

A Practitioner’s Guide to Systematic Prompt Evaluation for Information Extraction

Anna Steinberg Schulten^{1,5,*}, Malte Schierholz^{1,5}, Lisa Reichenbach²,
Frauke Kreuter^{1,4,5}, Alexander Fraser^{3,5}

¹Ludwigs-Maximilians-Universität München (LMU Munich),

²Deutsche Bundesbank, ³Technische Universität München (TUM)

⁴University of Maryland, ⁵Munich Center for Machine Learning (MCML)

Abstract

Practitioners across industries increasingly use large language models to extract information from unstructured documents such as financial filings, legal contracts, and sustainability reports, yet typically rely on generic online recommendations for prompt design that lack empirical grounding in their target setting. We address this gap with a practical methodology for systematically varying individual prompt components (prompt structure, contextual information, and model selection) and measuring their effect on extraction quality. We use greenhouse gas emissions extraction from corporate sustainability reports as a case study of fixed numeric extraction from high-stakes PDF documents that vary widely in structure and presentation and are dispersed across company websites. Instantiated in a retrieval-augmented generation pipeline, our evaluation shows that best practices from industry-facing prompting guides do not always hold: a structured prompt incorporating roles, definitions, extraction rules, and JSON formatting unexpectedly underperforms a simpler question-style prompt, while the benefit of adding tabular context proves both prompt- and model-specific. We condense our evaluation procedure and findings into a practitioner’s decision guide for systematically testing design choices on task-specific data rather than adopting generic recommendations by default.

1 Introduction

Across industries, organisations are increasingly using large language models to automate information extraction from unstructured documents from financial filings and legal contracts to medical records and sustainability reports, often building production pipelines with limited internal expertise in prompt engineering. When designing such pipelines, practitioners typically turn to publicly

available recommendations to inform their prompt design. According to the top search results for "prompt elements"¹, a typical prompt includes a role, a task, a context, and a format, with examples sometimes provided as a separate component.

However, these recommendations are rarely backed by systematic empirical evaluation in the practitioner’s target setting. Most research papers either only report the best-performing prompt for a specific application, such as the extraction of business process model information (see e.g. [Neuberger et al., 2025](#)) without describing the prompt development process, or they evaluate a particular prompting technique, like prompt chaining, focusing on a single component of the prompt ([Kwak et al., 2024](#)). This reflects a broader gap in the field: while a recent analysis of nearly 300 papers from top software engineering conferences found that 69.58% report the exact prompt used, only 46.5% mention that they refined or tuned their prompts during the research process, and only 44.1% compare different prompt variations and report their performance ([Korn et al., 2026](#)).

This paper presents a practical guide for practitioners seeking to systematically evaluate prompt design choices for their own use case. We use climate data extraction as a running example of a common practitioner task: extracting standardized numerical information from long, heterogeneous documents, as is also common in financial and regulatory reporting. Consistent and reliable data on company-level greenhouse gas (GHG) emissions is essential for effective governance of the climate crisis ([Network for Greening the Financial System, 2024](#)). In practice, however, companies report their GHG emissions in non-standardized, lengthy PDF documents that vary in size and structure due to the lack of a mandatory reporting format, and publish

*Corresponding author: anna.steinberg@lmu.de

¹Example websites: [Prompting Guide from DAIR.AI](#); [Learn Prompting](#)

them on their individual websites rather than to a centralized repository.

Concretely, we provide a methodology for systematically varying individual prompt components (prompt structure, contextual information, and model selection) and measuring their effect on extraction quality. We instantiate this in a retrieval-augmented generation (RAG) pipeline (see Figure 1), similar to Mankour et al. (2025) and Forster et al. (2026), but focused on the generative component. While the specific performance effects we estimate are tied to our GHG extraction setting, the evaluation procedure is designed to transfer to other extraction tasks by testing these choices on task-specific data. Our findings illustrate why systematic evaluation matters: performance does not improve monotonically across model generations, open-source models lag behind closed-source ones, a structured prompt with common best practices (roles, definitions, extraction rules, JSON formatting) unexpectedly loses in performance to a simple question-styled prompt. The benefit of tabular context is narrow: it improves only the question prompt in one model setting. Established best practices thus do not always hold under systematic evaluation.

To make these insights actionable, we condense our learnings into a practitioner’s handbook with concrete recommendations and release our pipeline as a Python package².

2 Related Work

We discuss related work along three lines: prompt engineering in practice, prompting for information extraction (IE), and tabular context in IE.

Prompt engineering in practice. Prompt engineering is extensively discussed in industry-facing resources such as vendor documentation, online tutorials, and prompt engineering guides, which typically converge on a set of recommended components: role descriptions, task instructions, context, examples, and output formats. These recommendations are widely adopted by practitioners, but are rarely subjected to systematic empirical evaluation in specific deployment settings. The academic literature partly contradicts such guidance: Zheng et al. (2024), for example, find that mentioning personas in prompts is of questionable value, and Tam et al. (2024) suggest that output format

restrictions have detrimental performance in reasoning tasks. More generally speaking, systematic empirical studies of LLM prompt design choices remain scarce. The most relevant analogues to our work are Atreja et al. (2025) and Guo et al. (2024), who systematically vary model selection and design features and consolidate their findings into evidence-based recommendations. Atreja et al. (2025) study LLM annotation compliance in a classification setting, varying definitions, output type, and prompt length, whereas Guo et al. (2024) examine input, output, and reasoning design when constructing samples for fine-tuning. In contrast, we study inference-time prompting for structured numeric information extraction from long, unstructured documents using non-fine-tuned models in a high-stakes practitioner setting. While both studies document substantial variation in the effectiveness of design choices across models and tasks, we complement this work by systematically quantifying these effects and their statistical uncertainty in the information-extraction setting.

A substantial literature has explored automatic prompt optimization techniques (Ramnath et al., 2025). Such techniques may plausibly outperform manually designed prompts, but require labelled training data, a resource often not available to practitioners and therefore not further considered in this guide.

