

Climatextract: A Systematic Prompt Evaluation for Climate Information Extraction

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

¹LMU Munich,²Deutsche Bundesbank, ³Technical University Munich

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

Abstract

Automated extraction of corporate greenhouse gas (GHG) emissions from sustainability reports is critical for climate crisis governance, yet the choice of prompt design, context provision, and model in LLM-based pipelines is largely guided by informal best-practice recommendations rather than systematic evaluation. With **Climateextract**, we assess these choices in a RAG pipeline, comparing closed-source against open-weight models, a structured "best-practice" prompt against a simple question-style prompt, and text-only against text-plus-table contexts. We find that performance is non-monotonic across model generations, a structured prompt with roles, definitions, extraction rules, and JSON formatting is significantly outperformed by a question-style prompt, and tabular context helps only in a specific model-prompt combination. A cost-benefit analysis shows that performance gains are often disproportionate to cost increases.

1 Introduction

Consistent and reliable data on company-level greenhouse gas (GHG) emissions are essential for effective climate policy, financial risk assessment, and sustainability research (Network for Greening the Financial System, 2024). In practice, however, due to a lacking mandatory reporting format, companies typically report their GHG emissions in non-standardized, unstructured, and lengthy PDF documents and upload them to their individual websites, rather than through a centralized repository. Scalable and transparent systems for automatic extraction of these values have thus been increasingly studied by the ClimateNLP community (Schimanski et al., 2024; Mankour et al., 2025; Forster et al., 2025).

Despite widespread deployment of LLMs for this task, the choice of prompt design, context pro-

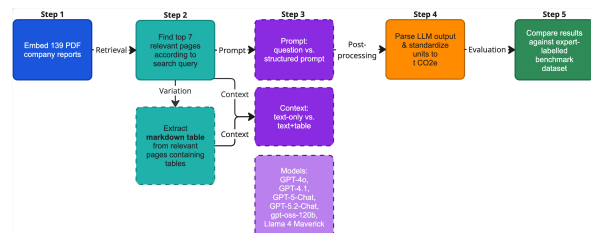


Figure 1: Overview of the **Climatextract** pipeline for extracting firm-level GHG emissions. Report pages are embedded and the most relevant ones retrieved, optionally with table extraction. An LLM is then prompted with the retrieved context, and its output is parsed and evaluated against the benchmark dataset. Dashed boxes mark the components varied in our systematic evaluation (context type, prompt template, model).

vision, and model remains largely guided by informal best-practice recommendations from online websites (e.g. [Prompting Guide from DAIR.AI](#)) and model providers, or by ad hoc decisions reported without systematic evaluation. Prior work either reports only the best-performing prompt for a specific use case (e.g. [Neuberger et al., 2025](#)) or evaluates a single prompting technique like prompt chaining in isolation ([Kwak et al., 2024](#)). This raises the question: do the widely recommended prompt design choices actually improve emissions extraction from corporate sustainability reports?

We address this question with **Climateextract**, a tool to perform systematic evaluation of prompt design choices for LLM-based GHG emissions extraction. Building on a retrieval-augmented generation (RAG) pipeline (Figure 1), similar to Mankour et al. (2025) and Forster et al. (2025), but we focus primarily on the LLM-based generation step (as opposed to retrieval) and assess three factors: model choice (four gpt generations and two open-weight models), prompt structure (a structured prompt following best practices vs. a simple question-style prompt), and context type (text-only vs. text with Markdown-serialized tables).

* Corresponding author: anna.steinberg@lmu.de

Our experiments show that design choices commonly recommended in online prompt-engineering guides, including structured prompt templates, table-aware context, and the newest available model, do not consistently translate into stronger extraction quality. Model release order is an unreliable guide to performance, and the open-weight models we test trail meaningfully behind their closed-source counterparts. A structured prompt built from a role, definitions, extraction rules, and JSON formatting is beaten by a simple question-style variant. The value of adding tabular context turns out to hinge on the specific model and prompt used. Cost differences between configurations are also often out of proportion to the corresponding gains in extraction quality. We release our pipeline as a Python package on PyPI¹ to support systematic prompt evaluation and GHG emission extraction on new climate corpora. All experiments are reproducible from the publicly available benchmark dataset (Beck et al., 2025).

