Does prompt design stop AI models inventing invoice data?

A pilot test of four prompts on two Claude models

Lewis Wilson

Independent researcher · lewis-wilson.com

Businesses now use AI models to read invoices. This study tested whether the way the prompt is written changes how often a model makes up values that are not on the page. One sentence allowing a blank answer cut invented values by about 80%. Worked examples and detailed field definitions made no difference.

1,920	80	4	2
MODEL CALLS	SYNTHETIC INVOICES	PROMPT DESIGNS	CLAUDE MODELS

ABSTRACT

Businesses increasingly use large language models to read documents such as invoices and turn them into structured data. The main risk is not misreading but invention: a confident, plausible value for a field that is not on the page. This pilot study tested whether prompt design changes that risk. Two Claude models (Opus 5.5 and Haiku 5.5) extracted six fields from synthetic UK invoices under four prompts: plain, structured, few-shot, and structured with a self-check instruction. Experiment 1 (40 invoices, 960 calls) found that both models read every printed value correctly under every prompt, but the first three prompts filled in a missing VAT figure every time, while the self-check prompt returned null every time. Experiment 2, designed in advance to test this directly (40 invoices with missing or inconsistent fields, 960 calls), confirmed the effect: the invented-value rate fell from 39 to 50% to 8 to 9%. The structured and few-shot prompts gave no improvement over a one-line plain prompt. One failure mode, reporting an account reference as the invoice number, was largely unaffected by any prompt.

Keywords: large language models; document extraction; hallucination; prompt engineering; invoices

KEY FINDINGS

1. Both models read every printed value correctly, under every prompt, including on rotated and low-quality scans. Accuracy was not the problem; invention was.
2. One explicit instruction ("if a field is missing, return null rather than guessing") cut invented values from 42 of 90 to 7 of 90 for Opus and from 35 of 90 to 8 of 90 for Haiku. Role-setting, field definitions and worked examples made no measurable difference.
3. The self-check prompt has a cost on the smaller model: Haiku withheld a correctly printed total on 8 of 30 responses where the printed figures did not add up.

1 Introduction

Extracting key fields from invoices and receipts is a long-standing document-AI task with clear commercial value, studied through shared benchmarks and purpose-built models (Huang, Z. et al., 2019; Park et al., 2019; Kim et al., 2022). General-purpose language models that read document images now let any small business automate it with a prompt instead of a trained system. The task is attractive because it is repetitive and the output looks easy to check. In practice, an extraction system is usually trusted once it has been spot-checked a few times, and its errors then flow quietly into accounts.

Two kinds of error matter. A misreading (a 7 read as a 1) is the classic optical character recognition problem, and errors at that stage propagate into everything downstream (Kim et al., 2022). An invented value is different: the model returns a confident, correctly formatted answer for something that is not on the document, often by working it out from other fields. In the terms of the hallucination literature, this is an extrinsic hallucination: output "that cannot be verified from the source content" (Ji et al., 2023, Section 2.1). Language models are known to produce "plausible yet nonfactual content" (Huang et al., 2025, abstract), and invented values are harder to catch than misreadings because they look right and are often arithmetically consistent with the rest of the record.

Practitioner guidance on prompting commonly recommends giving the model a role, defining each field, supplying worked examples and asking it to check its work. Worked examples, or few-shot prompting, became standard after Brown et al. (2020) showed that large models can perform tasks from a handful of demonstrations. Anthropic's own guidance adds a further step: "Explicitly give Claude permission to admit uncertainty. This technique can drastically reduce false information" (Anthropic, n.d.). The research evidence on which of these steps matters is mixed. Demonstrations may help mainly by showing the format and range of answers rather than by teaching the task (Min et al., 2022); performance can swing widely with small changes in prompt formatting (Sclar et al., 2024); and on Spanish electricity invoices, one recent unreviewed study found that few-shot examples raised accuracy by more than 19 percentage points (Gómez & Sánchez, 2026). We found few studies that measure invented values for fields missing from a document, as opposed to overall extraction accuracy, and fewer still that compare common prompt designs on this outcome using the same documents. This pilot asks one question: **does the way the prompt is written change how often a model gets invoice details wrong or makes them up?**

To answer it, two experiments compared four common prompt designs on the same synthetic invoices, where the correct answer to every field is known. Experiment 1 used ordinary invoices, including messy scans. Experiment 2, planned in advance, focused on invoices where a field was

missing or the figures did not add up, because that is where invented values occur. Section 2 describes the method, Section 3 the results, and Section 4 what they mean for businesses that use AI to extract data from documents.

