

Markdown vs Plain Text for LLMs

Author: Kevin Anderson

Date: 03/13/2026

Topic: A Whitepaper on Structure, Readability, Speed, and Output Quality in Human-Model Communication

Abstract

As large language models become core infrastructure for search, software development, enterprise automation, retrieval pipelines, and knowledge work, the format of a prompt is no longer a superficial presentation choice. It is part of the system design. The comparison between Markdown and plain text is especially important because these two formats sit closest to everyday prompting practice: one offers lightweight structure, the other offers almost none.

- Recent research shows that prompt format can materially affect model performance. A 2024 study comparing plain text, Markdown, JSON, and YAML found that formatting changes alone could significantly alter results across reasoning, code generation, and translation tasks, with some of the largest swings appearing in smaller models.

This paper argues that Markdown is generally the better default for most human-LLM collaboration because it adds low-cost semantic structure that improves readability, separation of instructions and context, and downstream usability, while plain text remains useful when tasks are simple, overhead must be minimized, or structure would add little value. OpenAI's prompt engineering guidance explicitly recommends using Markdown and XML to create logical boundaries in prompts, and OpenAI's Custom GPT instruction guidance states that Markdown and structured formatting can improve readability and precision.

Abstract – Core Thesis

The core thesis is therefore practical rather than absolute:

Plain Text

The simplest possible interface. Often optimizes **immediacy** in production terms.

Markdown

Usually the stronger interface for complex prompting. Offers structure without the rigidity of machine-bound formats like JSON. More often optimizes **clarity, consistency, and reuse**.

Emerging benchmark work on "Markdown Awareness" further supports the idea that Markdown quality is a distinct and measurable capability in LLM systems, rather than merely a cosmetic layer.

1. Introduction

The earliest public use of LLMs was dominated by plain text prompting. Users typed a question, added a few sentences of context, and received a response. That interaction felt natural because it mirrored conversation. But as LLM applications matured, prompting stopped being only conversational and became increasingly operational.

Prompts now contain rules, constraints, context passages, examples, role definitions, citations, formatting instructions, and output specifications. In that environment, the difference between plain text and lightly structured text becomes consequential.

Plain text is attractive because it is universal, fast, and frictionless. It imposes almost no syntactic burden. Anyone can write it, any interface can accept it, and the model can process it immediately. Yet plain text also forces humans and models to infer boundaries from prose alone. When a prompt contains multiple instruction layers, evidence excerpts, and constraints, those elements can blur together unless the author manually signals their roles very clearly.

Markdown emerged as a practical answer to that problem. It adds headings, lists, block quotes, tables, and code fences without becoming verbose or machine-rigid. OpenAI's API guidance explicitly notes that Markdown and XML can help models understand logical boundaries in prompts and context data. That guidance is important because it reflects a core production reality: the more complex a prompt becomes, the more valuable lightweight structure becomes.

This paper asks a focused question: when should practitioners use Markdown instead of plain text in LLM systems? The answer is not that Markdown always wins. It is that Markdown usually becomes more valuable as task complexity, collaboration, or output reuse increase. Plain text remains effective for simple interactions. Markdown becomes preferable when ambiguity, maintainability, and quality control start to matter.

2. Why This Comparison Matters

Among all prompt-format comparisons, Markdown versus plain text is arguably the most foundational. Before teams debate Markdown versus JSON or Markdown versus XML, they usually face a simpler decision: should we leave this as prose, or should we impose lightweight structure?

The research basis for asking that question is now strong enough to justify it. The 2024 prompt-format study directly compared plain text and Markdown, along with JSON and YAML, and found that prompt template choice can significantly affect performance. The study does not claim a single format universally dominates, but it clearly rejects the idea that formatting is irrelevant.



Buried Constraints

A model may ignore a constraint because it was buried in a paragraph.



Confused Context

It may confuse context with instruction because there is no clear separator.



Inconsistent Output

It may produce inconsistent output because the prompt did not signal hierarchy clearly enough.

