nothing is disallowing AI crawlers, in robots.txt *or* at the edge. Check the live robots.txt for CCBot (and GPTBot, ClaudeBot, Google-Extended) disallow lines. Then test the firewall layer, because a clean robots.txt means nothing if the WAF returns a 403 by user agent.

`Manual + Screaming Frog (custom UA)`

```
# 1. Read the live robots.txt
curl -s https://example.com/robots.txt

# 2. Does the server actually serve the bot, or block it?
curl -A "CCBot/2.0" -I https://example.com/
# Look for: HTTP/2 200  (good)  vs  HTTP/2 403 (blocked at edge)

# 3. Compare against a normal browser UA to confirm UA-based blocking
curl -A "Mozilla/5.0" -I https://example.com/
```

WHAT A PASS LOOKS LIKE

No AI-bot disallow in robots.txt, and a `200` response to the CCBot user agent. If the browser UA gets 200 but CCBot gets 403, you've found an edge/WAF block, fixable in the CDN dashboard, not in the site's files.

2 CC Index coverage audit

Now check reality, not just permission. Query the Common Crawl Index to see whether the domain is present, when it was last crawled, and roughly how many pages are indexed. A domain can be open but barely crawled.

`index.commoncrawl.org (free API)`

```
# Query a recent crawl index for a domain (swap in a current crawl ID)
curl "https://index.commoncrawl.org/CC-MAIN-2026-21-index?\
url=example.com/*&output=json" | head
# No results = not in snapshot. Few = shallow. Old timestamps = stale.
```

Tools: *Common Crawl Index Server (index.commoncrawl.org), free and public. Screaming Frog supports custom user-agent crawls. Crawl IDs follow the* `CC-MAIN-YYYY-WW` *pattern; use a current one from the CCF site.*

THE FIVE CHECKS · VERIFICATION

Is it really CCBot?

The user-agent string is not proof of identity. Anyone can send a request that says CCBot. Before you trust crawl numbers, or blame CCBot for load, confirm the request is genuine.

Common Crawl is aware that other crawlers falsely identify themselves as CCBot. The real CCBot runs from dedicated IP ranges with reverse DNS, so a logged request can be verified with a forward-confirmed reverse DNS check. Real CCBot resolves to a `.crawl.commoncrawl.org` hostname that resolves back to the same IP. An impostor will not.

```
# Verify a logged IP claiming to be CCBot (forward-confirmed reverse DNS)
host 18.97.14.84
# real CCBot → 18-97-14-84.crawl.commoncrawl.org

host 18-97-14-84.crawl.commoncrawl.org
# resolves back to the same IP → 18.97.14.84  (verified)

# The published IP ranges are also available as JSON:
curl -s https://index.commoncrawl.org/ccbot.json | head
```

Spotting an impostor in your logs

The pattern is easy to see once you know it. Genuine CCBot fetches `robots.txt` first and comes from a verifiable `.crawl.commoncrawl.org` host. A request that carries the CCBot user agent but originates from an unrelated IP that fails the reverse-DNS check is an impostor borrowing the name.

```
# Genuine CCBot: verifiable IP, fetches robots.txt, identifies honestly
3.41.188.32   "GET /robots.txt HTTP/1.1" 200  "CCBot/2.0 (https://commoncrawl.org/faq/)"
3.41.188.32   "GET / HTTP/1.1"           200  "CCBot/2.0 (https://commoncrawl.org/faq/)"

# Impostor: same UA string, but an IP that fails reverse-DNS verification
49.200.103.146  "GET /... HTTP/1.1"     404  "CCBot/2.0 (https://commoncrawl.org/faq/)"
← unverified IP
```

WHY THIS MATTERS FOR THE AUDIT

Impersonation distorts the picture in both directions. It can inflate apparent CCBot traffic so a client over-blocks in response, or it can get the real CCBot wrongly blamed and banned at the WAF. Verify by IP, not by user agent, before acting on any crawler claim.

Sources: *Common Crawl CCBot verification guidance (commoncrawl.org/ccbot): dedicated IP ranges with reverse DNS, forward-confirmed reverse DNS check, and published ranges at index.commoncrawl.org/ccbot.json. Example IP shown for the method is from CCF's own documentation; log lines are illustrative of the real-versus-impostor pattern.*

THE FIVE CHECKS · 3, 4 & 5 OF 5

Priority, structure & rendering.

