Journal of Information Systems Engineering and Management
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Restoring Machine-Readability in the Age of AI-Synthesized Answers

Bhanu Phanindra Babu Gogula
University of Central Missouri, USA

ARTICLE INFO

Received: 20 Feb 2026

Accepted: 22 Feb 2026

ABSTRACT

AI-based information retrieval systems have transformed content interaction patterns. Users increasingly expect answers in the form of synthesized responses, rather than interface navigation. Modern web architectures often rely on client-side rendering, which typically serves content by running JavaScript code and is difficult for automated agents to crawl, as JavaScript-infused pages appear as empty HTML shells to conventional web crawlers. This limits agent access to material until execution occurs. This poses concerns for technical documentation, educational materials, and knowledge repositories where complete and verifiable sources cannot be provided and alternative sources or incomplete content must be used. Dual-delivery architectures recognize requesting agent capabilities to deliver pre-rendered static HTML to automated browsers while delivering interactive content to human visitors. Three general categories exist for implementing dual-delivery architectures: network edge architectures, server-side rendered architectures, and hybrid architectures that deliver content using a single codebase in an accessible manner. True success relies on content equivalence between machine and human views, improved semantic markup and structured metadata vocabularies, and measurement frameworks through user-agent analysis and accessibility testing. The verifiable engineering practice of machine-readability and machine-visibility improvements can become the new standard of successful creative practice.

Keywords: Client-Side Rendering, Machine Accessibility, Dual-Delivery Architecture, Semantic Markup, Generative Engine Optimization

1. Introduction

These platforms have transformed information retrieval from ranked search engine results pages into synthesized results containing relevant information from a variety of sources in a single, coherent response. Studies of human information behavior in these new AI-augmented information environments have shown that conversational user interfaces have shifted user expectations away from linking to curated, direct, and coherent answers. Studies found increases in the share of users interacting with direct answers across different user groups [1]. One of the challenges of addressing this trend is architectural: much of the current web is built using client-side rendering technology, in which content is delivered by executing JavaScript code in the browser. Broadly, an analysis of modern web development best practices shows that single-page application frameworks and JavaScript rendering patterns have become dominant architectural models, and evidence suggests that a high fraction of modern Internet content requires running JavaScript client-side in order to deliver core content [2]. While this results in improved interactive experiences for human users, it creates an important visibility gap for AI retrieval systems, which often do not execute the whole rendering process. The result is a growing body of authoritative documentation, knowledge bases, and other technical content that sits, invisible, within the systems we use to find answers. This problem is not simply one of discoverability. Synthesized answers are less useful if there are no original sources to find and verify. Such answers may cite no sources, depend on secondary interpretations, and hallucinate information. In addition, content authors may miss out on receiving credit for their original work. Consumers receive an answer lacking original context, authority, and specificity. The difference between human-readable content served to browsers and automated agents that retrieve

Journal of Information Systems Engineering and Management
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

web pages has created a two-tiered web where access to information is contingent upon technical capability.

2. The Client-Side Rendering Visibility Gap

Modern web applications, especially single-page applications, often use frameworks that dynamically render user interface elements in the user's web browser. In these applications, the first server reply typically consists of a small amount of HTML such as a loading container and a script. Additionally, meaningful content is presented only after JavaScript has executed and the document structure has been fully generated and parsed. Multiple studies on web performance characteristics have reported that JavaScript execution is a major bottleneck in the sequence of rendering the web page, where the blocking time on the main thread can be several seconds, regardless of network or device capabilities [3].