2 Related Work

A common pattern across recent work on LLM-based extraction from sustainability reports is to rely on RAG pipelines and reformulate extraction tasks as question-answering problems, operating primarily on plain-text representations of the underlying PDFs (Ni et al., 2023; Vaghefi et al., 2023; Schimanski et al., 2024; Dimmelmeier 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). Recent work has moved toward large-scale, end-to-end extraction pipelines capable of handling hundreds of indicators, though most offer limited transparency on prompt design, invoking general "best practices" rather than empirically testing prompt components (Forster et al., 2025).

Since companies often report emissions data in tables, table handling is increasingly treated as a dedicated subprocess: Ali et al. (2025), for example, use an LLM to summarize tables into paragraphs before generating question-answer pairs, whereas we retain the tabular structure by converting raw text into Markdown tables and providing them as additional context, following the recommendation of Yue et al. (2025).

The closest 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) vary definitions, output type, and prompt length; Guo et al. (2024) vary input, output, and reasoning design for fine-tuning samples. Their results, however, reveal inconsistencies across models and tasks. We build on this line of work and investigate similar variations for prompt-based emissions extraction from sustainability reports.

3 Method

Our pipeline proceeds in 5 steps as displayed by Figure 1.

Step 1: Document Embedding. The pipeline takes raw PDF reports as input. In the first step, we extract the text of each PDF page via PDFReader and embed each page separately using OpenAI’s text-embedding-ada-002 (Greene et al., 2022). Pages are loaded via PDFReader and treated as atomic chunks, i.e. no sub-page or overlapping chunking is applied.²

Step 2: Retrieval. We 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.". Scope 1, 2 and 3 refer to different operational boundaries as they are specified by the reporting guidelines of the Greenhouse Gas Protocol (GHGP) (WRI and WBCSD, 2011)³. We then consider the top 7 pages in terms of cosine similarity to the query as relevant context for the prompting step. Prior 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. Since the retrieval query and top-k parameter are held constant across configurations, each report yields the same retrieved pages for every downstream comparison. Retrieval nevertheless remains part of the evaluated pipeline

²Pages exceeding the embedding model’s 8,192-token limit are truncated only for the embedding vector; the full page text is passed to the LLM at extraction time if that page is retrieved.

³Scope 1 emissions refer to direct greenhouse gas emissions from sources that are owned or controlled by the company. Scope 2 covers indirect GHG emissions arising from the purchase and use of electricity, disclosed as either according to location- or market-based methods (2lb and 2mb, respectively). Scope 3 emissions cover 15 diverse categories of residual up- and downstream value-chain emissions.

¹<https://pypi.org/project/climatextract/0.3.2/>

because retrieval failures, namely cases where relevant pages are not among the top-k retrieved, do occur for 2 of 139 reports in our benchmark (see Figure A4 in the appendix). Presenting results on pre-computed static context would eliminate this failure mode and inflate performance.

Optional table extraction. The raw text extracted from PDF documents does not capture the tabular layout, leading among other issues 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 and converting the filtered pages to markdown format to preserve structure.

Step 3: Prompting. We vary three factors:

- **Model:** To assess the trade-off between cost and performance, we test recent 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). To minimise stochastic variation, we set temperature to 0 for all models. Since the task is pure information extraction and does not involve logical or mathematical thinking, we consider the chat versions of gpt-5 and gpt-5.2 and do not consider any reasoning models for these experiments, with the exception of gpt-oss-120b where we set `reasoning_effort` to "none" to disable reasoning behaviour.
- **Prompt structure:** We compare the *question prompt* against the *structured prompt*. The *question prompt* (see Figure A1 in the appendix) states the task at the beginning, includes an instruction for the case of no values found and then lists all 44 combinations of 4 metrics (Scope 1, 2mb, 2lb and 3) and 11 years of interest as questions, appends the context and an example of the preferred response format, constructed by reiterating the first 3 questions and using placeholders. Following best practices, the *structured prompt* (see Figure A2 in the appendix) defines a role for the LLM, provides clear instructions, generates one-sentence long definitions of the scopes and lists the extraction rules reflecting the relevance criteria described below. For instance, the model is supposed to extract only absolute

values, not percentage increases or emission intensities.

- **Context type:** We compare passing only textual context of the relevant pages to passing textual and markdown converted tabular context of relevant pages.

Following the benchmark’s annotation schema, we aim to 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, not subcategories.

Step 4: Post-processing. We parse the LLM output, via regex for the question prompt and JSON parsing for the structured prompt, normalise units using a unit dictionary (standardising to tonnes of CO2 equivalents (CO2e)) and removing duplicates via custom prioritization rules (see Appendix A.3 for details).