Prompting for IE. In most research papers, the prompt design for information extraction follows the web guidelines by building a prompt up from instructions or a task description, a role or persona, relevant context, and desired output format. The specific design of each of the prompt components can be based on so-called prompt patterns (White et al., 2023). While the relevant context is usually the document passage containing the target information, it can also refer to restrictions guiding the LLM to distinguish relevant from irrelevant data (Neuberger et al., 2025). In the emerging ClimateNLP literature, a common pattern to avoid hallucinations is to rely on retrieval-augmented generation (RAG) pipelines and reformulate extraction tasks as question-answering problems (Schimanski et al., 2024). A recurring finding is that retrieval quality critically shapes extraction accuracy, while attempts to preserve document or tabular structure in embeddings often reduce performance (Mankour et al., 2025). However, most work offers only limited transparency on prompt design, often invoking general "best practices" rather than empirically

²<https://github.com/gist-sustainability/climatextract/releases/tag/v0.3.3>

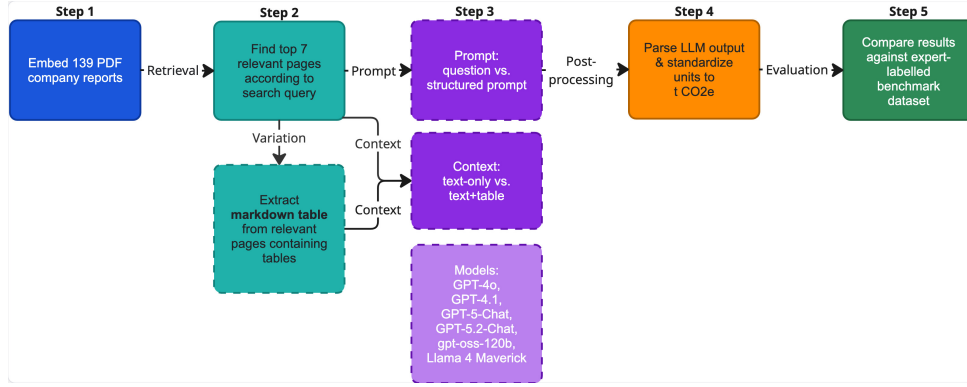


Figure 1: Pipeline for extracting firm-level CO2 emissions showing embedding step, retrieval with optional table extraction, prompting under varying templates, contexts, and model choices, output parsing and evaluating against benchmark dataset. The boxes with dashed contours visualize the variations in the respective processing step.

testing prompt components (Forster et al., 2026).

Tabular context and IE. Companies often report their emissions data in tables within their corporate sustainability reports. The challenge for LLMs when extracting tabular information consists of correctly attributing a value in a raw text embedding to the respective column and row without structural information. To address this issue, a common approach is to rephrase tabular information into a natural language description either to pass as context (Sui et al., 2024) or to generate a question-answer pair (Ali et al., 2025). Hegselmann et al. (2023) study various approaches to verbalize tabular information. In contrast, we process text and table in parallel and follow the recommendation of Yue et al. (2025) for a format which preserves "complete semantic information without excessive structural details" by transforming the table from raw text into a Markdown table with Docling (Auer et al., 2024), keeping the tabular structure intact. There also have been examples of open-weight models fine-tuned for table question-answering (Zhu et al., 2024). In these experiments, the goal is a complex table-reasoning task rather than a simple information extraction task. Consequently, we focus on the prompting component.

3 Method

3.1 Target information: Emission scopes

We seek to extract the respective company’s greenhouse gas emission values from their unstructured corporate sustainability and annual reports, provided for 4 different operational boundaries, called *Scopes*, as specified by the reporting guidelines of the Greenhouse Gas Protocol (GHGP) (WRI and WBCSD, 2011). Scope 1 emissions refer to

direct GHG emissions from company-owned or -controlled sources. Scope 2 covers indirect GHG emissions from the purchase and use of electricity, disclosed according to either location- or market-based methods (Scope 2 lb and Scope 2 mb, respectively). Finally, Scope 3 emissions include 15 diverse categories of residual up- and downstream value-chain emissions. The standard unit for all disclosed emissions is CO2 equivalents (CO2e).

3.2 Pipeline

We use a RAG-based pipeline which proceeds in 5 steps as displayed by Figure 1. In the first step, we embed the entire document using OpenAI’s text-embedding-ada-002 (Greene et al., 2022) and then identify relevant context by semantically searching for pages most relevant to the query "What are the total CO2 emissions in different years? Include Scope 1, Scope 2, and Scope 3 emissions if available." We then consider the top 7 pages in terms of cosine similarity to the query as relevant context for the prompting step³. In Step 3, we pass a task-tailored prompt to a large language model where we vary the prompt structure and context type (dark purple boxes with dashed borders) as well as the model (light-purple box). Post-processing in Step 4 involves parsing of the LLM output, unit normalization using a unit dictionary and removal of duplicates according to custom prioritization rules (see Appendix A.2 for details). For evaluation in the last step 5, we rely on a benchmark, expert-annotated dataset of sustainability information from a sample of 139 sustainability

³Prior empirical evaluations (see Appendix A.1) indicated that the number of retrieved pages set to 7 represents the best trade-off between accuracy and computational cost for our use case.

and annual reports covering the year range 2017 to 2022, where each entry represents a combination of report, year and scope. [Beck et al. \(2025\)](#) document the benchmark’s construction and describe the annotation protocol. The sector- and region-diverse set of companies authoring the reports are a random selection of around 4000 companies featured in two major indices (international "MSCI World Small Cap" and German "DAX"). Comparing the extracted values to the ground truth values from the benchmark dataset, we compute report-level precision, recall and F1-score. To assess the statistical significance of the performance differences between the experiments and the baseline, we perform fixed-effect regressions and compute one-sided p-values (more details in Appendix A.5).

4 Experimental setup

When deploying an extraction pipeline, practitioners are rarely in a position to redesign the underlying architecture; instead, they work with a small number of configurable components that are exposed by most off-the-shelf or plug-and-play systems. We therefore structure our investigation around the three levers most readily available in such settings: the choice of model, the design of the prompt, and the provision of contextual information. We evaluate how each component affects extraction quality for sustainability information extraction. Like most real-world extraction tasks, our use case involves specific constraints on what counts as a valid value. Following the benchmark’s annotation schema, we extract disclosed values only if they (i) refer to the whole company, (ii) are reported according to the operational boundaries of the scopes, (iii) are reported in absolute terms as CO2 or CO2e, and (iv) represent a total value rather than subcategories.

4.1 Baseline

For our baseline, we prompt a large language model with a question-answering inspired prompt, called *question prompt* (see Figure A1 in the appendix), and the relevant text context from the previous retrieval step. The prompt states the task at the beginning, includes an instruction for the case of no values found and then lists all combinations of 4 metrics (Scope 1, 2mb, 2lb and 3) and 11 years of interest as questions. The resulting 44 questions are followed by the context and an example of the preferred response format, constructed by reiterat-

ing the first 3 questions and using placeholders. To extract the numerical values from the string output, we use regular expressions in pipeline Step 4.