The comparison also matters because plain text remains the mental default for many users. That default is reasonable for quick, low-complexity tasks. But once prompts become repeatable assets rather than one-off chats, the costs of unstructured prompting accumulate. Markdown often becomes the first meaningful improvement because it increases clarity while preserving human readability. OpenAI's help guidance for Custom GPT instructions explicitly recommends headings, lists, and structured formatting for this reason.

3. Two Different Interface Styles

Plain text and Markdown are not just two visual styles. They represent two interface philosophies.

Plain Text – Minimalism

Assumes the model can infer structure from wording and order alone. Its strength is low friction. There are no syntax rules beyond ordinary writing conventions. In simple prompts, that can be enough. A direct question with a small amount of context often works perfectly well as plain text.

Markdown – Lightweight Semantic Scaffolding

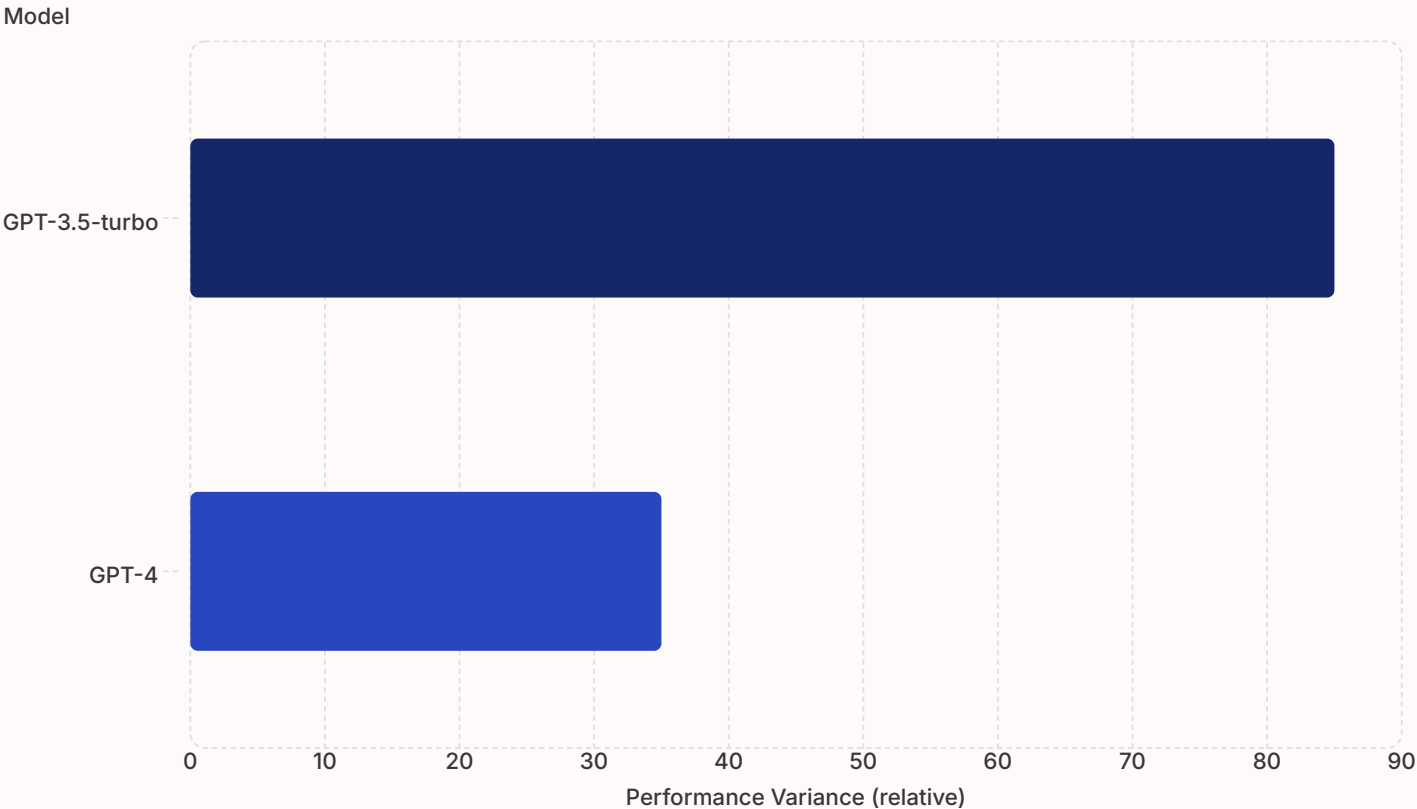
Does not fully formalize content the way JSON or XML do, but it introduces visible hierarchy. A heading can mark a role or section. A bulleted list can turn loose instructions into discrete requirements. A code fence can isolate literal content from surrounding prose. A block quote can clearly mark source text.