Step 5: Evaluation. We compare extracted values to the ground truth values from the benchmark dataset and compute report-level precision, recall and F1-score. We rely on a publicly available benchmark, expert-annotated dataset of sustainability information from a sample of 139 sustainability 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). The sector- and region-diverse set of companies publishing the PDF reports are a random selection of around 4000 companies featured in two major indices (international "MSCI World Small Cap" and German "DAX").

4 Results

4.1 Model experiments

Among the tested models (Table 1), gpt-5.2 delivers the strongest overall performance. Progress across gpt generations is uneven, however: while gpt-4.1 improves on gpt-4o across all three metrics, gpt-5 slips behind gpt-4.1 on precision and F1 despite reaching the highest recall of the closed-source models. The open-weight models fall considerably short of the gpt family. Notably, Llama-4 despite having more than twice the number of parameters of gpt-oss-120b (400B vs. 117B) shows highly unstable recall and ends up performing worse than the smaller gpt-oss-120b overall. Formal significance tests (indicated by stars in Table 1) confirm that gpt-5.2 significantly outperforms gpt-5, Llama-4, and

Model	Created	Precision	Recall	F1 Score
gpt-4o	11/24	0.656 (0.007)	0.929 (0.005)	0.769 (0.006)
gpt-4.1	04/25	0.658 (0.017)	0.944 (0.017)	0.775 (0.009)
gpt-5	08/25	0.644 (0.009)	0.950 (0.008)	0.768** (0.007)
gpt-5.2	12/25	0.779 (0.015)	0.878 (0.084)	0.823 (0.039)
Llama-4	04/25	0.182 (0.034)	0.817 (0.159)	0.293*** (0.043)
gpt-oss	08/25	0.812 (0.040)	0.281 (0.042)	0.416*** (0.047)

Table 1: Averages and standard deviations of 5 runs of model experiments. 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. Significance markers on F1 scores indicate one-sided hypothesis tests against the reference model (gpt-5.2). Details in Appendix A.4. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

Prompt Type	Input Mode	Precision	Recall	F1 Score
question	text	0.779 (0.015)	0.878 (0.084)	0.823 (0.039)
question	text+table	0.777 (0.021)	0.903 (0.014)	0.835 (0.017)
structured	text	0.760 (0.008)	0.860 (0.011)	0.807 (0.006)
structured	text+table	0.755 (0.020)	0.830 (0.105)	0.787 (0.049)

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

gpt-oss, but the differences to gpt-4o and gpt-4.1 are not statistically supported (see Appendix A.4).

What the model extracts, mirrors what companies report: Scope 1 is most often extracted, Scope 2 location-based exceeds market-based since it is the default per our instruction if Scope 2 is not explicitly disclosed, and Scope 3 is extracted least often given its disclosure is not mandatory (cf. Figure A5).

A closer inspection reveals that gpt-oss-120b’s low score is substantially a parsing artifact rather than a capability gap: 85.8% of its errors on non-NA reports stem from model-specific output formatting (e.g., bare numbered lists, Unicode punctuation substitution, missing colon separators) that our regex parser cannot handle, rather than incorrect values. Pairing gpt-oss-120b with structured JSON output instead of the regex-parsed question prompt raises its F1 from 0.416 to 0.786 (5-run average), on par with gpt-4.1 (see Appendix A.5 for details).

4.2 Prompt experiments

Using gpt-5.2, our best-performing model, we next compare the two prompt designs (Table 2). Con-

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.

trary to best practice, the simple question prompt (row 1) achieves higher precision, higher recall, and consequently a higher F1 score than the structured prompt (row 3); the F1 difference is statistically significant (one-sided $p < 0.05$; see Appendix A.4).

To identify which parts of the structured prompt account for this outcome, we ablate its components individually (Table 3). Their contributions differ sharply. Dropping the role description nudges F1 slightly upward, from 0.807 to 0.821; dropping the scope definitions has essentially no effect while dropping the extraction rules causes F1 to plummet. The strong effect of the extraction rules is worth highlighting, given that a qualitative inspection of extracted values shows that the model occasionally ignores them anyway⁴. Even the strongest ablation variant, however, still falls short of the question prompt’s F1, indicating that the structured format as a whole, not any single omitted component, drives the underperformance.