In longitudinal studies on the growing importance of JavaScript on the web, the adoption patterns demonstrate significant escalation over time. In 2001, among 10,000 examined websites, 8,256 had no data available, 1,317 maintained the same JavaScript inclusions, and 427 introduced new JavaScript inclusions, representing 24.48% of websites with new inclusions. By 2002, the distribution shifted to 7,952 websites with no data, 1,397 with same inclusions, and 651 with new inclusions, increasing to 31.79%. The trend continued in 2003 with 7,576 websites showing no data, 1,687 maintaining the same inclusions, and 737 adding new inclusions at 30.4%. In 2004, 7,100 websites had no data, 2,037 had the same inclusions, and 863 introduced new inclusions, representing 29.76%. The year 2005 showed 6,672 websites with no data, 2,367 with same inclusions, and 961 with new inclusions at 28.88%. By 2006, 6,073 websites had no data, 2,679 maintained the same inclusions, and 1,248 added new inclusions, reaching 31.78%. In 2007, 5,074 websites showed no data, 3,136 had the same inclusions, and 1,790 introduced new inclusions at 36.34%. The acceleration continued in 2008 with 3,977 websites having no data, 3,491 with same inclusions, and 2,532 with new inclusions, representing 42.04%. By 2009, only 3,111 websites had no data, 3,855 maintained the same inclusions, and 3,034 added new inclusions at 44.04%. Finally, in 2010, the pattern reached 1,920 websites with no data, 4,407 with same inclusions, and 3,673 with new inclusions, representing 45.46% of all websites—an increase of 86% from 2001 [2]. Analysis of the ten most popular remotely-included JavaScript files reveals the dominance of specific services across the top 10,000 ranked websites. Web analytics scripts demonstrate the highest prevalence, with the most popular analytics file included in 68.37% of tracked websites. Dynamic advertising scripts appear in 23.87% of sites, while an older version of web analytics scripts is included in 17.32% of sites. Social networking integration files show significant adoption, with primary social networking platforms present in 16.82% and 13.87% of websites respectively. Combined social networking and analytics services are included in 12.68% of sites. Additional web analytics and tracking scripts appear in 11.98% of websites, while market research beacons are included in 10.45%. Helper function libraries appear in 10.14% of sites, and secure analytics implementations are included in 10.12% of tracked websites [2]. This execution-dependent model means that the text content, logical structure and metadata required by information extraction are not available until the JavaScript code in the page is tracked and yielded in the client environment.

Journal of Information Systems Engineering and Management
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

