Extracting Greenhouse Gas Emission Values from Corporate Sustainability Reports

A Case Study on Evaluating and Finetuning Language Models on Long-Context Structured Information Extraction

March 11, 2024

https://github.com/nopperl/corporate_emission_reports

Motivation

Current State

- Data about corporate greenhouse gas (GHG) emissions is usually published only as part of sustainability report PDF files.
- This format is not machine-readable.
- Interested actors have to manually extract emission data from these reports.
- This process is tedious and time-consuming.

Potential Solution

An automatic information-extraction system could:

- Extract GHG emissions data from sustainability report PDF files.
- Provide the data in a machine-readable format.
- Save time and effort for interested actors.

Objective

Example

Example input (excerpt)

PERFORMANCE DATA TABLE

Key Performance Indicator	KPI Sub-Metric	Unit	Status for FY22	Footnote(s)
Total Energy Consumed	Total amount of energy consumed	GigaJoules	26,576	California only. Excludes self-generated solar energy.
Percentage of Energy Consumed Supplied from Grid Electricity	Percentage of Energy Consumed Supplied from Grid Electricity	Percentage	71%	
Scope 1 Emissions	Gross Scope 1 Emissions	Metric tons CO2e	581	California operations only. Excludes fugitive emissions.
	Emissions from Fertilizer	Metric tons CO2e	118	California operations only.
Scope 2 Emissions	Emissions from Electricity and Energy Purchases	Metric tons CO2e	1,220	California operations only.
Percentage of Estate Vineyards in Regions with High or Extremely High Baseline Water Stress	Percentage of Estate Vineyards in Regions with High or Extremely High Baseline Water Stress	Percentage	6%	
Number of Incidents of Non-Compliance Associated with Quantity / Quality of Water Permits, Standards, and Regulations	Number of Incidents of Non-Compliance Associated with Quantity / Quality of Water Permits, Standards, and Regulations	Number	0	
Training Completion on Responsible Marketing and Advertising Practices	Number of Training Hours Completed on Responsible Marketing and Advertising Practices	Hours	30	
Number of Incidents of Non-Compliance with Industry or Regulatory	Number of Incidents of Non-Compliance with Industry or	Number	0	

Example output

{"scope_1":581,"scope_2":1220,"scope_3":null,"sources":[16,17,56,7]}

Challenges

- Reports are very long $\rightarrow$ long context task.
- No available ground truth dataset of reports and extracted GHG emission values.

Approach

1. Assemble evaluation dataset.
2. Develop inference system to use language models for emission extraction.
3. Benchmark the performance of selected language models.
4. Generate finetuning dataset using best-performing large model.
5. Finetune the best-performing small model on the generated dataset.
6. Deploy the finetuned model.

Evaluation dataset

- 100 sustainability reports from geographically-diverse corporations and manually-extracted emission values.

System architecture

- Developed in Python.
- Inference using llama.cpp.
- Consists of four parts:
  - Input data is extracted from a sustainability report,
  - inserted into a prompt,
  - whis is used by the language model
  - to produce a structured output.

Input data and prompt

Simple RAG setup:

1. Plain-text semi-structured XHTML representation of report PDF is extracted using PyMuPDF.
2. Extracted text is split into chunks by page.
3. All chunks containing relevant GHG emission terms (such as `Scope 1`) are retrieved.
4. Retrieved plain-text chunks are inserted into a predefined prompt.

Resulting prompt length in tokens:

	max	mean	median	min
token_length	60063.00	14544.31	12184.50	1004.00

Output

- Model output is constrained using BNF grammar to produce a JSON according to a predefined schema.

Example output

```
{"scope_1":581,"scope_2":1220,"scope_3":null,"sources":[16,17,56,7]}
```

Evaluation

Models

Model	**Param Size**	**Context Length**
Mistral-7B-Instruct-v0.2	7B	32768
openchat-3.5-0106	7B	8192
Qwen-1.8B-Chat	1.8B	8192
Mixtral-8x7B-Instruct-v0.1	45B	32768
miqu-1-70b	70B	32764

Note: `Q5_K_M` quantization format of Mixtral-8x7B-Instruct-v0.1 and miqu-1-70b are used due to constrained resources.

Evaluation

Metrics

1. accuracy for every extracted emission value
2. source page retrieval accuracy

Evaluation

Result

scope 1	scope 2	scope 3	avg of scopes	sources	model
49	34	54	46	53	mistral
33	31	56	40	48	openchat
12	8	5	8	3	qwen-1.8B
69	72	57	66	74	miqu
70	71	69	69	64	mixtral

Finetuning

Dataset

- Collect 3233 sustainability reports different from the evaluation dataset.
- Generate outputs using Mixtral-8x7B-Instruct-v0.1 (the best performing model).
- Convert into instruction format.

Finetuning

Setup

- Finetune Mistral-7B-Instruct-v0.2 (the best performing $\leq$7B model).
- axolotl is used as finetuning framework.
- Finetuning a 7B model is resource intensive, especially for long sequences.
- Using standard configuration, only sequences up to a length of 6144 tokens can be trained.
- To enable training on sequences of up to 32768 tokens, following techniques are used:
  - LoRA
  - Flash Attention 2
  - ZeRO-3
  - bfloat16
  - bitsandbytes 8-bit AdamW optimizer.

Finetuning

Evaluation

scope 1	scope 2	scope 3	avg of scopes	sources	model
49	34	54	46	53	mistral
69	72	57	66	74	miqu
70	71	69	69	64	mixtral
65	62	69	65	77	**lora**

Contributions

- A manually-created evaluation dataset ($N = 100$),
- a synthetic finetuning dataset ($N = 3233$),
- an evaluation of multiple models (1.8B-45B) covering prompt strategies, numerical issues and self extend,
- an evaluation of different finetuning configurations,
- a finetuned Mistral-7B model which nearly matches and partially exceeds Mixtral (45B) on this task, and
- a web demo (`https://huggingface.co/spaces/nopperl/emission-extractor`).

Source code:
`https://github.com/nopperl/corporate_emission_reports`

Future work

- Evaluate multimodal vision-language models such as CogVLM for understanding visual information in PDF documents.
- Improve prompts using few-shot or chain-of-thought prompting (though token-expensive).
- Consider testing smaller encoder(-decoder) models such as XLM-RoBERTa or DeBERTa.