The comparison between question and structured prompts is not fully controlled: the two designs differ not only in output format but also in preamble content and length, and prior work has shown that requiring structured output (Tam et al., 2024) and increasing distance between task definition and context (Liu et al., 2024) can affect performance. Our ablation of individual preamble components (Table 3) partially addresses this by showing that removing the role description or scope definitions does not close the gap; a further ablation pairing the question prompt with JSON output (see Appendix A.6) also reduces rather than improves its F1, though with overlapping confidence intervals. A fully controlled comparison isolating output format from preamble structure remains an open direction. We thus interpret the finding as evidence that the structured prompt as a whole underperforms

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

the question prompt in this setting, rather than as an isolated claim about JSON output format.

Categorising each report’s extraction result under both prompts (see Figure A4 in the appendix) shows that more than 85% remain in the same category (correct, partially correct, incorrect) while the prompt design shifts only a handful of borderline cases. Two systematic error patterns appear under both prompts: value–year/scope misalignment, and extraction of subcategory values when both subcategories and totals for the same scope-year appear on the same page. Overall, the model handles reports without ground-truth values better than reports with them (91% correctly returning nothing vs. 51% correctly extracting the disclosed values).

4.3 Tabular context experiments

Adding the Markdown-serialized table as context yields a modest F1 gain for the question prompt, primarily through additional extractions (Figure A5) leading to improved recall (rows 1 and 2 of Table 2); this improvement is statistically significant (one-sided $p < 0.01$; see Appendix A.4).

For the structured prompt, the same intervention has the opposite effect, reducing both precision and recall. To probe how robust this pattern is, we repeat the experiments with our second-best model, gpt-4.1 (Table A5); here, tabular context degrades performance for both prompt types.⁵

Manual inspection of the 28 erroneous text-only extractions with ground-truth values in tables clarifies the limited effect: only 6 exhibit resolvable misalignment errors, and just 2 are corrected by the added context. The remaining 22 reports exhibit errors, unaddressed by tabular structure, such as irrelevant extractions or footnote misinterpretations.

All results reported in this paper are based on raw values. Inspecting the unit normalization as part of the post-processing in more detail, we find that text+table context increases the rate at which extracted units fail to resolve to a known category and instead fall into *Other* (Table A6), a discrepancy not reflected in the raw-value metrics reported above. Examining a representative report (Paramount Group, 2020, page 47), the unit “metric tons CO2e” appears only once, in the table header, and not alongside the individual Scope 1 values themselves. Under text-only context, the model infers this header unit and appends it to each ex-

traction; under text+table context, it omits the “e” suffix and appends “metric tons CO2” which is absent from the unit mapping. Notably, the markdown table only contains values and no units.

The unresolved unit rate is also uneven across scopes, with Scope 1 and Scope 2lb tracking closely and Scope 3 remaining lowest throughout (see Appendix A.9 for details).

5 Conclusion

Our systematic evaluation for LLM-based GHG emissions extraction suggests that commonly recommended prompt and model choices do not always translate into higher extraction quality. Newer model generations do not consistently improve upon older ones, and the open-weight models we tested trail meaningfully behind the closed-source gpt family, pointing to the value of empirically comparing candidate models on a representative slice of one’s own data rather than defaulting to the latest release. A structured prompt following common best practices is outperformed by a simpler question-style prompt, and the effectiveness of tabular context depends on the model-prompt combination. Cutting across these observations, we see a recurring precision-recall trade-off: gains in one metric typically come at the expense of the other, both across model generations and across structured prompt components. gpt-5.2 is a partial exception, achieving higher precision without a proportional loss in recall, but even here the pattern reappears when comparing configurations against one another. Extraction rules are the notable exception on the prompt side, contributing to both precision and recall, whereas role and scope definitions trade one for the other.

From a deployment cost perspective (see Appendix A.10), the best-performing configurations are considerably more expensive than their close alternatives: switching from gpt-4.1 to gpt-5.2 raises cost by roughly 46% for only a small F1 improvement, and the question prompt is 3.8× more expensive than the structured prompt on gpt-5.2 for a similarly modest gain. The prompt-side gap in particular reflects the question prompt’s output structure, which restates all 44 scope-year queries and produces roughly 15× more output tokens than the structured prompt’s JSON format. Cost differences between design choices can thus dwarf the corresponding performance gains, making the cheaper option often the more rational one.

⁵Since the Markdown conversion is identical across prompt conditions, any potential conversion errors would affect them symmetrically and cannot account for the diverging results.

