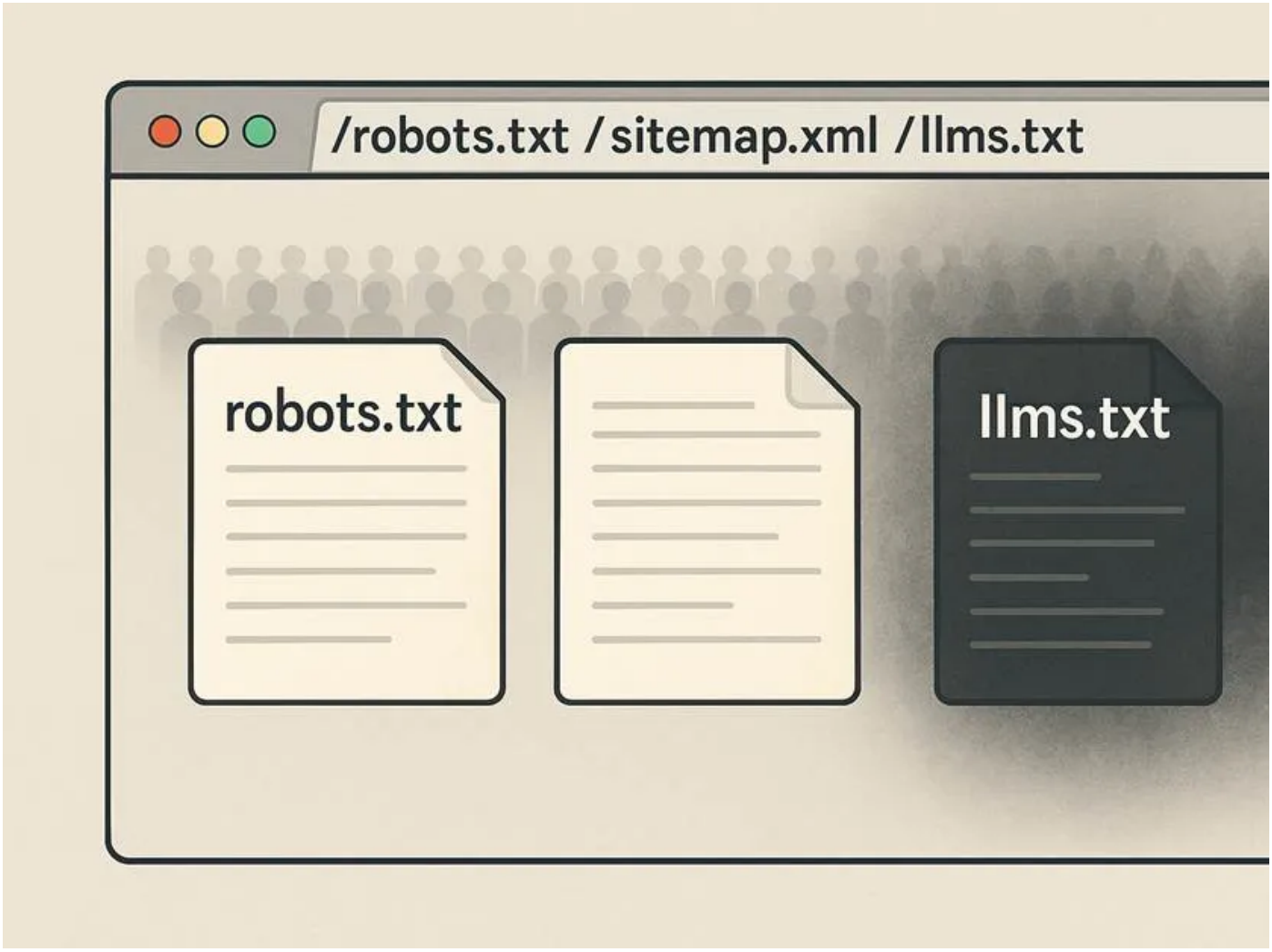
llms.txt: The Web's Next Great Idea, or Its Ne Spam Magnet

A look at the promise, the risk, and the platform hesitation.