2 Method

2.1 Documents

All documents were synthetic UK invoices generated by a Python script (seed 2026 for Experiment 1, seed 2027 for Experiment 2), so the correct answer for every field is known exactly. Every supplier, customer, address and number is fictional. Invoices use four layouts (classic, banner, receipt and split), a range of date and currency formats, and between one and six line items.

- **Experiment 1:** 40 invoices. 30 are clean. 10 (25%) are deliberately messy: rotated scans, low-quality scans, unusual fonts, or (on 3 invoices) no VAT line.
- **Experiment 2:** 40 new invoices, all clean, in four cases of 10 each:
- **missing VAT:** no VAT line is printed (correct answer: null);
- **missing invoice number:** no invoice number is printed, but a purchase order number and an account reference are (correct answer: null);
- **missing invoice date:** no issue date is printed, but a due date is (correct answer: null);
- **inconsistent total:** the printed total does not equal net plus VAT, as with a supplier typo (correct answer: the figures exactly as printed).

2.2 Fields and prompts

Six fields were extracted: supplier, invoice date, invoice number, net, VAT and total. Four prompts were compared (Table 1):

Table 1. The four prompts compared.

PROMPT	CONTENT
A, plain	"Extract the invoice details from this invoice", plus the six key names
B, structured	A role ("careful accounts-payable assistant"), a definition of each field including what it is not (for example, not a PO or account number), and a strict JSON output format
C, few-shot	B plus three worked examples (text transcriptions of other invoices)
D, self-check	B plus two instructions: confirm net + VAT = total, and "only report values that are printed on the invoice; if a field is missing, return null rather than guessing"

Prompt D is the only prompt that explicitly permits a null answer. The full prompt texts are available from the author on request.

2.3 Models and runs

Each invoice was sent as an image to two models, claude-opus-5-5 and claude-haiku-5-5 (default effort, medium), through the Anthropic Message Batches API on 19 and 20 October 2026. Each invoice was run three times under each prompt: 40 invoices × 4 prompts × 3 repeats = 480 calls per

model per experiment, 1,920 calls in total. Every raw response is stored with its prompt hash, model name and timestamp.

2.4 Scoring and analysis

Scoring rules were fixed before any model was run. Values are compared after normalisation (case, spacing, punctuation, currency symbols and common UK date formats), so a correct answer in a different format still counts; format compliance is reported separately. An **invented value** is a non-null answer for a field that is not printed. A **confident error** is any wrong non-null answer.

Confidence intervals are 95% cluster-bootstrap intervals over documents (10,000 resamples). Prompts are compared with exact McNemar tests on whole-document correctness, using the majority answer across the three repeats, with Holm adjustment across the six pairs. Experiment 2's primary outcome and analysis were written down and dated before any Experiment 2 data was generated (available from the author on request). The plan was not lodged with an external registry; "pre-registered" in this paper means fixed in advance in this way.

3 Results

3.1 Experiment 1: printed values are read correctly; missing values are invented

Both models read every printed field correctly under every prompt, on clean and messy invoices alike. Rotated and low-quality scans caused no errors.

All errors came from the three invoices with no VAT line. Under prompts A, B and C, both models returned a VAT figure every time (9 of 9 responses per prompt), equal to the total minus the net. Under prompt D, both returned null every time (0 of 9) (Table 2).

Table 2. Experiment 1 results by prompt (40 invoices, three repeats).

	A PLAIN	B STRUCTURED	C FEW-SHOT	D SELF-CHECK
Whole-document accuracy, Opus	92.5%	92.5%	92.5%	**100%**
Whole-document accuracy, Haiku	92.5%	92.5%	92.5%	**100%**
Missing VAT invented (both models)	9/9	9/9	9/9	**0/9**
Date in requested format, Haiku	48%	100%	100%	100%
Identical answer on all 3 repeats, Haiku	68%	100%	93%	100%

With only three affected invoices, the difference between D and the other prompts was not statistically significant (exact McNemar p = 0.25 for each comparison). This motivated Experiment 2.