These cues help both the author and the model separate functionally different parts of the prompt. OpenAI's official prompt engineering docs explicitly mention using Markdown formatting to establish logical boundaries.

- The practical implication is simple. Plain text relies on **inferential structure**. Markdown supplies **explicit but lightweight structure**. As tasks become more complex, supplied structure tends to outperform inferred structure because it reduces ambiguity and makes review easier.

4. What the Research Says About Format Sensitivity

The most important evidence for this paper comes from *"Does Prompt Formatting Have Any Impact on LLM Performance?"* The authors reformatted semantically equivalent prompts into plain text, Markdown, JSON, and YAML, then evaluated their effects across reasoning, code generation, and translation tasks using GPT-family models.



Their results showed that format can materially affect performance and that different models vary in how robust they are to such changes. GPT-3.5-turbo showed especially large variance, while GPT-4 was more stable but still not fully invariant.

This finding matters because it undermines a common assumption: that if the semantic content is constant, the format does not really matter. It does. Format changes how boundaries, emphasis, and relationships are represented in the context window.

A second relevant line of evidence comes from MDEval, which introduces Markdown Awareness as a measurable quality dimension. The benchmark uses 20,000 instances across English and Chinese and reports strong alignment with human judgments, including a Spearman correlation of 0.791 and accuracy of 84.1 percent. Its underlying claim is crucial for this paper: structure and readability in Markdown outputs are meaningful evaluation targets, not just cosmetic preferences.

20K

MDEval Instances

Across English and Chinese

0.791

Spearman Correlation

Alignment with human judgments

84.1%

MDEval Accuracy

Benchmark accuracy score

Taken together, these sources support a stronger conclusion than "Markdown looks nicer." They support the claim that format influences performance and that Markdown quality is a real, measurable capability.

5. Where Markdown Wins

1 Boundary Clarity

A heading labeled "Task" is more reliable than a sentence that says "Here is the task." A bullet list of constraints is easier to scan than constraints embedded in a paragraph. A fenced block of examples is less likely to blur into instructions than free-flowing prose. OpenAI's guidance endorses exactly this use of formatting to separate context and requirements more clearly.

2 Readability Under Scale

Plain text works well when a prompt is short. As the prompt grows, paragraphs become harder to inspect. Markdown scales better because it preserves navigability. Humans can jump between headings. Reviewers can identify missing sections quickly. Prompt assets stored in repositories or workspaces become easier to maintain.

3 Reuse

Once a prompt becomes a repeatable artifact instead of a one-time message, structure starts to matter much more. Markdown supports versioning, collaborative editing, documentation, and handoff across teams. That portability is one reason it has become so common in developer and documentation ecosystems.

4 Mixed-Content Tasks

Markdown supports mixed-content tasks far better than plain text. A single prompt can include prose, bullets, code samples, references, and sectioned instructions without feeling cluttered. This is especially valuable in research synthesis, content workflows, agent instructions, and RAG settings, where context is heterogeneous rather than purely conversational.

5 Human-Side Speed

Markdown often improves human-side speed. Not necessarily raw model latency, but drafting speed, revision speed, and review speed. A well-structured prompt is easier to fix than an unstructured one. In production environments, that often matters more than a marginal difference in token output.

6. Where Plain Text Wins

Plain text remains strong in three situations.

Simplicity

When the task is short, direct, and self-contained, adding Markdown may not provide meaningful benefit. A concise question with one or two conditions can work perfectly well in ordinary prose.

Conversational Fluidity

Plain text preserves the natural feel of dialogue. In consumer chat settings or early-stage ideation, that can be desirable. Over-structuring a casual interaction may add friction without improving results.