DUANE FORRESTER DECODES
NOV 09, 2025



At a recent conference I was asked if llms.txt mattered. I'm personally not a fa we'll get into why below. I listened to a friend who told me I needed to learn m about it as she believed I didn't fully understand the proposal, and I have to ad she was right. After doing a deep dive on it, I now understand it much better. Unfortunately, that only served to crystallize my initial misgivings. And while sound like a single person disliking an idea, I'm actually trying to view this fro perspective of the search engine or the ai-platform. Why would they, or why w they adopt this protocol? And that POV lead me to some, I think, interesting i

We all know that search is not the only discovery layer anymore. Large-langua model (LLM)-driven tools are rewriting how web content is found, consumed, represented. The proposed protocol called llms.txt attempts to help websites g those tools. But the idea carries the same trust challenges that killed earlier "h machine understand me" signals. This article explores what llms.txt is meant t I understand it), why platforms would be reluctant, how it can be abused, and must change before it becomes meaningful.

What llms.txt hoped to fix

Modern websites are built for human browsers: heavy JavaScript, complex nav interstitials, ads, dynamic templates. But most LLMs, especially at inference t operate in constrained environments: limited context windows, single-pass do reads, and simpler retrieval than traditional search indexers. The original prop from Answer.AI suggests adding an *llms.txt* markdown file at the root of a site, lists the most important pages, optionally with flattened content so AI system have to scramble through noise.

Supporters describe the file as "a hand-crafted sitemap for AI tools" rather tha crawl-block file. In short, the theory: give your site's most valuable content in cleaner, more accessible format so tools don't skip it or misinterpret it.

The trust problem that never dies

If you step back, you discover this is a familiar pattern. Early in the web's hist something like the meta keywords tag let a site declare what it was about; it w widely abused and ultimately ignored. Similarly, authorship markup (rel=auth tried to help machines understand authority, and again, manipulation followe Structured data (schema.org) succeeded only after years of governance and sha adoption across search engines. llms.txt sits squarely inside this lineage: a self declared signal that promises clarity but trusts the publisher to tell the truth. verification, every little root-file standard becomes a vector for manipulation.

The abuse playbook (what spam teams see immediately)

What concerns platform policy teams is plain: if a website publishes a file call llms.txt and claims whatever it likes, how does the platform know that what's matches the live content users see, or can be trusted in any way? Several explo open up:

1. Cloaking through the manifest. A site lists pages in the file that are hidde regular visitors or behind paywalls, then the AI tool ingests content nobo sees.
2. Keyword stuffing or link dumping. The file becomes a directory stuffed w affiliate links, low-value pages, or keyword-heavy anchors aimed at gamin retrieval.
3. Poisoning or biasing content. If agents trust manifest entries more than t of messy HTML, a malicious actor can place manipulative instructions or lists that affect downstream results.
4. Third-party link chains. The file could point to off-domain URLs, redirec or content islands, making your site a conduit or amplifier for low-quality
5. Trust laundering. The presence of a manifest might lead an LLM to assig

> weight to listed URLs, so a thin or spammy page gets a boost purely by appearance of structure.

The broader commentary flags this risk. For instance, some industry observers that llms.txt "creates opportunities for abuse, such as cloaking." And commun feedback apparently confirms minimal actual uptake: "No LLM reads them." absence of usage ironically means fewer real-world case studies of abuse, but i means fewer safety mechanisms have been tested.

Why platforms hesitate

From a platform's viewpoint the calculus is pragmatic: new signals add cost, r enforcement burden. Here's how the logic works.

First, **signal quality**. If llms.txt entries are noisy, spammy or inconsistent with site, then trusting them can reduce rather than raise content quality. Platform ask: will this file improve our model's answer accuracy or create risk of misinformation or manipulation?

Second, **verification cost**. To trust a manifest you need to cross-check it again live HTML, canonical tags, structured data, site logs, etc. That takes resources Without verification, a manifest is just another list that might lie.

Third, **abuse handling**. If a bad actor publishes an llms.txt manifest that lists misleading URLs which an LLM ingests, who handles the fallout? The site ow The AI platform? The model provider? That liability issue is real.

Fourth, **user-harm risk**. An LLM citing content from a manifest might produc inaccurate or biased answers. This just adds to the current problem we alread with inaccurate answers and people following incorrect, wrong or dangerous a

Google has already stated that it will *not* rely on llms.txt for its "AI Overviews'

and continues to follow "normal SEO." And John Mueller wrote: "FWIW no A currently uses llms.txt." So the tools that could use the manifest are largely sta the side-lines. This reflects the idea that a root-file standard without establish is a liability.