The structured prompts did help in one practical way: with the plain prompt, Haiku returned dates in the requested YYYY-MM-DD format only 48% of the time and gave the same answer on all three repeats for only 68% of invoices. Structure improved consistency and format, not correctness.

3.2 Experiment 2: the self-check instruction cuts invented values

Across the 30 invoices with a missing field (90 responses per prompt per model; Table 3, Figure 1):

Table 3. Experiment 2 invented-value rate on invoices with a missing field.

INVENTED-VALUE RATE	A PLAIN	B STRUCTURED	C FEW-SHOT	D SELF-CHECK
Opus	47% (42/90)	50% (45/90)	48% (43/90)	**8% (7/90)**
Haiku	39% (35/90)	43% (39/90)	40% (36/90)	**9% (8/90)**

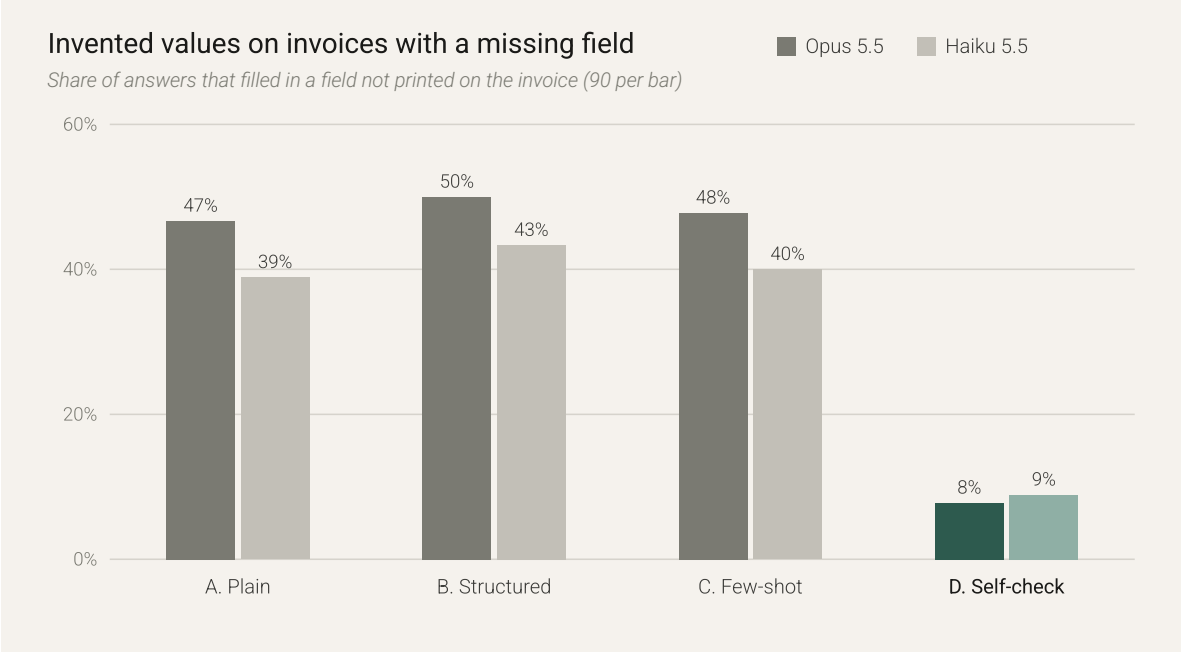


Figure 1. Invented-value rate on the 30 invoices with a missing field, by prompt (90 responses per bar). Only the self-check prompt, which permits a null answer, reduced invention.

For Opus, prompt D produced the correct whole-document answer on significantly more of these invoices than each of the other prompts (D correct where the other was wrong on 11 to 12 invoices, the reverse on none; Holm-adjusted $p \leq 0.005$ for all three comparisons). For Haiku the direction was the same and the comparison with B was significant (Holm-adjusted $p = 0.038$); the comparisons with A ($p = 0.090$) and C ($p = 0.059$) were not. Prompts A, B and C did not differ from each other for either model.

The effect depended strongly on which field was missing (Table 4, Figure 2):

Table 4. Invented-value rate by which field was missing (prompts A to C pooled vs prompt D).