4.2 Model experiments

Practitioners commonly assume that newer and larger models deliver better extraction quality, and often default to the latest closed-source release. To test this assumption and assess the underlying cost-performance trade-off, we evaluate the latest closed-source models (GPT-4o ([OpenAI, 2024](#)), GPT-4.1 ([OpenAI, 2025b](#)), GPT-5 ([Singh et al., 2025](#)) and GPT-5.2 ([OpenAI, 2025c](#)) against two open-weight models, gpt-oss-120b ([OpenAI, 2025a](#)) and Llama 4 ([Meta AI, 2025](#)). We focus on OpenAI models due to their deployment prevalence at the time of our experiments (May 2026). OpenAI models accounted for roughly two thirds of the AI assistant market, well ahead of Gemini and Claude ([Pohrebniyak, 2026](#)). We consider deployment prevalence particularly relevant given our focus on practitioners building applications with widely adopted APIs. Since the task is pure information extraction and does not require extensive reasoning, we focus on the non-reasoning variants of GPT-5 and GPT-5.2 and do not consider dedicated reasoning models. The exception is gpt-oss-120b, OpenAI’s first open-weight model, which we include to enable an open- versus closed-weight comparison within the OpenAI model family. Although gpt-oss-120b is classified as a reasoning model, we do not include it as representative of dedicated reasoning models, and its results should therefore not be interpreted as a comparison between reasoning and non-reasoning models. Extrapolating from previously published model sizes (GPT-3:175B params), we assume that the closed-source models are increasing in parameter size and capabilities and surpassing at least the gpt-oss-120b (117B params). Thus, we expect newer models to outperform older ones and larger models to perform better than smaller ones (Hypotheses $H_{1,j}^{\text{model}}$ where $j \in \{\text{gpt-4o}, \text{gpt-4.1}, \text{gpt-5}, \text{gpt-oss}, \text{Llama-4}\}$).

4.3 Prompt experiments

A widely recommended prompt structure includes role descriptions, explicit instructions, definitions, extraction rules, and an output schema. To test whether this structure improves extraction quality, we create the *structured prompt* (see Figure A2 in appendix) by defining a role for the LLM, providing clear instructions, generating one-sentence

long definitions of the scopes and listing the extraction rules reflecting the relevance criteria described above. For instance, the model is supposed to extract only absolute values, not percentage increases or emission intensities. Furthermore, to enable structured parsing, we include an explicit output schema in the prompt, defined as a Pydantic model. Although LLMs have strong in-context learning capabilities (Brown et al., 2020; Dong et al., 2024), we deliberately exclude few-shot examples. In our setting, a meaningful example would require providing both the relevant context from another report and its correct extraction, substantially increasing the length of an already extensive prompt. Moreover, given the substantial structural and formatting heterogeneity across reports, no single example is representative of the corpus, while a fixed example may anchor extraction to a particular report format. We therefore focus on instruction-based prompting and expect the structured prompt to outperform the question prompt thanks to the provided extraction rules (H_1^{prompt}). Moreover, we perform ablation studies to evaluate the individual component’s contribution and influence of structured parsing.

4.4 Tabular context experiments

Preserving table structure during preprocessing is a common design choice when documents contain critical information in tables. The raw text extracted from PDF documents often does not capture the structure of tables, leading to potential misattribution of values to incorrect years (e.g., a value for 2015 might be linked to 2016). Given that 85% of our ground-truth values are entered in tables, we follow Ali et al. (2025) to process these separately by checking if relevant pages contain tables, converting them to markdown format to preserve structure, and then providing this as additional context to the LLM. We hypothesize this approach will enhance our pipeline’s performance by resolving year misalignments (H_1^{context}).

5 Results

5.1 Model experiments

Table 1 shows that GPT-5.2 achieves the strongest performance overall, with statistically significant gains over GPT-5, Llama-4, and gpt-oss (leading us to reject the null only for those models; see Table A3 for details). Differences between the non-reasoning GPT models are smaller and not consistently significant, indicating that newer generations

Model	Created	Precision	Recall	F1 Score
gpt-4o	11/24	0.656 (0.007) [0.646, 0.662]	0.929 (0.005) [0.922, 0.935]	0.769 (0.006) [0.761, 0.775]
gpt-4.1	04/25	0.658 (0.017) [0.642, 0.684]	0.944 (0.017) [0.916, 0.958]	0.775 (0.009) [0.764, 0.783]
gpt-5	08/25	0.644 (0.009) [0.633, 0.654]	0.950 (0.008) [0.942, 0.960]	0.768 (0.007) [0.759, 0.775]
gpt-5.2	12/25	0.779 (0.015) [0.763, 0.800]	0.878 (0.084) [0.731, 0.936]	0.823 (0.039) [0.754, 0.850]
Llama-4	04/25	0.182 (0.034) [0.129, 0.217]	0.817 (0.159) [0.543, 0.927]	0.293 (0.043) [0.223, 0.329]
gpt-oss	08/25	0.812 (0.040) [0.760, 0.868]	0.281 (0.042) [0.238, 0.341]	0.416 (0.047) [0.370, 0.479]

Table 1: Averaged results of 5 runs of model experiments, accompanied by standard deviation and range. The closed-source models show a nonmonotonic increase in performance (gpt-4.1 better than newer gpt-5, newest gpt-5.2 leaping ahead). The open-weight models lag behind the closed-source models.

of these models do not uniformly outperform older ones: GPT-4.1 outperforms GPT-4o on all three metrics, but GPT-5 falls behind GPT-4.1 on precision and F1 despite achieving the highest recall. The two open-weight models also lag behind the closed-source GPT models in our end-to-end evaluation, although gpt-oss-120b is not strictly comparable because it is classified as a reasoning model. Moreover, an error decomposition reveals different underlying causes of the observed gaps. For gpt-oss-120b, 85.8% of errors are parsing failures, indicating that its measured underperformance largely reflects incompatibility with our parsing pipeline. In contrast, Llama-4’s errors predominantly reflect incorrect extractions (72.9%). Despite having more than twice as many parameters as gpt-oss-120b (400B vs. 117B), Llama-4 exhibits highly unstable recall and overall performs worse than gpt-oss-120b.

Takeaway. The performance of the open-weight models in our setting depends partly on their compatibility with the downstream parsing pipeline. Among the non-reasoning GPT models, model version is an imperfect proxy for extraction quality. Benchmarking candidate models together with the intended parsing approach on a representative subset of the specific extraction task is therefore advisable before committing to a default choice.

5.2 Prompt experiments

We run the prompt experiments with our preferred model GPT-5.2. The results are given in Table 2. Comparing only the two prompts (row 1 and 3), the question prompt surprisingly outperforms the

Prompt Type	Input Mode	Precision	Recall	F1 Score
question	text	0.779 (0.015) [0.763, 0.800]	0.878 (0.084) [0.731, 0.936]	0.823 (0.039) [0.754, 0.850]
question	text+table	0.777 (0.021) [0.742, 0.792]	0.903 (0.014) [0.887, 0.922]	0.835 (0.017) [0.808, 0.852]
structured	text	0.760 (0.008) [0.752, 0.771]	0.860 (0.011) [0.850, 0.878]	0.807 (0.006) [0.801, 0.815]
structured	text+table	0.755 (0.020) [0.722, 0.776]	0.830 (0.105) [0.643, 0.894]	0.787 (0.049) [0.703, 0.822]

Table 2: Averaged results of 5 runs of prompt and context experiments with gpt-5.2, accompanied by standard deviation and range. A simple question-style prompt outperforms the structured prompt. Adding tabular context improves results for question prompt slightly, but not consistently for structured prompt.

