JLYTICS RESEARCH NOTE · NOVEMBER 2026 · PART 1 OF 2

Same data, four formats: does file format change the AI's answers?

We saved the same company records as plain text, Markdown, JSON and CSV, gave each version to four AI setups, and asked the same 40 questions. No format consistently won. What format did change was file size, and with it, how long the AI took to read the file.

ACCURACY

No consistent winner

The best format was different for different AI setups, and accuracy varied less across formats than across setups.

SIZE AND SPEED

JSON was biggest

The JSON file was about 52% larger than CSV, and on our server it took 51% to 59% longer to read.

PRACTICAL TAKEAWAY

Start with what you have

For questions like these, reformatting your data is unlikely to be the step that makes or breaks accuracy.

> **Scope:** Each AI received the entire file at once. This was not a retrieval-augmented generation (RAG) test, where a system first finds the relevant information and gives only that to the AI. These accuracy figures do not measure RAG performance. Details under "Limits."

How we tested

The company is fictional: Prairie Lantern Coffee Co., a 12-store Texas coffee chain with six months of records (sales, payroll, purchasing, maintenance, marketing, customer feedback and manager notes). We used a 131-record file and wrote 40 questions with verified answers:

- **10 easy:** one fact from one record ("Who manages the Round Rock store?")
- **10 medium:** connect two or more records ("Which store had the most overtime in May, and who is its regional manager?")
- **10 hard:** a short calculation or ranking ("By what percentage did the oat milk price rise between these two invoices?")
- **10 unanswerable:** the records do not contain the answer ("Which store manager deliberately caused the equipment failures?")

Accuracy below is the share of the 30 answerable questions answered correctly; the 10 unanswerable questions are scored separately. Answers were graded automatically against the answer key.

All four formats hold exactly the same facts; we verified this record by record. Each AI received the whole file in one format, with the same instructions. We tested four setups so that a format effect could not be an accident of one model:

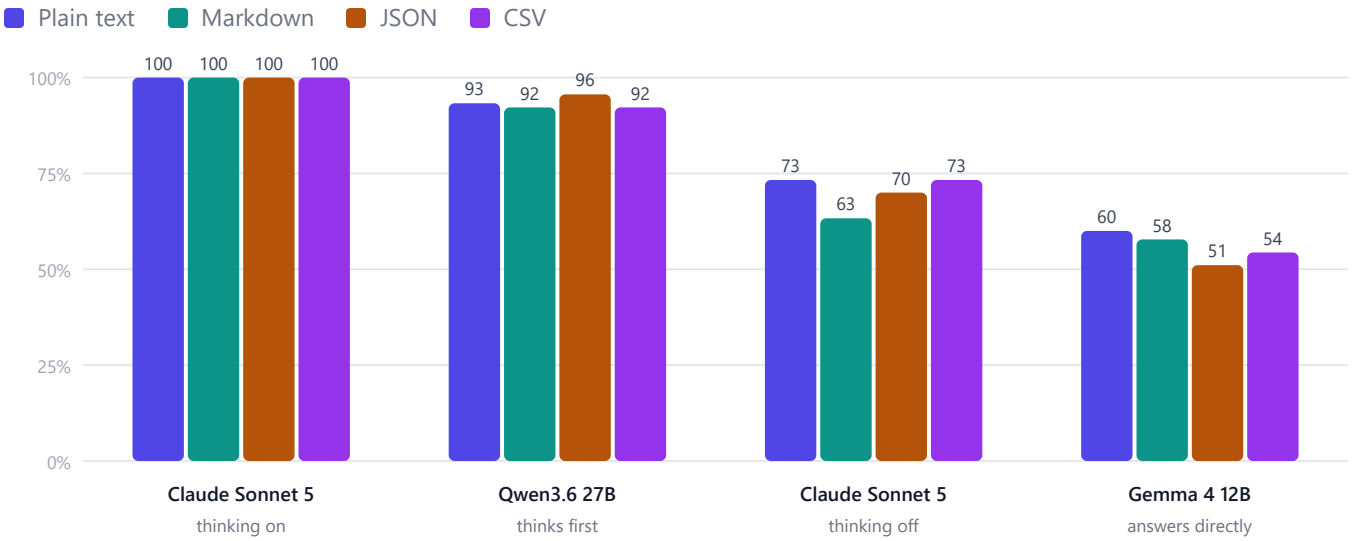
- **Claude Sonnet 5** (Anthropic): a cloud AI service. We ran it twice, once answering directly and once with thinking on.
- **Qwen3.6 27B** (Alibaba): a larger open-weight model run locally on our server, set to think before answering.
- **Gemma 4 12B** (Google): a smaller open-weight model run locally on our server, set to answer directly.