Overhead Minimization

Markdown is lightweight, but it is not free. Headings, bullets, and explicit sections take time to write and can add tokens. When the task does not need segmentation or reuse, plain text can be the faster authoring choice.

There is also a deeper point: plain text forces the author to think in language rather than in template form. For certain creative or exploratory tasks, that looseness can be beneficial. Structure helps consistency, but it can sometimes narrow expressiveness. That is one reason a universal claim in favor of Markdown would be too strong.

The right conclusion is not that plain text is obsolete. It is that plain text is best when the prompt is simple enough that structure adds little value.

7. Speed – Which One Is Faster?

The speed question should be handled carefully. There is not enough evidence to claim that Markdown universally generates faster than plain text in raw model latency terms. The format-sensitivity study shows that output quality and performance can change with format, but it does not establish a universal latency hierarchy in favor of either plain text or Markdown.

A better framework is to separate model speed from workflow speed.

Model Speed

- Time to first token
- Total response latency
- Output token count

Plain text may sometimes be lighter because it contains fewer structural markers. Markdown may sometimes add tokens through headings, bullets, and fences.

Workflow Speed

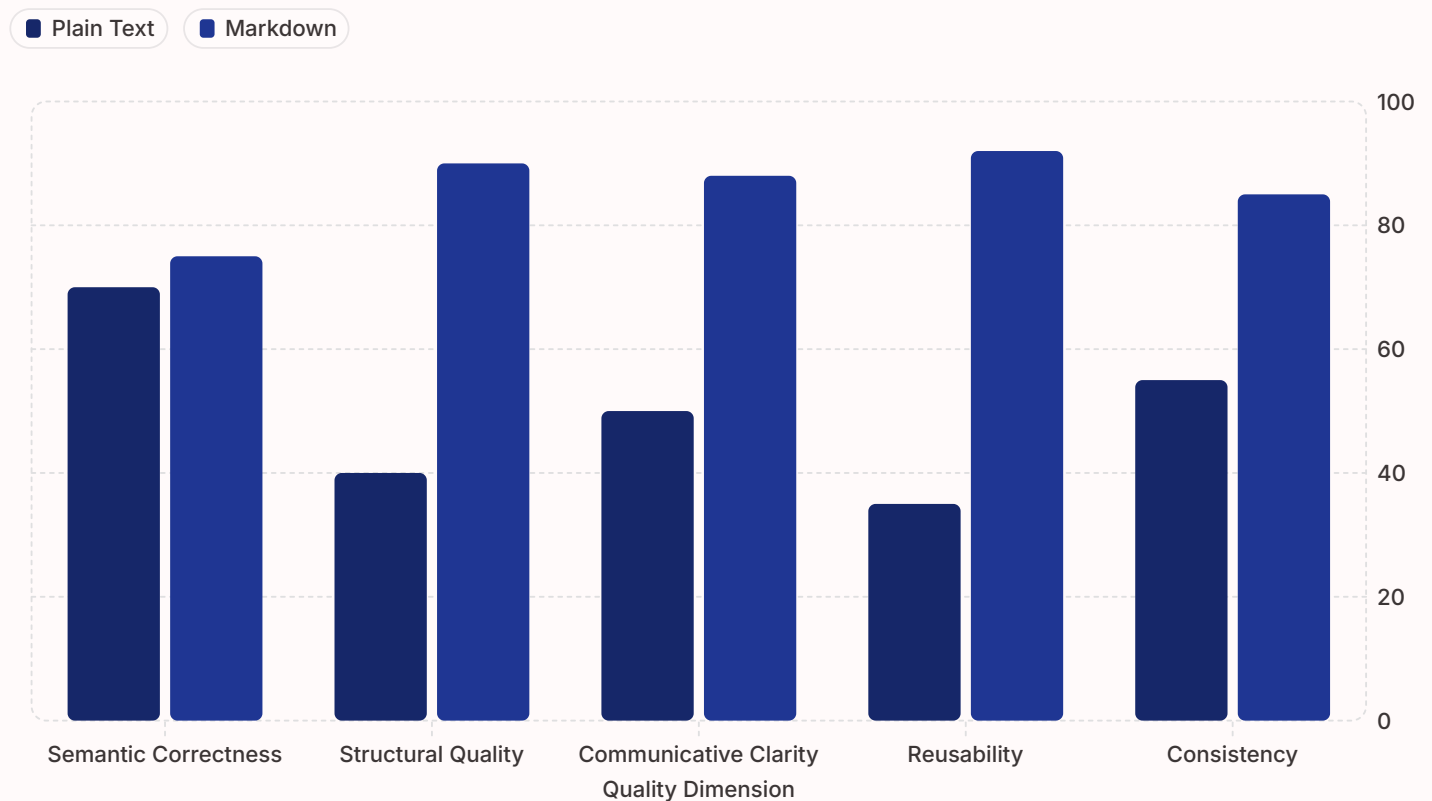
- Drafting time
- Debugging time
- Review time
- Time to usable output