Prompt	Precision	Recall	F1	$\Delta F1$
Full structured	0.760 (0.008)	0.860 (0.011)	0.807 (0.006)	—
– role	0.823 (0.008)	0.819 (0.013)	0.821 (0.006)	+0.014
– definitions	0.790 (0.031)	0.827 (0.013)	0.808 (0.017)	+0.001
– rules	0.634 (0.045)	0.726 (0.026)	0.676 (0.030)	-0.131

Table 3: Averages and standard deviations of 5 runs of ablations of structured prompt components, ordered by effect on F1. Removing extraction rules causes a substantial performance drop, while removing the role slightly improves performance and removing definitions has no measurable effect.

structured prompt in both precision and recall and thus also F1-score. The negative coefficient on the prompt dummy is statistically significant ($p < 0.05$; see Table A3), so H_1^{prompt} is not supported.

To better understand which elements of the structured prompt drive this outcome, we ablate its individual components (Table 3). The contributions are notably uneven: removing the role description slightly increases F1 from 0.807 to 0.821, removing the scope definitions leaves performance essentially unchanged (0.808), and removing the extraction rules causes a substantial drop to 0.676. In other words, the extraction rules carry almost all of the structured prompt’s value, while the role description appears mildly counterproductive and the definitions have no impact. The positive impact of extraction rules is particularly interesting, since qualitative analysis of extracted values reveals that some extraction rules are still ignored in some instances⁴. Notably, even the best performing ablation variant has an F1 score just as good as the question prompt. A naive adaption of the recommended full structured prompt bundle would underperform. Only with the extra work of the ablation higher performance was recovered.

⁴For instance, the pipeline extracts a value for a subcategory of Scope 3 instead of the total.

To rule out that the prompt comparison is confounded by output format, we additionally test the question prompt with JSON output (Table A2 in Appendix). Its F1 of 0.794 sits slightly below both the original question prompt (0.823) and the structured prompt (0.807), indicating that the output format does not explain the observed performance gap and, if anything, modestly hurts the question prompt.

Takeaway. Best-practice prompt components do not contribute uniformly, and adding more structure, whether through additional instruction components or stricter output formats such as JSON, does not reliably improve performance. Inspecting each component’s individual contribution rather than adopting recommended templates as a bundle can reveal which elements are doing the work and which are inert or counterproductive in a given setting.

5.3 Tabular context experiments

A common assumption is that preserving table structure during preprocessing improves extraction quality for documents with table-heavy content. Comparing rows 1 and 2 in Table 2 shows that adding the Markdown table as context leads to a slight improvement in F1 for the question prompt, driven by better recall. This difference is statistically significant at the 1% level (see Table A3). For the structured prompt, however, adding the tabular context decreases overall performance through both lower precision and lower recall. As a robustness check, we repeat the experiments with the second-best model, GPT-4.1 (see Table A4 in the appendix), where adding tabular context decreases performance for both prompts. The effect of tabular context is thus both model- and prompt-dependent. We note that the Markdown conversion process is identical across both prompt conditions, so any conversion errors should affect them equally and cannot explain the divergent results.

To better understand when tabular context helps, we manually inspect the 28 reports with values in tables that had erroneous extractions in the text-only setting. Only 6 of these contain misalignment between extracted values and their attributed years or scopes. In the text+table setting, the additional context resolves such errors for 2 of these 6 reports. The remaining 22 reports exhibit other error patterns, such as irrelevant extractions or footnote misinterpretations, which adding tabular structure cannot fix. Given the limited prevalence of mis-

alignment errors and the dominance of other failure modes, we cannot conclude that providing tabular context is universally beneficial.

Takeaway. The benefit of providing tabular context is not universal but depends on the interaction between model, prompt, and the dominant error patterns in the data. Before investing in table pre-processing, it is worth diagnosing whether year or scope misalignment is the main source of errors; if other failure modes dominate, structural pre-processing is unlikely to improve results.

5.4 Cost-benefit analysis

Beyond accuracy, token usage is a meaningful factor in practitioner deployment decisions, as it affects both computational requirements and, conditional on provider-specific pricing, monetary costs⁵.

To assess the advantage of the newest model, we fix the question prompt and text-only context and compare gpt-5.2 against gpt-4.1 in Table A5. The F1 improvement requires essentially no additional tokens: both models use approximately 1.47 million prompt tokens and 0.81 million output tokens. The 46% increase in monetary cost therefore reflects differences in provider pricing rather than greater token usage, while the improvement itself is not statistically significant ($H_{1,\text{gpt-4.1}}^{\text{model}}$, see Table A3).

Comparing the two prompt types with text-only context, the question prompt uses about 36% more tokens overall than the structured prompt, despite using about 9% fewer prompt tokens. This additional token usage yields an estimated F1 advantage of only 0.0085 ($p < 0.05$, see Table A3). Despite being statistically significant, this effect size is small enough that the more token-efficient structured prompt may be preferable in cost-sensitive deployments. The difference in token usage stems almost entirely from output length: the question prompt reiterates all 44 scope-year combinations in its response, producing roughly 15 times more output tokens than the structured prompt’s JSON format. Adding tabular context increases total token usage only moderately (by about 2% for the question prompt and 5% for the structured prompt) and changes F1 in opposite directions, improving the question prompt and reducing the structured prompt’s performance.

⁵The figures reported refer to averages over five runs of 139 reports per setting.

Takeaway. Token usage across design choices can differ substantially and can be disproportionate to the corresponding performance gains. When the expected gain is small or uncertain, the more token-efficient configuration is often the more rational choice.

6 Conclusion

We present a practical methodology for systematically evaluating prompt design choices in information extraction pipelines, using greenhouse gas emissions extraction from corporate sustainability reports as a running example. By varying model selection, prompt structure, and context provision in a controlled RAG pipeline, we find that several commonly recommended practices do not consistently improve performance in our setting: improvements between adjacent GPT generations are small and not always statistically significant, a structured prompt incorporating role descriptions, definitions, and extraction rules underperforms a simpler question-style prompt, and adding tabular context helps only in specific model-prompt combinations. A cost-benefit analysis further reveals that performance gains and token usage are not always proportional. These findings apply directly to our setting of fixed-schema numeric extraction from heterogeneous corporate reports and should not be interpreted as universal effects across IE tasks. The transfer to other tasks and domains is untested and should be verified.

Our broader contribution is the evaluation procedure rather than a universally optimal configuration. We consolidate this procedure into a practitioner’s decision guide (Figure A3), emphasizing that model, prompt and context choices should be tested empirically on representative task-specific data rather than adopted by default. We expect this procedure to be particularly applicable to similarly structured tasks, such as extracting standardized numeric indicators from financial or regulatory documents, even where the optimal configuration itself may differ.

Limitations

We briefly discuss some limitations of this work. The benchmark chosen is intentionally curated for quality and focuses on English-language data, which may limit direct generalization to larger or multilingual settings. Only numeric extraction was considered. Corporate GHG emissions are often

reported as unstructured tables or even as diagrams embedded inside PDF files. While other IE tasks have separate challenges, the selected extraction method will always depend on the source format. At the same time, the dataset spans a wide range of document structures and report lengths, reflecting the heterogeneity commonly found in real-world corporate reporting. Nevertheless, the transferability of our empirical findings to other information extraction tasks remains to be established.