Limitations

Several caveats should be noted. Our evaluation relies on a small benchmark that is deliberately curated for annotation quality and consists solely of English-language reports, which may constrain how far the findings extend to larger or multilingual corpora. That said, the benchmark covers a broad diversity of document structures and report lengths, mirroring the variability observed in real-world corporate reporting. We believe that the resulting findings should therefore remain meaningful within this scope. Extending our evaluation methodology to additional sustainability corpora is a natural direction for future work, which the released `climatextract` package is designed to facilitate.

The original PDF reports cannot be redistributed through our repository. Readers wishing to reproduce our experiments must retrieve the source documents themselves through the public links accompanying the benchmark dataset (Beck et al., 2025).

More broadly, the design space for LLM-based information extraction, spanning prompt structure, contextual input, and model choice, is vast, and our experiments necessarily cover only a principled subset of it. Findings might therefore not transfer to configurations we did not test.

Our model selection is likewise deliberately narrow: we focus on the closed-source gpt family given its industry prevalence, complemented by gpt-oss and Llama-4 as established representatives of the open-weight landscape. Alternative model families may behave differently.

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 Sus-](#)

[tainability 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.

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.

Andreas Dimmelmeier, Hendrik Doll, Malte Schierholz, Emily Kormanyos, Maurice Fehr, Bolei Ma, Jacob Beck, Alexander Fraser, and Frauke Kreuter. 2024. [Informing climate risk analysis using textual information - a research agenda](#). In *Proceedings of the 1st Workshop on Natural Language Processing Meets Climate Change (ClimateNLP 2024)*, pages 12–26, Bangkok, Thailand. Association for Computational Linguistics.

Kerstin Forster, Lucas Keil, Victor Wagner, Maximilian A. Müller, Thorsten Sellhorn, and Stefan Feuerriegel. 2025. [Assessing corporate sustainability with large language models: Evidence from europe](#). Working Paper 202, TRR 266 Accounting for Transparency Working Paper Series.

R. Greene, T. Sanders, L. Weng, and A. Neelakantan. 2022. [New and improved embedding model](#). Accessed: 2026-01-22.

Biyang Guo, He Wang, Wenying 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.

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 Language Processing Workshop 2024*, pages 303–317, Miami, FL, USA. Association for Computational Linguistics.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. [Lost in the middle: How language models use long contexts](#). *Transactions of the Association for Computational Linguistics*, 12:157–173.

Rafik Mankour, Yassine Chafai, Hamada Saleh, Ghassem 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.
- Jingwei Ni, Julia Bingler, Chiara Colesanti-Senni, Mathias Kraus, Glen Gostlow, Tobias Schimanski, Dominik Stambach, Saeid Ashraf Vaghefi, Qian Wang, Nicolas Webersinke, Tobias Wekhof, Tingyu Yu, and Markus Leippold. 2023. [CHATREPORT: Democratizing sustainability disclosure analysis through LLM-based tools](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 21–51, Singapore. Association for Computational Linguistics.
- 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.
- 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.
- 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.
- Saeid Ashraf Vaghefi, Dominik Stambach, Veruska Muccione, Julia Bingler, Jingwei Ni, Mathias Kraus, Simon Allen, Chiara Colesanti-Senni, Tobias Wekhof, Tobias Schimanski, Glen Gostlow, Tingyu Yu, Qian Wang, Nicolas Webersinke, Christian Huggel, and Markus Leippold. 2023. [Chatcli-mate: Grounding conversational ai in climate science](#). *Communications Earth & Environment*, 4(1):480.
- 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.

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.

We evaluated 12+ retrieval configurations on all 139 reports, varying the number of retrieved pages k and the context window size in Table A1. Using gpt-4o-mini for consistent LLM evaluation, we compared configurations on precision, recall, F1, retrieval failures, and cost. Table A1 reports results for varying k with a context window of 0 adjacent pages, the best-performing window size. Setting $k = 7$ retrieved all ground-truth pages for 137 of 139 reports at competitive precision and F1, minimising retrieval failures. While $k = 6$ performed comparably in our dataset, we selected $k = 7$ as a more robust choice for future datasets.

A.2 Prompt designs

Figure A1 shows the question prompt template. The red dashed box was added when we adopted gpt-5.2, whose output includes Markdown symbols, not parsable by the regex schema. The instruction to reprint the first three questions in the response format was necessary to counteract the model’s observed tendency to “forget” early questions in the output.

