Scott Tischler

FOUNDER & CHAIRMAN · AIRECOMMEND.AI

TECHNICAL

Do Core Web Vitals and Page Speed Matter for AI Citations?

By Scott Tischler · October 11, 2026 · 8 min read

Mostly no — raw page speed and your Core Web Vitals scores are not a meaningful lever for getting cited by ChatGPT, Perplexity, Gemini, or Google's AI answers. The thing people confuse with speed, though, absolutely matters: whether a crawler can fetch your page and read your actual content without needing a browser to run a pile of JavaScript first. That is a rendering and accessibility question, not a "shave 300 milliseconds off your load time" question — and it is where I would spend the effort.

This comes up constantly because so many of us carry SEO muscle memory. We spent a decade being told Core Web Vitals were a ranking factor, we obsessed over Largest Contentful Paint and Cumulative Layout Shift, and now the instinct is to assume the same checklist wins AI visibility. It does not map cleanly. Let me explain where the SEO reflex helps and where it quietly wastes your time.

What Core Web Vitals measure — and who they're for

Core Web Vitals are a set of human-experience metrics: how fast the main content paints, how responsive the page is to input, how much the layout jumps around while loading. They exist to describe what a *person* feels when a page loads in a *browser*. That framing is the key. They are about the rendered, interactive experience on a device held by a human.

An AI system fetching your page to extract facts is not that human. In most cases it is a crawler or a fetch process pulling your HTML to read the text. It does not care whether your hero image caused a layout shift or whether a button was tappable within 200 milliseconds. It cares whether the content it needs is present and legible in what it retrieved. A page can score terribly on Core Web Vitals and still be perfectly readable to a machine, and a page can score beautifully and still hide its real content behind scripts the fetcher never runs.

So when someone tells me they are going to "fix their AI visibility" by optimizing Web Vitals, my honest answer is that they are optimizing the right page for the wrong reader.

The thing that actually bites: JavaScript-rendered content

Here is the failure I see repeatedly, and it is the one worth your attention. A modern site is built so that the HTML a fetcher receives is nearly empty — a skeleton — and the real content gets injected by JavaScript after the page loads in a browser. Humans never notice, because their browser runs the script in a fraction of a second. But many AI crawlers do not execute JavaScript, or do not execute it reliably, or do not wait around for it. What they retrieve is the skeleton. Your carefully written answer, your pricing, your FAQ, your proof — none of it is in the document they actually read.

This is the real "speed-adjacent" issue, and it masquerades as a performance problem because both live in the land of front-end engineering. But the fix is not making the page faster. The fix is making sure your meaningful content is in the server-rendered HTML — present in the raw document before any script runs. Server-side rendering, static generation, or at minimum ensuring your primary content and key facts are in the initial markup. If a model cannot see it without a browser, assume it cannot see it.

The single best test costs you nothing: fetch your own page the way a simple client would and look at what comes back. View the raw source, or pull it with a basic request, and ask whether your actual answer is sitting there in the text. If you only see a loading shell and a bundle of scripts, that is your AI-visibility problem, and no Web Vitals score will fix it.

I want to be concrete about how backwards this gets, because it trips up good engineering teams. A site can be a technical showpiece — instant perceived load, buttery interactions, a perfect green performance report — and still be nearly invisible to a machine, precisely *because* all that polish was achieved by pushing rendering into the browser. The same architecture that makes the page feel fast to a human is what empties out the document a crawler receives. Meanwhile a plain, server-rendered page that a performance purist would sniff at can be trivially readable by every engine, because its content is just sitting there in the HTML. Performance and retrievability are not the same axis, and optimizing hard on one can quietly cost you the other. That is the trap.

Where speed does matter — just not the way you think

I do not want to oversell "speed never matters," because there is a real edge case. Crawlers operate on budgets. If your pages are so slow to respond that fetches time out, or your server is flaky and intermittently fails, a crawler can give up before it gets your content, and it may crawl you less often over time. That is not a Core Web Vitals problem in the LCP-and-CLS sense. It is a server-response-time and reliability problem — time to first byte, uptime, not choking under crawl load.

So the speed that matters for machines is "does the server hand over the document promptly and reliably," which is a much lower and blunter bar than the polished human-experience metrics. A page

can be slow to become *interactive* for a human while still returning its HTML to a crawler in a snap. Those are different clocks. Optimize the crawler clock — fast, reliable server responses — and you have handled the part of speed that touches AI visibility. The rest of the Web Vitals checklist is a user-experience and conventional-SEO investment, which is fine, just do not file it under AEO.

Crawlability is the real checklist

If raw speed is mostly a distraction, what is the actual technical checklist for being readable by AI? It is a crawlability-and-accessibility list, and it looks like this.