More generally, the space of possible design choices for LLM-based information extraction, including prompt design, context type, and model selection, is inherently large. Our experiments therefore focus on a principled but necessarily selective subset of this space, and results may not directly transfer to all alternative configurations. In particular, future research could examine whether few-shot examples improve extraction performance despite the associated prompt-length costs and risk of anchoring to specific document formats. Automatic prompt optimization techniques promise a solution to optimize over larger subspaces iteratively, though this would increase cost.

This study makes use of tabular context by converting it to Markdown format. Future research could examine if tables should be verbalized as an additional step, where each cell is represented as a self-contained statement with entity, metric, year, value, and unit.

Finally, our model selection is intentionally narrow: we evaluate the closed-source GPT family because of its industry dominance⁶, and the open-weight gpt-oss and Llama-4 models as well-established representatives of the open-weight landscape at the time of writing. Other models may exhibit different performance characteristics, further reinforcing the need to validate model choices on representative task-specific data.

Use of AI Assistants

The authors used large language model assistants (Claude) to support the writing and revision process. Specifically, AI assistants were used to suggest alternative phrasings, improve sentence-level clarity and explore structural options for the presentation of results. All conceptual contributions, experimental design, data analysis, and final wording decisions were made by the authors. The authors

reviewed and verified all AI-assisted output before incorporation into the paper.

References

- Mohammed Ali, Abdelrahman Abdallah, and Adam Jatowt. 2025. [SustainableQA: A Comprehensive Question Answering Dataset for Corporate Sustainability and EU Taxonomy Reporting](#). *ArXiv*, abs/2508.03000.
- Shubham Atreja, Joshua Ashkinaze, Lingyao Li, Julia Mendelsohn, and Libby Hemphill. 2025. [What’s in a prompt?: A large-scale experiment to assess the impact of prompt design on the compliance and accuracy of llm-generated text annotations](#). *Proceedings of the International AAAI Conference on Web and Social Media*, 19:122–145.
- Christoph Auer, Maksym Lysak, Ahmed Nassar, Michele Dolfi, Nikolaos Livathinos, Panos Vagenas, Cesar Berrospi Ramis, Matteo Omenetti, Fabian Lindlbauer, Kasper Dinkla, Lokesh Mishra, Yusik Kim, Shubham Gupta, Rafael Teixeira de Lima, Valery Weber, Lucas Morin, Ingmar Meijer, Viktor Kuropiatnyk, and Peter W. J. Staar. 2024. [Docling technical report](#). Technical report, AI4K Group, IBM Research.
- Jacob Beck, Anna Steinberg, Andreas Dimmelmeier, Laia Domenech Burin, Emily Kormanyos, Maurice Fehr, and Malte Schierholz. 2025. [Addressing data gaps in sustainability reporting: A benchmark dataset for greenhouse gas emission extraction](#). *Scientific Data*, 12(1):1497.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. 2024. [A survey on in-context learning](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1128, Miami, Florida, USA. Association for Computational Linguistics.
- Kerstin Forster, Lucas Keil, Victor Wagner, Maximilian A. Müller, Thorsten Sellhorn, and Stefan Feuerriegel. 2026. [Assessing corporate sustainability with large language models: Evidence from Europe](#). *Nature Communications*, 17(1):5940.
- R. Greene, T. Sanders, L. Weng, and A. Neelakantan. 2022. [New and improved embedding model](#). Accessed: 2026-01-22.

⁶60.6% market share compared to 15.1% for Gemini and 5% for Claude as of May 2026 (Pohrebniyak, 2026)

- Biyang Guo, He Wang, Wenyilin Xiao, Hong Chen, ZhuXin Lee, Songqiao Han, and Hailiang Huang. 2024. [Sample design engineering: An empirical study on designing better fine-tuning samples for information extraction with LLMs](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 573–594, Miami, Florida, US. Association for Computational Linguistics.
- Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. 2023. [Tabllm: Few-shot classification of tabular data with large language models](#). In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 5549–5581. PMLR.
- Alexander Korn, Lea Zaruchas, Chetan Arora, Andreas Metzger, Sven Smolka, Fanyu Wang, and Andreas Vogelsang. 2026. [Reporting llm prompting in automated software engineering: A guideline based on current practices and expectations](#). In *Proceedings of the 2026 IEEE/ACM Third International Conference on AI Foundation Models and Software Engineering, FORGE '26*, page 184–194, New York, NY, USA. Association for Computing Machinery.
- Alice Kwak, Clayton Morrison, Derek Bambauer, and Mihai Surdeanu. 2024. [Classify first, and then extract: Prompt chaining technique for information extraction](#). In *Proceedings of the Natural Legal Language Processing Workshop 2024*, pages 303–317, Miami, FL, USA. Association for Computational Linguistics.
- Rafik Mankour, Yassine Chafai, Hamada Saleh, Ghasen Ben Hassine, Thibaud Barreau, and Peter Tankov. 2025. [Climate finance bench](#). *Preprint*, arXiv:2505.22752.
- Meta AI. 2025. [LLaMA 4 Model Card](#). Accessed: 2026-02-09.
- Network for Greening the Financial System. 2024. [Information note: Improving greenhouse gas emissions data](#). Accessed: 2023-10-12.
- Julian Neuberger, Lars Ackermann, Han van der Aa, and Stefan Jablonski. 2025. [A universal prompting strategy for extracting process model information from natural language text using large language models](#). In *Proceedings of the 43rd International Conference on Conceptual Modeling, ER 2024*, pages 38–55, Cham. Springer Nature Switzerland.
- OpenAI. 2024. [Gpt-4o system card](#). *Preprint*, arXiv:2410.21276.
- OpenAI. 2025a. [gpt-oss-120b & gpt-oss-20b model card](#). *Preprint*, arXiv:2508.10925.
- OpenAI. 2025b. [Introducing gpt-4.1 model family](#). Accessed: 2026-01-22.
- OpenAI. 2025c. [Update to gpt-5 system card: Gpt-5.2](#). Accessed: 2026-01-22.
- Ivan Pohrebniyak. 2026. [100+ eye-opening chatgpt statistics: Tracing the roots of generative ai to its global dominance](#). Master of Code Global. Accessed: 2026-08-25.
- Kiran Ramnath, Kang Zhou, Sheng Guan, Soumya Smruti Mishra, Xuan Qi, Zhengyuan Shen, Shuai Wang, Sangmin Woo, Sullam Jeoung, Yawei Wang, Haozhu Wang, Han Ding, Yuzhe Lu, Zhichao Xu, Yun Zhou, Balasubramaniam Srinivasan, Qiaojing Yan, Yueyan Chen, Haibo Ding, and 2 others. 2025. [A systematic survey of automatic prompt optimization techniques](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 33078–33110, Suzhou, China. Association for Computational Linguistics.
- Tobias Schimanski, Jingwei Ni, Roberto Spacey Martín, Nicola Ranger, and Markus Leippold. 2024. [ClimRetrieve: A benchmarking dataset for information retrieval from corporate climate disclosures](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17509–17524, Miami, Florida, USA. Association for Computational Linguistics.
- Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, Akshay Nathan, Alan Luo, Alec Helyar, Aleksander Madry, Aleksandr Efremov, Aleksandra Spyra, Alex Baker-Whitcomb, Alex Beutel, Alex Karpenko, and 465 others. 2025. [Openai gpt-5 system card](#). *Preprint*, arXiv:2601.03267.
- Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. 2024. [Table meets llm: Can large language models understand structured table data? a benchmark and empirical study](#). In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining, WSDM '24*, page 645–654, New York, NY, USA. Association for Computing Machinery.
- Zhi Rui Tam, Cheng-Kuang Wu, Yi-Lin Tsai, Chieh-Yen Lin, Hung-yi Lee, and Yun-Nung Chen. 2024. [Let me speak freely? a study on the impact of format restrictions on large language model performance](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1218–1236, Miami, Florida, US. Association for Computational Linguistics.
- Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C. Schmidt. 2023. [A prompt pattern catalog to enhance prompt engineering with chatgpt](#). In *Proceedings of the 30th Conference on Pattern Languages of Programs, PLoP '23*, USA. The Hillside Group.

