

[All Resources](#) / [Insights](#)

The Case Against llms.txt: Why the Hype Outpaces the Reality

A critical look at llms.txt, a proposed standard for AI visibility that's high-effort, low-impact, and not widely adopted.

Jun 5 • daydream team



The Case Against llms.txt: Why the Hype Outpaces the Reality

daydream

At daydream, we've taken a closer look at the proposed llms.txt [standard](#), a well-intentioned attempt to make website content more accessible to large language models (LLMs). We've previously explored what the standard aims to do, why it was created, and how it fits into the broader push for AI visibility.

While the concept is sound in theory, limited adoption and lack of support from major platforms make it hard to see llms.txt becoming a widely accepted standard anytime soon.

What is llms.txt trying to do?

The llms.txt standard, originally proposed by [Jeremy Howard](#) of Answer.AI, outlines a structured way for websites to surface LLM-readable content. It introduces a few key components:

- **The llms.txt file:** A Markdown file hosted at `/llms.txt` that serves as an index of links to high-value content

on your site, like API docs, onboarding flows, support articles, etc.

- **.md page mirrors:** Each link in llms.txt is expected to have a Markdown version at the same path (e.g. `/docs/start → /docs/start.md`) to provide a clean, token-efficient version of the content.
- **llms-full.txt (optional):** A concatenated file bundling all the .md content into one large markdown file, often used to simulate a context window for models.

The idea is to reduce friction for LLMs that struggle to parse bloated HTML, JavaScript-heavy interfaces, or deeply nested page structures. It's positioned as a lightweight, structured alternative to HTML crawling.

Despite the clean theory, the standard lacks traction and carries serious tradeoffs.

Problem #1: The Maintenance Overhead is Real

Let's start with the most immediate issue: llms.txt is high-maintenance.

To adopt the standard fully, a team must:

- Curate which pages are "LLM-relevant"
- Write and maintain short descriptions
- Create and host Markdown mirrors for each page
- Optionally generate and update a llms-full.txt bundle
- Sync all changes with the primary HTML content

This introduces ongoing operational complexity, especially without strong automation support, which isn't readily available across platforms or documentation tools.

Unlike sitemap.xml, which can be generated automatically and adheres to a well-established format, llms.txt requires substantial manual oversight. There's no authoritative schema or tooling to define what makes a page "LLM-worthy," how descriptions should be structured, or how frequently the file should be updated. As a result, implementation is both ambiguous and brittle.

If the markdown files fall out of sync with your primary web content, LLMs may ingest outdated or misleading data. This can lead to hallucinated outputs or citation mismatches. Worse, if those raw markdown files are indexed or referenced in search, users could be directed to broken or stripped-down pages that lack full context.

This burden hits smaller teams hardest. While large orgs can afford dedicated pipelines for mirrored documentation, most startups and mid-sized companies can't justify the cost, especially when there's no proven benefit.

Problem #2: It Solves a Moment-in-Time Problem

The llms.txt standard is a workaround for a set of limitations that may soon become irrelevant.

The core pain it tries to address is that HTML is too noisy for LLMs: navbars, scripts, tooltips, and layout bloat inflate token counts without adding context. That's fair, but model architectures are evolving fast.

We're already seeing:

- Longer context windows (e.g., Gemini 1.5 with 1M+)
- Improved vision capabilities
- Semantic chunking and contextual prioritization
- Better HTML parsing and document preprocessing

In other words, the models are catching up.

Soon, LLMs will likely be able to parse sites more like human users do, prioritizing meaning over markup. At that point, maintaining a sidecar Markdown spec becomes an unnecessary detour.

The llms.txt format, then, feels like a **bridge to nowhere**: a short-term fix with long-term upkeep.

Problem #3: Redundant with Existing Standards

The llms.txt file functions similarly to a filtered sitemap, but with some notable differences. It includes human-written descriptions of each linked page—unlike sitemaps, which rely on structured metadata or meta descriptions pulled directly from page HTML.

Still, the overlap is significant. Sitemaps already convey:

- Page hierarchy
- Update frequency
- Crawl priority
- Canonical discovery

The claim that LLMs “don’t like HTML” is also misleading. It’s not that models dislike HTML—it’s that web pages contain a lot of extraneous information that gets in the way of clean ingestion. Preprocessing HTML more effectively allows models to extract meaning without needing a markdown fallback.

llms.txt assumes manual curation, redundant .md hosting, and new infrastructure. Meanwhile, existing formats like robots.txt, sitemap.xml, and schema.org metadata are already supported, standardized, and baked into most modern web tooling.

If anything, llms.txt competes with those signals without offering any clearer path to visibility.

Problem #4: No One Is Using It

This is the most fundamental issue.

Despite a few dev-centric companies adopting the standard—Cloudflare, Mintlify, Hugging Face—no major LLM provider has confirmed support. Not OpenAI, not Google, not Anthropic, not Meta.

In fact, [Google’s John Mueller](#) addressed this directly on Reddit (from his personal account):

“AFAIK, none of the AI services have said they’re using llms.txt (and you can tell when you look at your server logs that they don’t even check for it). To me, it’s comparable to the keywords meta tag — this is what a site-owner claims their site is about.”

If crawlers aren’t fetching the file, it’s doing nothing. No indexing. No retrieval. No influence on ranking or LLM citations.

Even the optional .md pages and llms-full.txt files are unconfirmed as ingestion sources.

That makes this standard a speculative effort. You're investing engineering resources into a system that no known LLM respects.

Here's Our Take

We're generally in favor of content standards that improve visibility, structure, and performance, especially for AI-native applications.

llms.txt isn't that.

It's:

- Not widely adopted by LLM providers
- Redundant with sitemap.xml and structured metadata
- High effort to implement and maintain
- Susceptible to version drift and stale documentation
- Solving a problem that may soon be irrelevant

If you want to experiment with it for internal testing or niche use cases, there's no harm. For most websites, we don't see a compelling reason to adopt it.

The best way to prepare your site for AI discovery is still:

1. Clean, semantic HTML
2. Comprehensive structured data (schema.org, JSON-LD)
3. Server-side rendering
4. Open access (robots.txt, Google-Extended, GPTBot directives)
5. Content that's built to answer real user questions, not just optimize for machines

You don't need a sidecar file full of markdown to be AI-friendly. You need clarity, structure, and intent baked into your primary content itself. Let's build toward that, rather than betting on standards no one uses.

The future of search is up
don't get left behind

Gain actionable insights in real-time as we build and apply th

[INSIGHTS](#) May 15

Make AI Engines Trust (and Cite) Your Content

[INSIGHTS](#) May 15

Can't You Just Do It All?

[INSIGHTS](#) May 8

How OpenAI Crawls Your Website

[Explore the daydream library](#)

Build an organic growth engine that drives results

[Book a demo](#)

NOTES ON AI, GROWTH, AND THE JOURNEY FROM 0 TO 1000+

2026 © daydream Labs, Inc.

[View in browser](#)

[Subscribe](#)