INVENTED-VALUE RATE BY CASE	OPUS A–C	OPUS D	HAIKU A–C	HAIKU D
Missing VAT	100%	**0%**	100%	**0%**
Missing invoice number	40%	23%	17 to 30%	27%
Missing invoice date	0 to 10%	0%	0%	0%

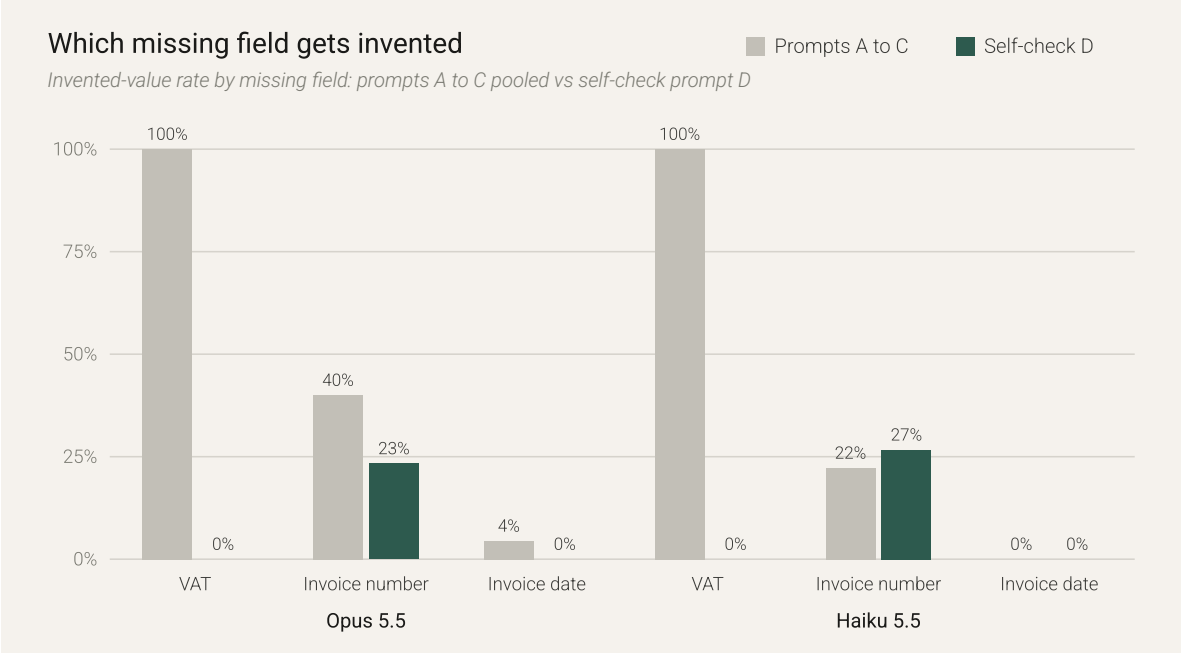


Figure 2. Invented-value rate by missing field, prompts A to C pooled against prompt D. Missing VAT drove almost all of the effect; missing invoice numbers were barely helped.

- **Missing VAT** drives almost all of the effect. Without the instruction, both models always calculated VAT as total minus net. With it, they never did.
- **Missing invoice number** was barely helped. The invented "invoice numbers" were not made up from nothing: they were the account reference printed elsewhere on the invoice (short codes such as EF361), on the same few invoices each time. This happened even though prompts B to D define the invoice number as "not a purchase order, account, customer… number". A wrong field confidently mapped to the right label is the hardest error for a human checker to spot.
- **Missing invoice date** was rarely a problem. Both models almost always left it empty; Opus returned a date in 4 responses (3 under B, 1 under C).

3.3 The cost of caution

On the 10 invoices whose printed total did not add up, the correct behaviour is to report the printed figures as they are. Opus did so under every prompt, every time. Haiku did so under A and B; under C it once "corrected" a total; and under D it returned null for the total on 8 of 30 responses (5 of 10 invoices), withholding a figure that was clearly printed. The self-check instruction made the smaller model over-cautious when the arithmetic check failed.