WRI and WBCSD. 2011. [The greenhouse gas protocol: A corporate reporting and accounting standard - revised version](#). Accessed: 2026-01-15.

Chongjian Yue, Xinrun Xu, Xiaojun Ma, Lun Du, Zhiming Ding, Shi Han, Dongmei Zhang, and Qi Zhang. 2025. [Extract information from hybrid long documents leveraging llms: A framework and dataset](#). In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Mingqian Zheng, Jiaxin Pei, Lajanugen Logeswaran, Moontae Lee, and David Jurgens. 2024. [When “a helpful assistant” is not really helpful: Personas in system prompts do not improve performances of large language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15126–15154, Miami, Florida, USA. Association for Computational Linguistics.

Fengbin Zhu, Ziyang Liu, Fuli Feng, Chao Wang, Moxin Li, and Tat Seng Chua. 2024. [Tat-llm: A specialized language model for discrete reasoning over financial tabular and textual data](#). In *Proceedings of the 5th ACM International Conference on AI in Finance, ICAIF ’24*, page 310–318, New York, NY, USA. Association for Computing Machinery.

A Appendix

A.1 Retrieval experiments

Similarity k	Precision	Recall	F1 Score	True Positives	Cost (€)	Retrieval Failures
3	0.431	0.530	0.475	262	0.274	14
4	0.432	0.568	0.491	284	0.363	10
5	0.432	0.581	0.495	287	0.451	6
6	0.424	0.583	0.491	299	0.541	2
7	0.431	0.608	0.502	310	0.631	2
8	0.433	0.612	0.507	318	0.719	2
9	0.407	0.601	0.485	300	0.807	2

Table A1: Retrieval experiments, varying the parameter k , the retrieved number of pages most similar to the query.

Through systematic experimentation across 12+ configurations using all 139 files in our dataset and varying number of retrieved pages k and the context window size, retrieving the 7 most similar pages (w.r.t. our query) with a context window of 0 emerged as the optimal parameter combination for retrieval. These experiments were conducted at an earlier point in time and used gpt-4o-mini for consistent LLM performance evaluation. The best configuration achieved the highest recall (60.8%), maintained competitive precision (43.1%), minimised retrieval failures (2 reports), and delivered superior cost-efficiency (€0.631) compared to all tested alternatives. We display the results from the experiments varying only the parameter k in Table A1.

A.2 Pipeline Step 4: Parsing

For the question prompt, we employ a schema of regular expressions to extract the values and units from the string output of the LLM. This strict procedure proved stable until the introduction of gpt-5.2 whose output now includes markdown elements. Thus, an adaptation of the prompt was necessary (see A.3 for more details). For the structured prompt, the LLM output consistently contains a JSON-formatted string output which is directly parsed into a JSON object in Python.

For both prompt types, we then normalise the units to allow for more flexible matching with the benchmark dataset values as well as between-companies comparisons. Since companies use different unit variants for the same unit, e.g. "1,000 t CO2" and "metric kilotons CO2e", we have collected all distinct unit variants in the dataset and normalised them to either "t CO2e", "kt CO2e", "Mt CO2e" or "Other" if the unit is unknown. During parsing, we map the extracted unit to its normalised variant using this dictionary. Additionally, we have saved a factor for each known unit allowing us to standardise all values to "t CO2e", making comparisons in downstream tasks easier.

We need to process multiple extractions for the same scope-year combination, referred to as duplicates, to ensure that we retain only one scope-year combination for each report. To achieve this, we have developed a set of custom prioritization rules that are applied sequentially.

First, we filter out standard duplicates, which are entries with the same value and unit on the same page. Next, if we encounter multiple entries for the same scope-year combination that have different units, we retain the entry recorded in "t CO2e," as this is our preferred unit. Finally, for any entries that represent the same scope-year combination and that share the same units, we select the entry from the majority page among all extractions for that specific report.

A.3 Prompt designs

We consider two prompt types, the question prompt and the structured prompt, for our experiments. The question prompt in Figure A1 starts with a brief task description and an extraction rule concerning Scope 2. To run our pipeline with the gpt-5.2 model, we needed to add the comment in the red dashed box because without this information the model outputs markdown symbols which are

not parsable by our regex schema.

The main body of the question prompt are the questions combining each of the 4 scopes with a year from the requested year range 2013-2023, summing up to 44 questions. To these questions, the context is attached and the first three questions reprinted with a format instruction for the output. The latter is necessary because prior experiments have revealed the model’s tendency to "forget" early questions and this way the focus is re-established.

Extract key pieces of information from this sustainability report.
IMPORTANT: Do NOT use any markdown formatting like **bold, *italic*, or headers. Write CO₂ (not CO₂) when specifying units.**
 If a particular piece of information is not present, output 'Not specified'.
 If the report does not mention if Scope 2 is location-based or market-based, assume it is Scope 2 location-based. Do not extract it as Scope 2 market-based.