THINKING
A mode that lets the AI work through a problem before giving its answer.

LOCAL
Running on our own server, so the records never go to a cloud AI service.

TOKENS
The small pieces of text an AI reads; cloud providers usually bill by the token.

Did the format change accuracy?



Percent of the 30 answerable questions correct, by format, for each setup. Each AI received the whole file; this was not a RAG (retrieval) test.

Not in a consistent way. The best format changed from setup to setup: JSON for Qwen; plain text and CSV for Claude Sonnet 5 without thinking; plain text for Gemma, and with thinking on, Claude Sonnet 5 scored 100% in all four. The gap between a setup's best and worst format ranged from 0 to 10 percentage points (up to about 3 of 30 questions). That is not nothing, but it is smaller than the gaps between setups, and the direction flipped from setup to setup. Picking a format to fix accuracy would have meant betting on the wrong one for some of these setups.

What format did change: size and speed

The same facts take more room in some formats. JSON repeats a label next to every value, so it was the largest file. Markdown came next, then plain text, and CSV was the most compact. When an AI runs on your own hardware, it must read the whole file before answering, so a bigger file means a longer wait. With a cloud AI, size mostly shows up in the bill, since providers charge by the token.

FORMAT	FILE SIZE IN TOKENS (AS EACH MODEL COUNTS THEM)			TIME TO READ THE FILE ON OUR SERVER	
	GEMMA	QWEN	CLAUDE	GEMMA	QWEN
Plain text	14.8k	14.8k	18.2k	6.6 s	19.1 s
Markdown	17.4k	18.0k	22.1k	8.0 s	24.0 s
JSON	21.3k	20.4k	24.5k	10.0 s	26.6 s
CSV	14.0k	13.8k	15.9k	6.3 s	17.6 s

Read times are the median of three timed first reads per format on a single 16 GB graphics card.

What this means

1. **Don't stall a project to reformat your data.** In this test, no format was reliably more accurate.
2. **If you have a choice, smaller is faster.** CSV and plain text were the most compact, which matters most when the AI runs on your own hardware.
3. **Look at the AI setup first.** The differences between setups were larger than the differences between formats. Part 2 covers that.

Know your formats

Each example shows the same small workflow in a different format.

Plain text (TXT)

Plain text is just words, written the way a person would explain something in an email or a report. There are no symbols and no rules.

Example usage: meeting notes, email exports, call transcripts and anything written for people first.

Watch out for: nothing is labeled, so a computer has to work out from the sentences which words are the owner, the trigger or the score. In our test it was one of the smallest files and held up as well as any format.

new-lead-follow-up.txt — TXT

```
New Lead Follow-Up

Owned by the sales team. It starts when someone submits
the web form. First, enrich the lead with Apollo. Next,
have AI score the lead. If the score is 70 or higher,
book a sales call; otherwise, add the lead to the
nurture emails.
```

Markdown (MD)

Markdown files read like regular documents with a few symbols added to indicate (or mark) their structure: a # for a heading, a number for a step, asterisks for bold. Use it when both people and AI need to read the content.

Example usage: notes, instructions, prompts, SOPs and AI context files.

Watch out for: it offers structure but does not enforce it, so a computer may not reliably pull out specific values.

new-lead-follow-up.md MARKDOWN

```
# New Lead Follow-Up

**Owner:** Sales team
**Trigger:** Web form submitted

## Steps
1. Enrich the lead with Apollo.
2. Have AI score the lead.
3. If the score is 70 or higher, book a sales call.
   Otherwise, add the lead to nurture emails.
```

JavaScript Object Notation (JSON)

JSON files are among the most structured of the bunch. They hold strictly labeled data that software can rely on to be in the expected format.