- **Content in the server-rendered HTML.** The non-negotiable. Your real answer must be in the raw document, not injected later by client-side scripts.
- **Crawlers allowed.** Check that your robots rules and any bot-management or firewall layer are not quietly blocking the AI user-agents you want to reach you. It is startlingly common to block the very crawlers you are trying to attract.
- **Clean, semantic structure.** Real headings, lists, and paragraphs. Content organized so a machine can tell what answers what. This is the extraction-friendly formatting that makes a passage easy to lift and quote.
- **Reliable server responses.** Promptly returned HTML, consistent uptime, no intermittent failures that make a crawler back off.
- **No content locked behind interaction.** Anything that only appears after a click, a tab switch, or a scroll-triggered load may be invisible to a fetcher. If it matters, make it present by default.

Notice that not one item on that list is a Core Web Vitals metric. That is the point. The technical work that moves AI visibility is about *access to your content*, not the subjective smoothness of the experience.

The crawler-blocking item deserves a second look, because it is the one that makes people wince when they find it. Plenty of sites sit behind a security or bot-management layer that was configured, reasonably, to keep scrapers and bad traffic out. The problem is that these layers often treat AI crawlers as exactly the kind of automated traffic they were built to stop — so the very engines you are trying to earn citations from get served a challenge page or an outright block instead of your content. You can do everything else right and still be invisible because a firewall rule from two years ago is quietly turning the model away at the door. If you have a bot-management vendor or a web application firewall in front of your site, make checking its treatment of AI user-agents an explicit item, not an assumption. I have seen this single misconfiguration undo an otherwise excellent setup.

My honest take on where to spend the hour

If you have one hour of engineering time to improve how machines read your site, do not spend it chasing a green Web Vitals score. Spend it confirming your content is in the raw HTML, that crawlers are allowed in, and that your server responds reliably. Those three things are the whole ballgame for retrievability, and most sites fail at least one of them without realizing it.

Keep caring about Core Web Vitals for the reasons they were invented — real humans do bounce from slow, janky pages, and conventional search still weighs experience. A good experience is good business. Just be clear-eyed that it is a human-experience and traditional-SEO investment, not an AI-citation one. My read, and I will hold this loosely because these systems evolve: the engines want your content legible and fetchable far more than they want it fast. Make yourself easy to read first. Make yourself fast for humans second. Do not confuse the two, and do not let a performance dashboard convince you that you have done your AEO homework when the real work is one layer underneath.

Key takeaways

- Core Web Vitals measure the human, in-browser experience — a reader that AI crawlers mostly aren't. Optimizing them for AI visibility is the right page for the wrong reader.
- The real issue that hides behind 'speed': JavaScript-rendered content. Many AI crawlers don't run your scripts, so if your real content isn't in the server-rendered HTML, they never see it.
- Free test: fetch your own page as a simple client and view the raw source. If you only see a loading shell and scripts, that's your AI-visibility problem — no Web Vitals score fixes it.
- The speed that does matter for machines is server response time and reliability — promptly returned HTML and consistent uptime, so crawlers don't time out and back off. That's a much blunter bar than LCP/CLS.
- The real technical checklist is crawlability and access: content in the raw HTML, crawlers allowed through robots and firewall layers, clean semantic structure, reliable responses, nothing locked behind interaction.
- Keep caring about Core Web Vitals for humans and conventional SEO — just don't file it under AEO. Make yourself easy to read first, fast for humans second.

Frequently asked questions

If my site scores poorly on Core Web Vitals, will AI engines refuse to cite me?

Not on the basis of the scores themselves. A page can score badly on Largest Contentful Paint or layout shift and still be perfectly readable to a crawler, because those metrics describe a human's in-browser experience, not whether the HTML is fetchable. The thing that actually blocks citation is your content not being present in the server-rendered HTML, or crawlers being blocked — fix those, not your Web Vitals grade.

How do I know if my content is invisible to AI crawlers?

Fetch your own page the way a simple client would — view the raw page source or pull it with a basic request — and check whether your actual answer, pricing, or FAQ text is sitting there in the document. If you only see a loading skeleton and a bundle of scripts, your content is being injected by JavaScript that many crawlers won't run, and it's effectively invisible to them.

So should I stop optimizing page speed entirely?

No — just redirect it. Keep server response time fast and reliable, because crawlers operate on budgets and will back off from pages that time out or fail intermittently; that's the speed dimension that touches AI visibility. Keep Core Web Vitals healthy for real human visitors and conventional SEO too. Just don't mistake a green performance dashboard for AEO work — the AI-citation work is one layer underneath, in retrievability.

About the author. Scott Tischler is the Founder & Chairman of AIrecommend.ai and a practitioner-authority on AI search and Answer Engine Optimization. With 20+ years in marketing technology — including American Express, MetLife, and UBS — and executive and professional study at Wharton, Harvard, and Oxford, he helps businesses become the ones AI recommends.

© 2026 Scott Tischler · scotttischler.com scott@airecommend.ai · AIrecommend.ai