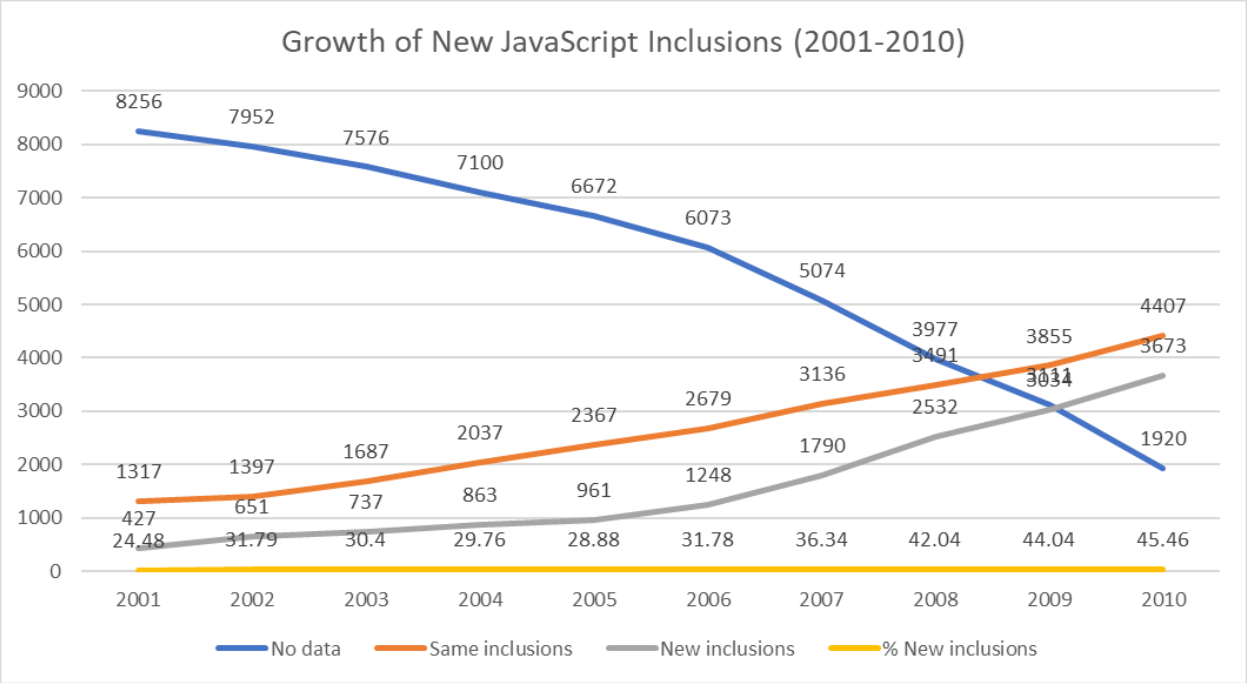

Figure 1: Growth of New JavaScript Inclusions from 2001 to 2010 [2]

Year	No data	Same inclusions	New inclusions	% New inclusions
2001	8256	1317	427	24.48
2002	7952	1397	651	31.79
2003	7576	1687	737	30.4
2004	7100	2037	863	29.76
2005	6672	2367	961	28.88
2006	6073	2679	1248	31.78
2007	5074	3136	1790	36.34
2008	3977	3491	2532	42.04
2009	3111	3855	3034	44.04
2010	1920	4407	3673	45.46

Table 1: Growth of New JavaScript Inclusions (2001-2010) Figure [2]

Journal of Information Systems Engineering and Management
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

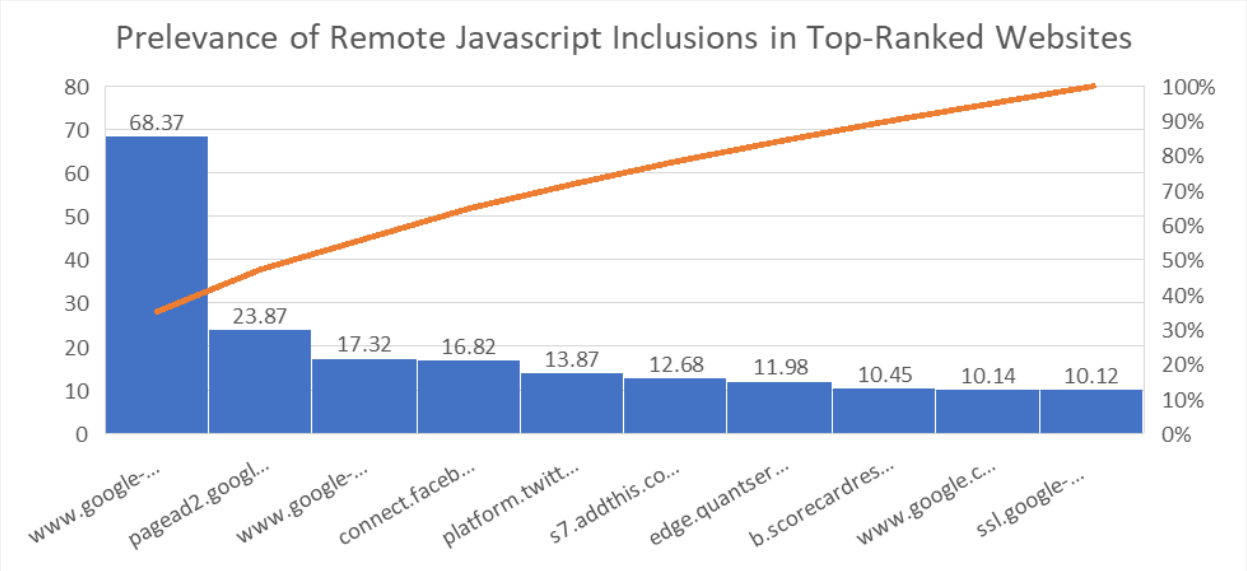

Figure 2: Prevalence of Remote JavaScript Inclusions in Top-Ranked Websites [2]