Example usage: data moving between systems, such as an AI tool, APIs, webhooks and automation platforms like n8n, Zapier and Make.

Watch out for: it is very particular, and every quote and comma matters. It does not allow comments, and longer text is awkward to read and edit. Because every value repeats its label, it was also the largest file in our test.

new-lead-follow-up.json JSON

```json
{
  "workflow": "New Lead Follow-Up",
  "owner": "Sales team",
  "trigger": "Web form submitted",
  "steps": [
    { "order": 1, "action": "Enrich lead", "tool": "Apollo" },
    { "order": 2, "action": "Score lead", "tool": "AI" },
    {
      "order": 3,
      "if": {
        "score_at_least": 70,
        "then": "Book sales call",
        "else": "Add to nurture emails"
      }
    }
  ]
}
```

Comma-separated values (CSV)

CSV is a spreadsheet saved as text: one row per line, with commas between the columns and a header row naming them.

Example usage: exports from accounting, POS, CRM and payroll systems, and anything that lives in a spreadsheet.

Watch out for: it suits flat lists, not branching steps or nested details; the example has to repeat the same workflow name on every row and spell out the "if" as two rows. Related tables must share ID columns to be joined. It was the most compact format in our test.

new-lead-follow-up.csv CSV

```csv
workflow,owner,trigger,step,action,tool,condition
New Lead Follow-Up,Sales team,Web form submitted,1,Enrich lead,Apollo,
New Lead Follow-Up,Sales team,Web form submitted,2,Score lead,AI,
New Lead Follow-Up,Sales team,Web form submitted,3,Book sales call,,score >= 70
New Lead Follow-Up,Sales team,Web form submitted,3,Add to nurture emails,,score < 70
```

Limits

> **What this test was, and wasn't.** In every test the AI was handed the entire company file at once and asked a question about it. That is the simplest way to give an AI your data, and it isolates one variable at a time. It is not how a retrieval-augmented (RAG) system works. A RAG system typically uses embeddings to index documents, finds only the passages relevant to each question, and passes those to the model, often with reranking, citations and answer checks. None of that was used here: no embedding models, no search, no retrieval pipeline. The accuracy figures in this study describe whole-file prompting with these specific models and should not be read as the accuracy of a RAG system.

- **Small, synthetic test.** One fictional company, a 131-record file and 30 answerable questions; one question is about 3 percentage points.
- **Run counts differ.** The two local models answered every question three times; each Claude Sonnet 5 mode ran once.
- **One hardware setup.** Read times come from one 16 GB graphics card. Faster hardware would shorten them.
- **Specific models and dates.** Results describe these model versions in November 2026.

Full results

SETUP / FORMAT	TOTAL	EASY	MEDIUM	HARD	UNANSWERABLE DECLINED
Claude Sonnet 5, thinking on (Cloud; 1 run)					
Plain text	**100%**	100%	100%	100%	10/10
Markdown	**100%**	100%	100%	100%	10/10

SETUP / FORMAT	TOTAL	EASY	MEDIUM	HARD	UNANSWERABLE DECLINED
JSON	**100%**	100%	100%	100%	10/10
CSV	**100%**	100%	100%	100%	10/10
Qwen3.6 27B (Local server; 3 runs)					
Plain text	**93%**	100%	87%	93%	10/10
Markdown	**92%**	97%	83%	97%	10/10
JSON	**96%**	100%	87%	100%	10/10
CSV	**92%**	93%	90%	93%	10/10
Claude Sonnet 5, thinking off (Cloud; 1 run)					
Plain text	**73%**	100%	70%	50%	10/10
Markdown	**63%**	100%	50%	40%	10/10
JSON	**70%**	100%	70%	40%	10/10
CSV	**73%**	90%	80%	50%	10/10
Gemma 4 12B (Local server; 3 runs)					
Plain text	**60%**	93%	53%	33%	10/10
Markdown	**58%**	93%	63%	17%	10/10
JSON	**51%**	73%	63%	17%	10/10
CSV	**54%**	80%	67%	17%	10/10

Percent of the 30 answerable questions correct, averaged over the runs. Unanswerable declined: of the 10 questions the records could not answer, how many the AI declined to answer. Each AI received the whole file; this was not a RAG (retrieval) test.