4 Discussion

What helps. The most effective change was one sentence giving the model explicit permission to say a field is not there. This is an independent test of the vendor's own advice (Anthropic, n.d.), and it fits the wider research on abstention, where instruction hints such as "answer the question only if answerable" are an established way to reduce hallucination (Wen et al., 2025, Section 3.3.2). It is also consistent with evidence that language models carry some signal about what they do and do not know (Kadavath et al., 2022): here, both models recognised that VAT was missing as soon as they were allowed to say so. Elaborate prompting, with a role, careful field definitions and worked

examples, improved output format and consistency but did nothing for invented values, which is consistent with the view that demonstrations mainly convey format (Min et al., 2022). A practical reading: models are strongly inclined to complete the form, and they need to be told that leaving a field empty is an acceptable answer.

The contrast with Gómez and Sánchez (2026), where few-shot examples helped a great deal, is probably explained by the task. Their study measured accuracy across many fields on text versions of the invoices, where errors in reading and mapping fields leave room for examples to help. In this pilot the models read every printed value correctly, so the only errors left were values that were not there, and examples of complete invoices give no guidance on what to do when a field is missing.

What does not. Prompting did not fix field confusion. When a plausible substitute was printed (an account reference instead of an invoice number), even the best prompt let it through on about a quarter of responses. Field definitions that explicitly excluded account numbers did not stop it. This error type needs a check outside the model, such as a format rule for invoice numbers or a lookup against the supplier's known numbering. Published extraction systems already use checks of this kind: discarding any value that does not appear in the document's text (Perot et al., 2024), and checking that each value appears verbatim in the text and that totals, taxes and line items add up (Hemmer et al., 2025). A verbatim check alone would have caught every invented VAT figure in this pilot, because a computed value never appears on the page.

Trade-offs differ by model. The self-check prompt had no downside for the larger model but made the smaller model withhold correct values when the figures did not reconcile. Over-caution of this kind is a known side effect of abstention techniques, including self-checking (Wen et al., 2025, Section 5). The right prompt therefore depends on the model and on which error is more costly: a missing value that a person fills in, or an invented one that nobody notices.

Implications for practice. For a business adopting AI extraction, the findings suggest four low-cost steps:

1. include an explicit "return null if not printed" instruction;
2. test the system on documents with missing fields, not only on complete ones, because a test set of complete documents would have shown near-perfect accuracy here;
3. add simple rule-based checks for the fields models confuse (invoice number vs account reference);
4. re-test whenever the model or prompt changes.

5 Limitations

- Synthetic documents of one type (UK invoices). Real invoices are more varied and may be harder.
- Two models from one provider, tested at one point in time. Results may change as models are updated.
- A small sample: 10 invoices per case in Experiment 2, so rates for individual cases are indicative only.
- Results hold for these four prompt wordings. Language models can be highly sensitive to small changes in prompt format, and that sensitivity differs between models (Sclar et al., 2024).

- Prompts were compared as whole packages. Prompt D adds both a null instruction and an arithmetic check, and this design cannot separate their effects.
- The few-shot examples were text transcriptions, not images.
- This is an unreviewed pilot study by a single author.

6 Conclusion

In this pilot, both Claude models read printed invoice values perfectly but routinely invented values for fields that were not there. A single instruction permitting a null answer cut invented values by about 80%, while standard prompt-engineering techniques had no effect on them. One error type, an account reference reported as the invoice number, survived every prompt and needs a check outside the model. The broader lesson for anyone deploying AI on documents is to test on what is missing, not only on what is there.

Data and code availability

The synthetic invoices, prompts, raw model responses, analysis scripts and the Experiment 2 plan are available from the author on request. All results in this paper can be regenerated from the stored responses without further API calls. Total API cost for both experiments: about US$8.50.

Competing interests

The author declares no competing interests.

Funding

This work received no specific funding. API costs were paid by the author.

Use of AI

Lewis Wilson designed this study. AI tools (Claude Code, Anthropic) were used to generate the synthetic invoices, run the models and collect the data, analyse the results, and help draft the text. The two models under test are named in Section 2.3. The author takes full responsibility for the content of this paper.