Why you're crawled at the rate you are, and whether what's crawled is actually usable.

3 Harmonic Centrality check

Look up the domain's centrality and rank in the Common Crawl Web Graph. A low rank means the domain is deprioritised in crawl budget, so even with open access it gets crawled shallowly and infrequently. Flag low centrality as a strategic risk and a link-building target aimed at core-connected sites.

`webgraph.metehan.ai`

4 Structured data completeness

Entities without structured data are harder to represent cleanly in training and harder for models to attribute. Audit Schema.org markup on key pages: organisation, article/product, author, breadcrumb. Missing or invalid markup is a low-effort, high-leverage fix.

`Google Rich Results Test`

5 Server-side rendering audit

Many AI crawlers behave like early Googlebot: they fetch HTML but may not execute JavaScript. If the content only exists after JS runs, the crawler may capture an empty shell. Validate by comparing the raw fetch against the rendered page.

`curl + fetch compare`

```
# Does the real content exist in the raw HTML, before any JS runs?
curl -s https://example.com/key-page | grep -i "headline text"

"match" = content is server-rendered and crawler-visible.
"no match" = content is JS-injected; AI crawlers may see an empty page.
```

THE DELIVERABLE

Package the five results into a one-page scorecard per domain: pass/fail on access and rendering, present/absent in the index, a centrality grade, and a structured-data gap list, each with a specific remediation. That scorecard is the billable artifact.

SECTION 04 · THE TOOLKIT

The toolkit.

Free tools that power every step of the audit. Two anchors, four supporting.

OFFICIAL CC TOOL

index.commoncrawl.org

The Common Crawl Index Server. Every crawl, every captured URL, searchable. This is how you answer "are they in the archive, and when were they last seen?"

COMMUNITY TOOL

webgraph.metehan.ai

A CC Rank Checker built by Metehan Yesilyurt on top of Common Crawl Web Graph data, the fastest way to read a domain's Harmonic Centrality and crawl priority today. CCF has its own centrality tooling on the way; until then, this is the quickest option.

Supporting tools

- **Screaming Frog**: crawl as a custom user agent (e.g. CCBot) to surface UA-specific blocks and rendering gaps at scale.
- **curl**: the quickest way to test edge/WAF behaviour and raw-HTML rendering, one command at a time.
- **Google Rich Results Test**: validate Schema.org structured data on key pages.
- **Your CDN dashboard** (AI-crawler or bot-management settings): where most accidental blocks are actually fixed.

COST REMINDER

The Common Crawl data and index are free and need no authentication. At scale, processing full WARC archives on cloud infrastructure can get expensive, but the audit checks in this guide query the lightweight index and live pages, so they stay free.

Note: *webgraph.metehan.ai is an independent community project, not an official Common Crawl Foundation product, and CCF plans to release its own centrality tool. The Common Crawl data, index, and Web Graph rank data are all free and need no authentication. Verify any single data point against the official index or rank files where a decision hinges on it.*

REMEMBER

If you're not in the crawl, you're not in the model.

And if you're not in the model, you may not be in the market. The old world was index and rank. The new world is train and retrieve, and the work starts one layer earlier than most SEO has ever reached.

THE CHECKLIST

- Run the CCBot access check on your top client domains. Test robots.txt *and* the WAF.
- Confirm presence in the CC Index, and note the last crawl date.
- Pull a centrality read and flag any domain that's deprioritised.
- Spot-check structured data and server-side rendering on key pages.
- Fix the easy wins in the CDN dashboard first; they're the highest-leverage.

Resources

- commoncrawl.org
- commoncrawl.org/web-graphs
- commoncrawl.org/ccbot
- index.commoncrawl.org
- webgraph.metehan.ai (CC Rank Checker)

“

When we include more communities, more cultures, and more languages in the crawl, we are not just improving training data. We are making sure AI reflects all of humanity. Not just a slice of it.

STEPHEN BURNS · COMMON CRAWL FOUNDATION