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:

0. What is the title
1. What are the Scope 1 emissions in 2013
2. What are the Scope 1 emissions in 2014
3. What are the Scope 1 emissions in 2015
4. What are the Scope 1 emissions in 2016
5. What are the Scope 1 emissions in 2017
6. What are the Scope 1 emissions in 2018
7. What are the Scope 1 emissions in 2019
8. What are the Scope 1 emissions in 2020
9. What are the Scope 1 emissions in 2021
10. What are the Scope 1 emissions in 2022
11. What are the Scope 1 emissions in 2023
12. What are the Scope 2 (market-based) emissions in 2013
13. What are the Scope 2 (market-based) emissions in 2014
14. What are the Scope 2 (market-based) emissions in 2015
15. What are the Scope 2 (market-based) emissions in 2016
16. What are the Scope 2 (market-based) emissions in 2017
17. What are the Scope 2 (market-based) emissions in 2018
18. What are the Scope 2 (market-based) emissions in 2019
19. What are the Scope 2 (market-based) emissions in 2020
20. What are the Scope 2 (market-based) emissions in 2021
21. What are the Scope 2 (market-based) emissions in 2022
22. What are the Scope 2 (market-based) emissions in 2023
23. What are the Scope 2 (location-based) emissions in 2013
24. What are the Scope 2 (location-based) emissions in 2014
25. What are the Scope 2 (location-based) emissions in 2015
26. What are the Scope 2 (location-based) emissions in 2016
27. What are the Scope 2 (location-based) emissions in 2017
28. What are the Scope 2 (location-based) emissions in 2018
29. What are the Scope 2 (location-based) emissions in 2019
30. What are the Scope 2 (location-based) emissions in 2020
31. What are the Scope 2 (location-based) emissions in 2021
32. What are the Scope 2 (location-based) emissions in 2022
33. What are the Scope 2 (location-based) emissions in 2023
34. What are the Scope 3 emissions in 2013
35. What are the Scope 3 emissions in 2014
36. What are the Scope 3 emissions in 2015
37. What are the Scope 3 emissions in 2016
38. What are the Scope 3 emissions in 2017
39. What are the Scope 3 emissions in 2018
40. What are the Scope 3 emissions in 2019
41. What are the Scope 3 emissions in 2020
42. What are the Scope 3 emissions in 2021
43. What are the Scope 3 emissions in 2022
44. 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 gpt-5.2.

Figure A2 shows the structured prompt template. Beyond the relevance criteria listed in the main text, the prompt details further extraction rules addressing not-intended transformations (e.g. rejecting percentage increases or intensity ratios).

The output schema in Figure A3 is defined as a Pydantic model in Python and serialised to JSON, specifying a list of KPI entries, each representing one scope-year combination, with the four fields year, kpi_name (the scope), value, and unit. Each field carries a natural-language description and example value that are passed to the LLM as part of the structured prompt to guide extraction; these are omitted from Figure A3 for brevity but are visible in the released code.

A.3 Pipeline Step 4: Parsing

Unit normalization. Companies use different unit variants for the same quantity (e.g. "1,000 t CO₂" and "metric kilotons CO₂e"). We collect all distinct variants in the dataset and normalise them via a dictionary to one of "t CO₂e", "kt CO₂e", "Mt CO₂e", "MM Mt CO₂e", "kg CO₂e" or "Other" if unknown. A multiplier per known unit allows standardising all values to "t CO₂e" for downstream comparability.

Duplicate resolution. To retain only one entry per scope-year combination, we apply three prioritisation rules sequentially: (1) retain only one entry among duplicates sharing the same value and unit on the same page; (2) among entries with different units, retain the one in "t CO₂e"; (3) among entries with matching units, retain the one from the majority page, breaking ties by the earliest page.

A.4 Statistical Testing

To assess the significance of the differences reported in Section 4, we fit fixed-effects regressions on report-level F1 scores. Report-level averaging is used because extractions within the same report cannot be assumed to be independent. Each comparison of interest uses one experimental condition as the reference and enters the other as a dummy variable; the coefficient on the dummy is the estimated F1 difference. We cluster standard errors at both the report and run level: reports differ in complexity, and even at temperature = 0 LLM APIs can produce small run-to-run variation due to backend nondeterminism.