References

Anthropic. (n.d.). *Reduce hallucinations*. Claude Platform Docs. Retrieved October 22, 2026, from https://platform.claude.com/docs/en/test-and-evaluate/strengthen-guardrails/reduce-hallucinations

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., … Amodei, D. (2020). Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, & H. Lin (Eds.), *Advances in Neural Information Processing Systems 33* (pp. 1877–1901). Curran Associates. https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html

Gómez, J., & Sánchez, J. (2026). *Information extraction from electricity invoices with general-purpose large language models* (arXiv:2604.25927). arXiv. https://doi.org/10.48550/arXiv.2604.25927

Hemmer, A., Coustaty, M., Bartolo, N., & Ogier, J.-M. (2025). Neurosymbolic information extraction from transactional documents. *International Journal on Document Analysis and Recognition, 28*(3), 475–485. https://doi.org/10.1007/s10032-025-00530-0

Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., & Liu, T. (2025). A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems, 43*(2), 1–55. https://doi.org/10.1145/3703155

Huang, Z., Chen, K., He, J., Bai, X., Karatzas, D., Lu, S., & Jawahar, C. V. (2019). ICDAR2019 competition on scanned receipt OCR and information extraction. In *2019 International Conference on Document Analysis and Recognition (ICDAR)* (pp. 1516–1520). IEEE. https://doi.org/10.1109/ICDAR.2019.00244

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., & Fung, P. (2023). Survey of hallucination in natural language generation. *ACM Computing Surveys, 55*(12), 1–38. https://doi.org/10.1145/3571730

Kadavath, S., Conerly, T., Askell, A., Henighan, T., Drain, D., Perez, E., Schiefer, N., Hatfield-Dodds, Z., DasSarma, N., Tran-Johnson, E., Johnston, S., El-Showk, S., Jones, A., Elhage, N., Hume, T., Chen, A., Bai, Y., Bowman, S., Fort, S., … Kaplan, J. (2022). *Language models (mostly) know what they know* (arXiv:2207.05221). arXiv. https://doi.org/10.48550/arXiv.2207.05221

Kim, G., Hong, T., Yim, M., Nam, J., Park, J., Yim, J., Hwang, W., Yun, S., Han, D., & Park, S. (2022). OCR-free document understanding transformer. In S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, & T. Hassner (Eds.), *Computer vision – ECCV 2022* (Lecture Notes in Computer Science, Vol. 13688, pp. 498–517). Springer. https://doi.org/10.1007/978-3-031-19815-1_29

Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H., & Zettlemoyer, L. (2022). Rethinking the role of demonstrations: What makes in-context learning work? In Y. Goldberg, Z. Kozareva, & Y. Zhang (Eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing* (pp. 11048–11064). Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.emnlp-main.759

Park, S., Shin, S., Lee, B., Lee, J., Surh, J., Seo, M., & Lee, H. (2019). *CORD: A consolidated receipt dataset for post-OCR parsing* [Paper presentation]. Document Intelligence Workshop at the Conference on Neural Information Processing Systems (NeurIPS 2019). https://openreview.net/forum?id=SJl3z659UH

Perot, V., Kang, K., Luisier, F., Su, G., Sun, X., Boppana, R. S., Wang, Z., Wang, Z., Mu, J., Zhang, H., Lee, C.-Y., & Hua, N. (2024). LMDX: Language model-based document information extraction and localization. In L.-W. Ku, A. Martins, & V. Srikumar (Eds.), *Findings of the Association for Computational Linguistics: ACL 2024* (pp. 15140–15168). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.findings-acl.899

Sclar, M., Choi, Y., Tsvetkov, Y., & Suhr, A. (2024). Quantifying language models' sensitivity to spurious features in prompt design or: How I learned to start worrying about prompt formatting. In *The Twelfth International Conference on Learning Representations (ICLR 2024)*. https://openreview.net/forum?id=RIu5lyNXjT

Wen, B., Yao, J., Feng, S., Xu, C., Tsvetkov, Y., Howe, B., & Wang, L. L. (2025). Know your limits: A survey of abstention in large language models. *Transactions of the Association for Computational Linguistics, 13*, 529–556. https://doi.org/10.1162/tacl_a_00754

Cite as: Wilson, L. (2026). Does prompt design stop AI models inventing invoice data?: A pilot test of four prompts on two Claude models (Version 1.0) [Pilot study, preprint]. https://lewis-wilson.com/research/prompt-design-extraction
Licence: © 2026 Lewis Wilson. Licensed under CC BY 4.0 (https://creativecommons.org/licenses/by/4.0/).