Use the following format:

- What is the title
- What are the Scope 1 emissions in 2013
- What are the Scope 1 emissions in 2014
- What are the Scope 1 emissions in 2015
- What are the Scope 1 emissions in 2016
- What are the Scope 1 emissions in 2017
- What are the Scope 1 emissions in 2018
- What are the Scope 1 emissions in 2019
- What are the Scope 1 emissions in 2020
- What are the Scope 1 emissions in 2021
- What are the Scope 1 emissions in 2022
- What are the Scope 1 emissions in 2023
- What are the Scope 2 (market-based) emissions in 2013
- What are the Scope 2 (market-based) emissions in 2014
- What are the Scope 2 (market-based) emissions in 2015
- What are the Scope 2 (market-based) emissions in 2016
- What are the Scope 2 (market-based) emissions in 2017
- What are the Scope 2 (market-based) emissions in 2018
- What are the Scope 2 (market-based) emissions in 2019
- What are the Scope 2 (market-based) emissions in 2020
- What are the Scope 2 (market-based) emissions in 2021
- What are the Scope 2 (market-based) emissions in 2022
- What are the Scope 2 (market-based) emissions in 2023
- What are the Scope 2 (location-based) emissions in 2013
- What are the Scope 2 (location-based) emissions in 2014
- What are the Scope 2 (location-based) emissions in 2015
- What are the Scope 2 (location-based) emissions in 2016
- What are the Scope 2 (location-based) emissions in 2017
- What are the Scope 2 (location-based) emissions in 2018
- What are the Scope 2 (location-based) emissions in 2019
- What are the Scope 2 (location-based) emissions in 2020
- What are the Scope 2 (location-based) emissions in 2021
- What are the Scope 2 (location-based) emissions in 2022
- What are the Scope 2 (location-based) emissions in 2023
- What are the Scope 3 emissions in 2013
- What are the Scope 3 emissions in 2014
- What are the Scope 3 emissions in 2015
- What are the Scope 3 emissions in 2016
- What are the Scope 3 emissions in 2017
- What are the Scope 3 emissions in 2018
- What are the Scope 3 emissions in 2019
- What are the Scope 3 emissions in 2020
- What are the Scope 3 emissions in 2021
- What are the Scope 3 emissions in 2022
- What are the Scope 3 emissions in 2023

Document:
 {context_str}

0. What is the title: Our responsibility. Report {self.min_year}
 1. What are the Scope 1 emissions in 2013: <value> <unit>
 2. What are the Scope 1 emissions in 2014: <value> <unit>

Figure A1: Question prompt template consisting of a question for every scope-year combination in the respective year range 2013-2023 for every scope type Scope 1, 2 (market-based), 2 (location-based) and 3. The text in the red box had to be added to run our pipeline with gpt5.2.

The structured prompt is inspired by widespread best practices and states the role and objective at the beginning, followed by LLM-generated one-sentence definitions of the 4 scopes. The largest chunk of the structured prompt consists of specifications which aim to guide the LLM to distinguish

relevant from irrelevant extractions. The relevant extractions are defined as values which

1. refer to the whole company and
2. are reported according to the operational boundaries of the scopes and
3. are reported in absolute terms as CO₂ or CO₂ equivalents emissions and
4. represent a total value, not subcategories.

Additional to these relevance criteria, further extraction rules referring to any not-intended transformations are detailed in the prompt. Before passing the context, the request year range is specified.

You are a climate analyst tasked with extracting specific absolute numerical data from corporate reports.

Your objective is to extract only the absolute values for the following Key Performance Indicators (KPIs) related to CO₂ emissions across the entire company.

Role & Objective

Scope 1 CO₂ Emissions: Direct GHG emissions from sources owned or controlled by the organization (e.g., fuel combustion, company-owned vehicles).

Scope 2 (market-based) CO₂ Emissions: Indirect GHG emissions from purchased energy, calculated based on energy procurement choices (e.g., renewable energy contracts).

Scope 2 (location-based) CO₂ Emissions: Indirect GHG emissions from purchased energy, calculated using the average emissions intensity of the local electricity grid.

Scope 3 CO₂ Emissions: Indirect GHG emissions from the organization's value chain, both upstream and downstream (e.g., supply chain, business travel, product use).

Only extract values which refer to the whole company.

Only extract absolute values representing total CO₂ emissions (e.g., in tons).

Do not extract any relative values such as percentages, year-over-year changes, or trends. Ignore all values that are expressed as percentages (%) or involve relative comparisons (e.g., increases, decreases, or changes over time).

Footnotes or annotations in metric names should be treated as references and ignored for the extraction process.

Do not modify values based on footnotes, annotations, or any external data sources.

Do not perform any calculations or transformations on the values. Extract and report the data exactly as presented. Do not invert values.

Ensure your extraction only includes absolute values for the defined KPIs, strictly following these guidelines.

If the subcategory "market-based" or "location-based" is not mentioned for a value of Scope 2, always assume that the value refers to the KPI Scope 2 (location-based).

Do not extract all subcategories of Scope 3, but only total values if total values are available for Scope 3.

Only extract value referring to Scope 1, Scope 2 or Scope 3 separately. Do not extract values representing a sum of any of the scopes.

Year range for the search: only extract values from 2013 to 2022.

Here is the excerpt:
 {context_str}

Figure A2: Structured prompt template consisting of role and objective, definitions of scope types, extraction rules and desired year range.

A.4 Ablation of output format

We perform an ablation study with respect to the output format by testing the question prompt with the same JSON output format as the structured prompt. The results are given in Table A2. The JSON output format leads to a decrease in F1 score from 0.823 to 0.794. This indicates that the output format does not explain the performance gap between the question and structured prompts and, if anything, slightly hurts the question prompt’s performance.

Prompt components	Precision	Recall	F1 Score
question prompt	0.779 (0.015)	0.878 (0.084)	0.823 (0.039)
question prompt with JSON output	0.777 (0.031)	0.812 (0.019)	0.794 (0.022)
structured prompt	0.760 (0.008)	0.860 (0.011)	0.807 (0.006)

Table A2: Averages and standard deviations of 5 runs of ablations of prompt output format. The question prompt with JSON output format performs worse than the question prompt with regex and the structured prompt.

Hypothesis (H1)	Directional hypothesis	Coefficient (SE)	Outcome
H_1^{prompt}	$F1_{\text{structured}} > F1_{\text{question}}$	-0.0085** (0.0031)	not supported
H_1^{context}	$F1_{\text{text+table}} > F1_{\text{text}}$	0.0167*** (0.0037)	supported
H_1^{model}	$F1_{\text{gpt-5.2}} > F1_{\text{gpt-4o}}$	0.0064 (0.0045)	not supported
H_1^{model}	$F1_{\text{gpt-5.2}} > F1_{\text{gpt-4.1}}$	0.0048 (0.0046)	not supported
H_1^{model}	$F1_{\text{gpt-5.2}} > F1_{\text{gpt-5}}$	0.0077** (0.0041)	supported
H_1^{model}	$F1_{\text{gpt-5.2}} > F1_{\text{Llama-4}}$	0.1485*** (0.0227)	supported
H_1^{model}	$F1_{\text{gpt-5.2}} > F1_{\text{gpt-oss}}$	0.2713*** (0.0246)	supported

Table A3: One-sided t-tests of directional hypotheses on report-level F1 scores from a fixed-effects regression. Each row reports the coefficient on the dummy variable corresponding to the first condition in the directional hypothesis, with clustered standard errors (ReportName & run_id) in parentheses. Significance: *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

A.5 Statistical testing

We perform fixed-effects regression and compute one-sided p-values on report-level F1 scores of the different configurations to match the directional hypotheses presented in the paper. Report-level averaging is done because we cannot assume that extractions for the same report are independent from each other.