Each test corresponds to a directional hypothesis of the form $F1_A > F1_B$ derived from best-practice recommendations. We compute one-sided p-values against the null that the opposite direction holds. Table A2 reports the coefficient, clustered standard error, and outcome for each comparison.

Two observations are worth flagging. First, the prompt hypothesis $F1_{\text{structured}} > F1_{\text{question}}$ was significantly rejected in the tested direction, meaning the question prompt outperforms the structured prompt at $p < 0.05$. Second, comparisons of gpt-5.2 against gpt-4o and gpt-4.1 do not reach significance, so although gpt-5.2 achieves the highest F1 in absolute terms, the gap to the closest earlier generations is not statistically supported.

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).		Definitions
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 invent values. Ensure your extraction only includes absolute values for the defined KPIs, strictly following these guidelines. If the subtype "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 the scopes.		Specifications
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.

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

Table A2: Directional hypothesis tests 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 (report and run) in parentheses. Significance: *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

A.5 Output Format for gpt-oss-120b

Section 4.1 mentions that 85.8% of gpt-oss-120b's errors on non-NA reports stem from output-formatting issues our regex parser cannot handle, rather than incorrect values. To contextualize this, we compare against Llama-4 (alternative open-weight model) and gpt-4.1 (second-best model): only 13.1% and 3.8% of their errors are formatting-driven. This contrast underscores that gpt-oss-120b's apparent underperformance is largely an artifact of its output format. We therefore re-run gpt-oss-120b under the structured JSON prompt to assess its capability independent of output formatting; Table A3 reports the results, averaged over 5 runs.

```
{
  "$defs": {
    "KpiEntry": {
      "type": "object",
      "properties": {
        "year": {"type": ["integer", "null"]},
        "kpi_name": {"type": ["string", "null"],
          "enum": ["1", "2 (market-based)", "2 (location-based)", "3", null]},
        "value": {"type": ["number", "null"], "minimum": 0},
        "unit": {"type": ["string", "null"]}
      }
    }
  },
  "type": "object",
  "properties": {
    "KPI_Entries": {"type": "array", "items": {"$ref": "#/$defs/KpiEntry"}}
  },
  "required": ["KPI_Entries"]
}
```

Figure A3: Output schema serialised from Pydantic model

Prompt Type	Precision	Recall	F1
question	0.812 (0.040)	0.281 (0.042)	0.416 (0.047)
structured	0.760 (0.013)	0.814 (0.017)	0.786 (0.012)

Table A3: gpt-oss-120b performance by output format, averaged over 5 runs. Standard deviations in parentheses.

A.6 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 A4. 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 A4: 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.

A.7 Report-level analysis

We categorise each report’s extractions under both prompts and track how individual reports shift between categories. Reports are first split by whether the ground-truth benchmark contains disclosed emissions (non-NA reports) or not (NA reports). NA reports are classified as either correct (no values returned) or incorrect (at least one irrelevant value extracted). Non-NA reports are classified as completely correct, partially correct, or completely incorrect, with a separate retrieval-failure category for reports where the retrieved pages did not contain the relevant ground-truth information. Figure A4 shows the resulting transitions. Zooming in on the shifts: 4 NA reports switch from incorrect under the question prompt to correct under the structured prompt, but 2 correctly-handled NA reports shift the other way, extracting irrelevant values such as Scope 3 subcategory totals. Among non-NA reports, 2 of 139 are classified as retrieval failures where ground-truth values were spread across pages and at least one relevant page was not retrieved. The persistence of subcategory-

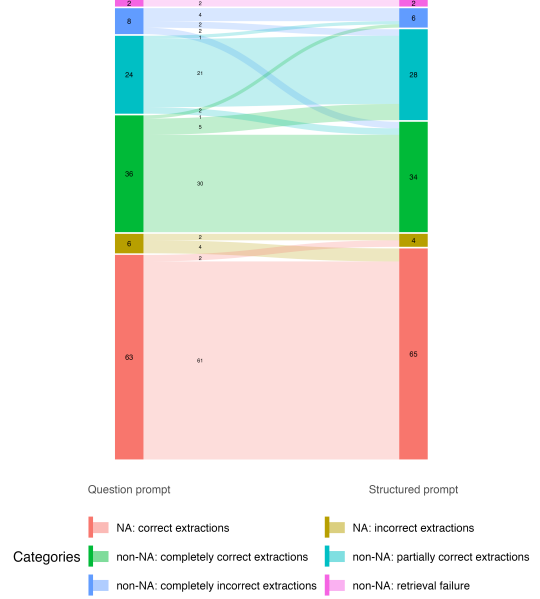


Figure A4: Diagram comparing report-level extraction results between the question prompt and the structured prompt when using gpt-5.2 model. Most reports remain in the same category. Improvements for some are offset by regression for other reports.

extraction errors under the structured prompt indicates that the LLM does not consistently follow the extraction rules included in the prompt, unlike the more successful adherence reported by Mankour et al. (2025); Neuberger et al. (2025). Another recurring error is the model’s tendency to interpret footnote numerals as part of a value.