While this architecture works well for human users using a web browser that can execute the contained JavaScript, it does not work for machine agents that only download the same pages but do not execute them. The huge majority of web reading systems, search engine crawlers, and AI systems are essentially lightweight HTTP clients that parse HTML without executing JavaScript. When crawlers (or other bot software) reach client-side rendered pages, the bots only receive the initial HTML shell, with little to no content, with text, headings, structured data and semantic HTML never coming through. Empirical research comparing web measurement technologies has shown that Resource Timing APIs and HTTP Archive files provide contradictory content availability metrics. For example, the number of objects fetched per webpage varies by at least 67 objects in 10% of the measured webpages, and the median difference in load time to reach a webpage is 19.1%, depending on whether the page load was not in response to an initial HTTP redirection [4]. JavaScript-rendered information is systematically more difficult for content extraction systems to index than the server-rendered equivalent. Indexed JavaScript-rendered content gets a 30-50% lower success rate than HTML content.

Other effects could include the loss of, or inability to access, technical documentation that describes an API's endpoints, configuration options, troubleshooting steps, or other material. Content-heavy educational resources and lesson plans' knowledge structure would be lost when only skeleton markup is available. Internal enterprise knowledge bases, which would feed internal AI-assisted workflows, would become less valuable if their knowledge was unable to be extracted. The search barrier creates information silos where primary sources may be authoritative but cannot be synthesized because the retrieval systems do not access the material but, instead, depend on archived pages, secondary interpretations, or bits of information on third-party sites.

3. Generative Engine Optimization as Dual-Audience Architecture

In response to detection of the requesting agent, the suggestion is to return pages in a correctly structured form. For retrieval systems or bot user agents requesting a page, an entire HTML document containing all main page content is returned pre-rendered in immediately parsable form. For human visitors with a regular browser, the same underlying resources are delivered in an

Journal of Information Systems Engineering and Management
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

interactive, JavaScript-powered experience without compromising existing user experiences and with no visible gap between machine and human consumption. Empirical studies of web crawling at scale conclude that automated crawlers can operate at scale and perform reliable user-agent classification. In crawling studies of popular domains, systematic detection and verification of the patterns have been achieved [5]. The cross-domain JavaScript inclusion patterns in the top-ranked pages show that most websites include remote JavaScript from external domains, indicating that client-side execution is common and that dual delivery of information for human and machine consumers is of increasing importance.

It can be implemented at any layer of the system architecture, e.g., the network edge, where requests are not sent to the origin servers but inspected for user-agent strings. The responses can be pre-cached pages intended for the bots and unchanged pages for users. Other server-side techniques serve fully rendered pages with the response to the incoming web request. Hybrid architectures are built on top of the static generation of immutable content and the server-side generation of dynamic and interactive content to provide better performance and accessibility. Edge computing architectures have demonstrated platforms for proximity-based content to be served from the network edge in less than 100 ms to improve the latency for latency-sensitive applications. Similar to connected vehicle communication systems where sub-100 ms response times preserve quality of experience, vehicle-to-infrastructure communication systems have a coverage of 300-500m [6]. The deployments can take many forms, such as radio access network base stations aggregated over multiple sites in enterprise and metropolitan public access networks. These architectural strategies allow a single code base to output multiple versions for different purposes, which avoids the overhead and inconsistency of maintaining parallel content management systems.