Each row tests an alternative hypothesis H_1 stating that one experimental condition yields a higher F1 score than another, against the null H_0 that the opposite direction holds. The hypotheses are based on best-practice recommendations and the literature review.

We cluster the standard errors at the report and run level. This is reasonable because reports may have different levels of complexity. Further, it is not possible to control the temperature setting of the gpt5.2 model through the API, so runs may have different seeds.

A.6 Second-best model

As a robustness check, we repeat the prompt and context experiments with the second-best model, namely gpt-4.1. The results are given Table A4. In contrast to the results with gpt-5.2, the additional tabular context weakens performance for both prompt types. These findings indicate that gpt-5.2 can make use of the tabular information with the right prompt while gpt-4.1 cannot.

Prompt Type	Input Mode	Precision	Recall	F1 Score
question	text	0.674 (0.040) [0.642, 0.750]	0.953 (0.027) [0.916, 1.000]	0.789 (0.034) [0.764, 0.857]
question	text+table	0.664 (0.095) [0.504, 0.750]	0.922 (0.168) [0.451, 1.000]	0.769 (0.120) [0.476, 0.857]
structured	text	0.582 (0.031) [0.541, 0.624]	0.853 (0.014) [0.838, 0.869]	0.692 (0.021) [0.667, 0.725]
structured	text+table	0.587 (0.020) [0.554, 0.602]	0.836 (0.053) [0.744, 0.872]	0.689 (0.031) [0.635, 0.713]

Table A4: Averaged results of 5 runs of prompt and context experiments with gpt-4.1, accompanied by standard deviation and range. A simple question-style prompt outperforms the structured prompt. Adding tabular context deteriorates results for both prompt types.

A.7 Cost-benefit analysis

Model	F1 Score	Costs (€)	Prompt Tokens	Output Tokens
gpt-4.1	0.775	9.51	1469340	821556.0
gpt-5.2	0.823	13.92	1468518	810563.8

Table A5: Cost-benefit comparison between two best models, gpt-4.1 and gpt-5.2, using the question prompt and text-only context. Spending 46% more on average leads to a 6% boost in performance. Note that the figures presented refer to averages of 5 runs of 139 reports for a given setting.

Prompt Type	Input Mode	F1 Score	Costs (€)	Prompt Tokens	Output Tokens
structured	text	0.807	3.59	1622519	53699.2
structured	text+table	0.787	3.72	1709280	51940.0
question	text	0.823	13.92	1468518	810563.8
question	text+table	0.835	14.70	1684756	839233.4

Table A6: Cost-benefit analysis for gpt-5.2 model for different prompt and context type settings. Using the question-style prompt instead of the structured prompt for text-only context requires 36% more tokens overall, driven by approximately 15 times more output tokens, while yielding a 2% higher F1 score. Note that the figures presented refer to averages of 5 runs of 139 reports for a given setting.

A.8 Practitioner’s decision guide

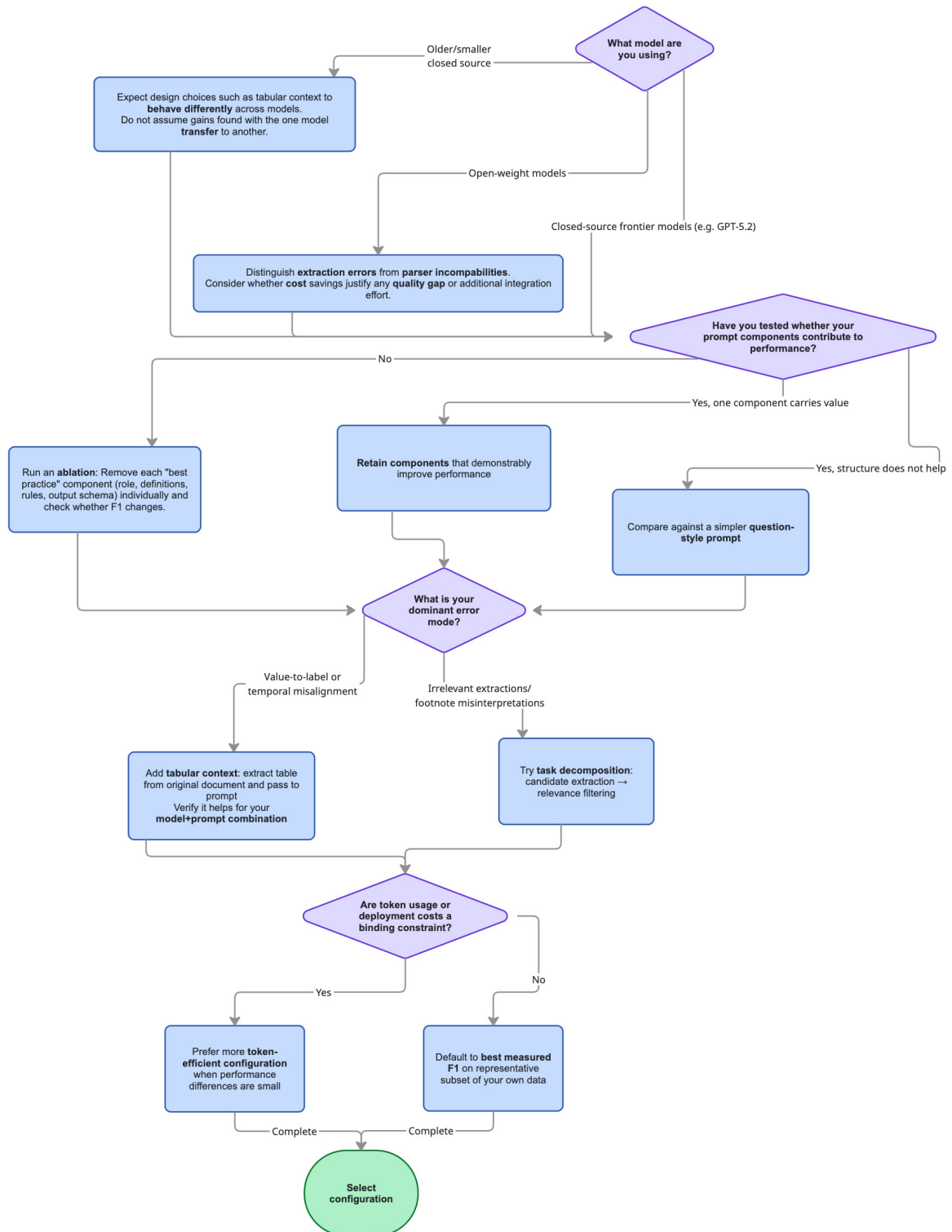


Figure A3: A practitioner's decision guide for systematically evaluating design choices in LLM-based information extraction pipelines. Diagnostic questions (diamonds) translate the case-study findings into evaluation steps (rectangles) to be tested on representative task-specific data, with resource requirements as a cross-cutting consideration.