This is where Markdown often gains ground. A structured prompt may take slightly longer to write upfront, but it can reduce confusion, shorten revisions, and lower the chance of prompt failure.

- ❏ **So which is faster?** For one-off simple requests, plain text often wins on immediacy. For repeated or complex tasks, Markdown often wins on end-to-end efficiency. OpenAI's own documentation recommending Markdown for logical boundaries implies this workflow benefit directly.

8. Quality – Which One Produces Better Results?

Quality is multi-dimensional. It can refer to semantic correctness, consistency, readability, structure, completeness, or usability.



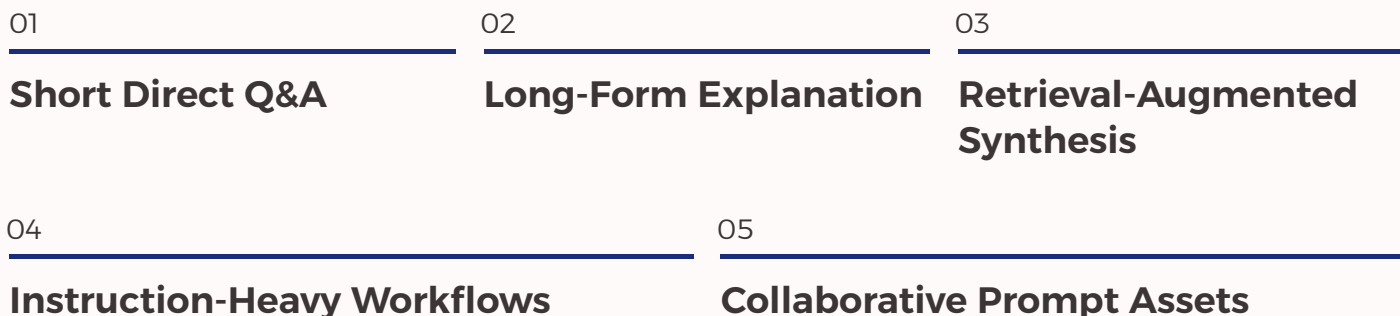
Plain text may perform adequately or even very well when the task is narrow and the desired answer is straightforward. But as prompts become layered, quality often degrades because important distinctions are less explicit.

Markdown improves a different class of quality: structural quality and communicative quality. It helps the model see sections more clearly, and it helps humans understand and evaluate the response more easily. MDEval is relevant here because it treats Markdown structure and readability as a meaningful capability, not just formatting flair.

That does not mean Markdown guarantees higher semantic accuracy. It means it often improves the conditions under which accurate, consistent, and reusable outputs are produced. Put differently, plain text can be sufficient, but Markdown more often becomes advantageous as quality requirements become broader than "answer the question."

9. Recommended Evaluation Framework for Teams

Teams deciding between Markdown and plain text should benchmark both on their real workloads. A useful evaluation should include at least five task families:



These categories expose the tradeoff between immediacy and structure clearly.