Why adoption without governance fails

Every successful web standard has shared DNA: a governing body, a clear voca and an enforcement pathway. The standards that survive all answer one questi early…"Who owns the rules?"

Schema.org worked because that answer was clear. It began as a coalition betv Bing, Google, Yahoo, and Yandex. The collaboration defined a bounded vocab agreed syntax, and a feedback loop with publishers. When abuse emerged (fak reviews, fake product data) those engines coordinated enforcement and refine documentation. The signal endured because it wasn't owned by a single comp left to self-police.

Robots.txt, in contrast, survived by being minimal. It didn't try to describe cor quality or semantics. It only told crawlers what *not* to touch. That simplicity re its surface area for abuse. It required almost no trust between webmasters and platforms. The worst that could happen was over-blocking your own content; was no incentive to lie inside the file.

llms.txt lives in the opposite world. It invites publishers to self-declare what m most and, in its full-text variant, what the "truth" of that content is. There's nc consortium overseeing the format, no standardized schema to validate against enforcement group to vet misuse. Anyone can publish one. Nobody has to resp And no major LLM provider today is known to consume it in production. May are, privately, but publicly, no announcements about adoption.

What would need to change for trust to build

To shift from optional neat-idea to actual trusted signal, several conditions mu
met, and each of these entails a cost in either dollars or human time, so again,

- First, **manifest verification**. A signature or DNS-based verification could
  llms.txt file to site ownership, reducing spoof risk. (cost to website)
- Second, **cross-checking**. Platforms should validate that URLs listed corre
  live, public pages, and identify mismatch or cloaking via automated check
  to engine/platform)
- Third, **transparency and logging**. Public registries of manifests and logs c
  updates would make dramatic changes visible and allow community audit
  to someone)
- Fourth, **measurement of benefit**. Platforms need empirical evidence that
  ingesting llms.txt leads to meaningful improvements in answer correctnes
  citation accuracy or brand representation. Until then this is speculative. (c
  engine/platform)
- Finally, **abuse deterrence**. Mechanisms must be built to detect and penali
  spammy or manipulative manifest usage. Without that, spam teams simpl
  negative benefit. (cost to engine/platform)

Until those elements are in place, platforms will treat llms.txt as optional at be
irrelevant at worst. So maybe you get a small benefit? Or maybe not…

The real value today

For site owners, llms.txt still may have some value, but not as a guaranteed pat
traffic or "AI ranking." It can function as a content alignment tool, guiding int
teams to identify priority URLs you want AI systems to see. For documentatio
sites, internal agent systems, or partner tools that you control, it may make se

publish a manifest and experiment.

However, if your goal is to influence large public LLM-powered results (such a by Google, OpenAI or Perplexity), you should tread cautiously. There is no pul evidence those systems honor llms.txt yet. In other words: treat llms.txt as a "r of your content strategy, not a "magnet" pulling traffic. Of course, this means the file(s) and maintaining them, so factor in the added work v. whatever retur believe you will receive.

Closing Thoughts

The web keeps trying to teach machines about itself. Each generation invents format, a new way to declare "here's what matters." And each time the same q decides its fate: "Can this signal be trusted?" With llms.txt the idea is sound, b trust mechanisms aren't yet baked in. Until verification, governance and empi proof arrive, llms.txt will reside in the grey zone between promise and problen

Discussion about this post

Write a comment...

ANDREEA LEONTE Nov 17

Liked by Duane Forrester Decodes

Thanks for writing this, it clarifys a lot. Your insights on AI topics are consistently spot c

LIKE (1) REPLY

1 reply by Duane Forrester Decodes

Steve Wiideman Nov 9

Liked by Duane Forrester Decodes

I couldn't agree more about the importance of adoption and governance with llms.txt.

That being said, from a would've/could've/should've perspective, we haven't seen any said manifesto, provided the document is simplified, void of promotional language, and our knowledge graph efforts.

If they use it eventually, great! If not, we lost 15 minutes and didn't create anything that remotely spammy.

LIKE (1) REPLY

2 replies by Duane Forrester Decodes and others

3 more comments...