4. Content Parity and Structured Semantics

For consistency between the two delivery formats, a dual delivery system should ensure that the human and machine versions do not differ in their omission, paraphrase, or reordering of information. A machine version should have the same content at the same level of granularity and in the same logical order as the human version. Information extraction techniques have a greater chance of success when regularities are observed in the structural patterns. Web data extraction techniques are based on identifying document schemas (repeated structure), semantic boundaries, and hierarchies in HTML documents [7]. HTML documents may be considered semi-structured data because of their nested structure, which can be leveraged by information extraction systems and by machine-facing document versions [7]. Publishing organizations with dual delivery architectures therefore need to specify validation rules for content consistency so that machine-facing document deliveries convey both the same textual content and structural relationships and the same semantic HTML markup and structured metadata as required by human and machine document consumers. Semantic markup conveys machine-readable information about the structure, the relationships between content, and the properties of the metadata [8]. Heading elements (e.g., h1-h6), ARIA landmark regions, and lists help systems understand the document hierarchy. Markup vocabulary schemas such as Schema.org, Open Graph, and JSON-LD specify a page's type, author, publication date, and top category. Such semantically improved pages comply with accessibility standards and may have a positive impact with limited investment. Today, content management systems can automatically create structured annotations from metadata markup inside native content. Publishing platforms are machines to organize knowledge as much as they are intended for human readers. In that respect, supporting a diverse audience of humans with assistive technology and machines performing information retrieval requires a similar, systematic, semantically rich format.

Journal of Information Systems Engineering and Management
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

5. Evaluation and Measurement Frameworks

Evaluation of machine accessibility before and after remediation should include objective measures of how well the content is returned in the retrieved HTML, whether the semantic structure is preserved, whether valid structured metadata is employed, and whether information needed to support citation, such as authorship and publication information, is included.

Comprehensively designed evaluation approaches consider quantitative measurements such as completeness ratios, semantic tag coverage ratios, and metadata field population ratios, as well as qualitative evaluations of the degree to which information extracted from one or multiple information sources is represented in a semantically meaningful composition while retaining context. ACT Rules define a format to specify tests, which can support both automatic and manual evaluation approaches [9]. The ACT Rules Format delineates tests that can be transparently applied and yield reproducible results, documenting how the test procedure describes the applicability and expected outcomes of each element it tests [9].

Modern conformance evaluation methods distinguish between automated and human tests. Research on accessible testing methods shows that some success criteria are within reach of automated programmatic tests, whereas others require the skill of human testers [9]. Systems and organizations architected in a dual-delivery model must define validation rules that articulate the semantic equivalence between the two representations to ensure that the machine-facing representations preserve the structure and relationships that convey meaning to both human and automation consumers.

With repeatable tests like this, HTML returned for different user agent options can be compared. By submitting different bot user agent strings and comparing the resulting HTML, gaps in content can be discovered and fixes can be verified. In detections of bots, research shows that user agent strings are above 90% accurate between bot and human agents, and behavior-based models push the accuracy rate above 96% [10].

A direct comparison of different bot detection methods has been carried out by evaluating the performance of three distinct models on web access log data. The effectiveness of user-agent and behavioral feature analysis has been demonstrated through comprehensive testing [10]. The first model, a logistic regression classifier trained exclusively on user-agent strings with 691 words as features, achieved an Area Under the Curve (AUC) of 0.933, an accuracy of 0.902 (90.2%), a precision of 0.997, and a recall of 0.813. The second model, a LightGBM tree-based classifier that incorporated both user-agent strings (691 words) and user behavior features, demonstrated superior performance with an AUC of 0.990, an accuracy of 0.965 (96.5%), a precision of 0.963, and a recall of 0.969. The third model, another LightGBM classifier utilizing feature reduction with L1 regularization that reduced the user-agent vocabulary from 691 words to just 17 words while still including user behavior features, maintained nearly equivalent performance with an AUC of 0.989, an accuracy of 0.964 (96.4%), a precision of 0.963, and a recall of 0.968 [10]. This feature reduction demonstrates that the number of user-agent vocabulary words can be substantially decreased from 691 to 17 while maintaining equivalent performance.

Model	Features	AUC	Accuracy	Precision	Recall

Journal of Information Systems Engineering and Management
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Logistic Regression	User Agent (691 words)	0.933	0.902	0.997	0.813
LightGBM	User Agent (691 words) + User Behavior	0.990	0.965	0.963	0.969
LightGBM	User Agent (17 words) + User Behavior	0.989	0.964	0.963	0.968

Table 2: Performance Comparison of Bot Detection Models [10]

User-agent strings exhibit structure: programming language identifiers, OS identification, and browser features. As a result, automated requests can be systematically characterized [10]. This makes visibility optimization as much an engineering endeavor as a subjective one.