Speed Metrics

- Authoring time
- Time to first token
- Total response time
- Output token count
- Revision time

Quality Metrics

- Semantic accuracy
- Adherence to constraints
- Readability
- Completeness
- Reviewer effort

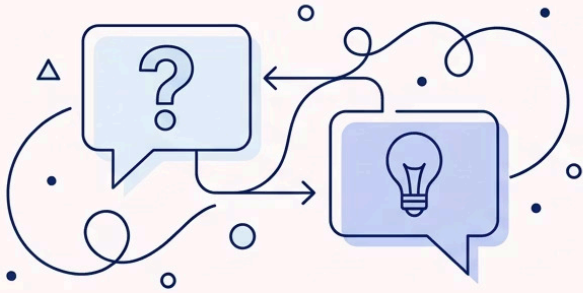
- ❑ The best hypothesis is not "Markdown always beats plain text." It is that plain text is best for simple, one-shot interaction, while Markdown is best for prompts that need hierarchy, consistency, collaboration, or reuse. That hypothesis is the one most consistent with the current literature and official guidance.

10. Best-Practice Architecture

Plain Text for Simplicity, Markdown for Assets

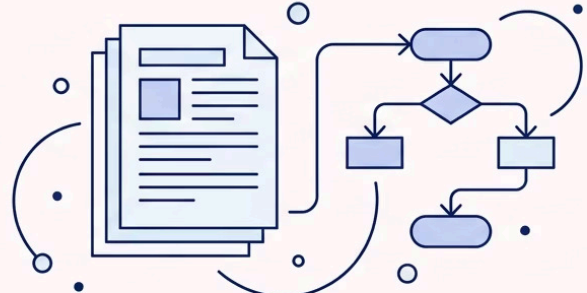
The strongest production pattern is usually layered rather than ideological.

Plain Text:



Quick questions
Exploratory chat
Light ideation

Markdown:



Reusable instructions
Research templates
Content workflows

This division reflects how most teams naturally mature. They begin with plain text because it is fast. They move toward Markdown when scale, consistency, and maintenance begin to matter. OpenAI's prompt and instruction guidance supports this trajectory by recommending structured formatting when clarity and precision are important.

11. Limitations & 12. Conclusion

Limitations

This comparison has limits.

Limited Dedicated Benchmarks

Direct public benchmarking focused exclusively on Markdown versus plain text across many modern frontier models remains limited. The strongest evidence comes from broader prompt-format studies rather than a dedicated canonical benchmark.

Model Behavior Changes Over Time

Improvements in instruction following, context handling, and output controls may alter the relative advantage of structure.

Task Context Matters

A result that holds in long-form synthesis may not hold in casual chat or short extraction tasks. Any blanket claim without workload context is methodologically weak.

Conclusion

The comparison between Markdown and plain text is really a comparison between implicit and explicit lightweight structure. Plain text is faster to begin with, more conversational, and often sufficient for simple tasks. Markdown adds headings, lists, fences, and visible hierarchy that help both humans and models manage complexity.

When the main goals are simplicity and immediacy, plain text remains a perfectly valid interface. When the main goals are clarity, consistency, reuse, and human-model collaboration on complex tasks, Markdown is usually the stronger default. Research on prompt formatting and Markdown Awareness supports treating this choice as an engineering variable rather than a stylistic preference.

The most practical answer is not to abandon plain text. It is to recognize where plain text stops scaling cleanly. Markdown becomes valuable precisely at that point: when the prompt is no longer just a message, but a working document.

References

1

Prompt Format Impact Study

He, J., Rungta, M., Koleczek, D., Sekhon, A., Wang, F. X., and Hasan, S. *Does Prompt Formatting Have Any Impact on LLM Performance?* arXiv:2411.10541.

2

MDEval Benchmark

Chen, Z., Liu, Y., Shi, L., Wang, Z.-J., Chen, X., Zhao, Y., and Ren, F. *MDEval: Evaluating and Enhancing Markdown Awareness in Large Language Models.* arXiv:2501.15000.

3

OpenAI Prompt Engineering Guide

OpenAI. *Prompt engineering guide.*

4

OpenAI Custom GPT Guidelines

OpenAI. *Key Guidelines for Writing Instructions for Custom GPTs.*

5

OpenAI API Best Practices

OpenAI. *Best practices for prompt engineering with the OpenAI API.*

Next in this series: *Markdown vs YAML for LLMs*, or the *benchmark synthesis article* to close the loop cleanly.