A.8 Second-best model

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

Table A5: 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.

Table A5 reports the prompt-context experiments repeated with the second-best model, gpt-4.1. In contrast to gpt-5.2, adding tabular context weakens performance for both prompt types, indicating that the benefit of tabular context is not robust across models.

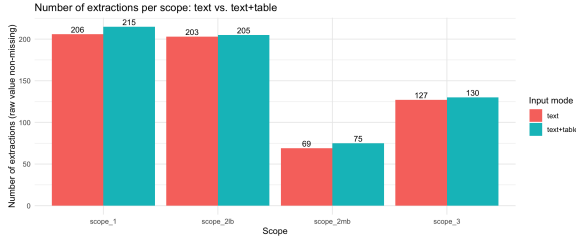


Figure A5: Number of non-missing extractions per emissions scope, comparing text-only and text+table context for gpt-5.2 model and question prompt type. Text+table yields equal or higher counts across all scopes, with the largest relative gains for Scope 2 market-based and Scope 3.

unit_cat	text (%)	text+table (%)
t CO2e	68.5	59.2
kt CO2e	13.2	12.3
Other	9.8	19.7
Mt CO2e	7.5	8.0
MM Mt CO2e	1.0	0.0
kg CO2e	0.0	0.8

Table A6: Distribution of resolved unit categories under text and text+table context for gpt-5.2 model and question prompt type. Text+table shifts mass from *t CO2e* toward *Other*, consistent with the model dropping the “e” suffix when the unit-less structured table is included.

A.9 Tabular context: recall gains and unit-resolution

This appendix provides scope-level detail supporting the effects of tabular context (text+table) on extraction recall and unit resolution discussed in Section 4.3.

Table A6 reports the full distribution of resolved unit categories under text-only and text+table context, including the share falling into *Other* when no key in the unit mapping matches. Figure A6 isolates this unresolved share and breaks it down by scope. Scope 1 and Scope 2lb track each other closely in both settings because they are conventionally reported as adjacent line items sharing a single unit descriptor (e.g., one column header spanning both rows in a summary table); a mapping miss for that shared string therefore fails both scopes simultaneously. Case numbers are too small for a detailed analysis of Scope 2mb: Almost all unresolved units from Scope 2mb under text+table originate from a single PDF file. Scope 3 remains lowest throughout, consistent with its typically being reported in a separate category-breakdown table rather than alongside Scope 1/2, and so not exposed to the same shared-unit failure mode.

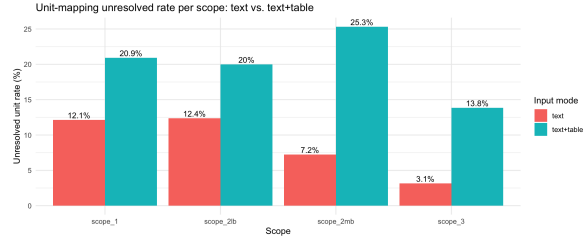


Figure A6: Rate of extracted units unresolved to a known category (mapped to *Other*) per scope, text vs. text+table context for gpt-5.2 model and question prompt type. Text+table increases the unresolved rate across every scope.

A.10 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 A7: 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 A8: 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.

Tables A7 and A8 report the cost figures underlying the discussion in the conclusion. Figures represent averages of 5 runs over 139 reports; embedding costs are excluded, since embedding is a one-time cost per report.

Table A7 compares the two best models under the question prompt and text-only context: switching from gpt-4.1 to gpt-5.2 increases cost by 46% on average for a 6% F1 improvement (0.775 → 0.823).

Table A8 compares prompt-context combinations for gpt-5.2. Adding tabular context adds only 0.78 EUR to the question prompt and 0.12 EUR to the structured prompt, but its effect on F1 is direc-

tional: $+1.4\%$ for the question prompt and -2.4% for the structured prompt.