By making structured testing possible, organizations can objectively report and measure their accessibility progress. Quantitative measures range from ensuring a high degree of content parity between their human and machine views to ensuring that all of the metadata fields of critical attribution elements are populated. This enables engineers to treat machine accessibility as a testable quality attribute instead of an aspirational design goal. Organizations seeking measurement-driven optimization can use baseline visibility scores for their content infrastructure, observe the impact of remediation efforts, and validate whether optimizations are achieving the desired accessibility outcomes without regressions or unintended alterations to content.

Conclusion

With this shift towards AI-mediated information discovery also comes the need to reconsider how we apply methods of adequate information architecture to content. If automated information retrieval systems are to become the primary intermediaries between knowledge bases and their users, machine usability will be equally as important as human-facing interfaces. In dual delivery systems it is possible for resources to be constructed and served to both user agents without either being aware the other exists, thus addressing the underlying visibility problems posed by JavaScript mechanisms that render web content. Organizations that acknowledge automated agents as legitimate participants in their content ecosystem can thus become able and meaningful actors on emerging information networks. By committing to strict content equivalence standards across delivery modalities, explicit semantic markup practices and accessibility progress measurement through evaluation frameworks, content publishers can bridge the visibility divide between web experiences rendered on the client-side and machine-readable formats so that their knowledge assets are properly attributed, cited, and subscribed to AI generated answers, all while preserving the interactivity and expressiveness of the modern web.

Journal of Information Systems Engineering and Management
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

References

[1] Tazin Afrin, et al., "Effective Interfaces for Student-Driven Revision Sessions forArgumentative Writing," in CHI ', 8-13, 2021. [Online]. Available: https://dl.acm.org/doi/epdf/10.1145/3411764.3445683
[2] Nick Nikiforakis, et al., "You Are What You Include:Large-scale Evaluation of Remote JavaScript Inclusions," in CCS'12, October 16–18, 2012. [Online]. Available: https://dl.acm.org/doi/epdf/10.1145/2382196.2382274
[3] Abhinav Jangda, et al., "Not So Fast: Analyzing the Performance of WebAssembly vs. Native Code," Proceedings of the 2019 USENIX Annual Technical Conference. July 10–12, 2019. [Online]. Available: https://www.usenix.org/system/files/atc19-jangda.pdf
[4] Theresa Enghardt, et al., "Web Performance Pitfalls," in Proc. Int. Conf. Passive and Active Measurement, 2019, pp. 286-303. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-15986-3_19
[5] Sebastian Lekies, et al., "25 Million Flows Later - Large-scale Detection of DOM-based XSS," in Proc. ACM SIGSAC Conf. Computer and Communications Security, 2013. [Online]. Available: https://dl.acm.org/doi/epdf/10.1145/2508859.2516703
[6] Yun Chao Hu, et al., "Mobile Edge Computing: A key technology towards 5G," ETSI (European Telecommunications Standards Institute), First edition – 2015. [Online]. Available: https://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp11_mec_a_key_technology_towards_5g.pdf
[7] Emilio Ferrara, et al., "Web data extraction, applications and techniques: A survey," in Knowledge-Based Systems, Volume 70, November 2014, Pages 301-323. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0950705114002640?via%3Dihub
[8] Martin Dostal, et al., "Semantic Markup for Web Applications," in China National Convention Center, 2013. [Online]. Available: https://www.researchgate.net/publication/258239887_Semantic_Markup_for_Web_Applications
[9] W3C Accessibility Guidelines Working Group, "Accessibility Conformance Testing (ACT) Rules Format 1.1," World Wide Web Consortium, W3C Recommendation, 2026. [Online]. Available: https://www.w3.org/TR/act-rules-format/
[10]Takamasa Tanaka, et al., "Bot detection model using user agent and user behavior for web log analysis," Procedia Computer Science, vol. 176, pp. 1621-1625, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050920320871?ref=pdf_download&fr=RR-2&rr=9c9096c9e8b270b5