Can Long-Context Language Models Subsume Retrieval, SQL, and More?

Anonymous Author(s)
Affiliation
Address
email

Abstract

Long-context language models (LCLMs) have the potential to revolutionize our approach to tasks traditionally reliant on external tools like retrieval systems or databases. Leveraging LCLMs' ability to natively ingest and process entire corpora of information offers numerous advantages. It enhances user-friendliness by eliminating the need for specialized knowledge of tools, provides robust end-to-end modeling that minimizes cascading errors in complex pipelines, and allows for the application of sophisticated prompting techniques across the entire system. To assess this paradigm shift, we introduce LOFT, a benchmark comprising of real-world tasks requiring context up to millions of tokens designed to evaluate LCLMs' performance on in-context retrieval and reasoning. Our findings reveal that LCLMs can already achieve textual, visual, and audio retrieval performance comparable to specialized systems such as Gecko and CLIP, while still facing challenges in areas like multi-hop compositional reasoning required in SQL-like tasks. Notably, prompting strategies significantly influence performance, emphasizing the need for continued research as context lengths grow. Overall, LOFT provides a rigorous testing ground for LCLMs, showcasing their potential to supplant existing paradigms and tackle novel tasks as model capabilities scale.[1]

1 Introduction

Long-context language models (LCLMs) [42, 35, 4, 8] hold the promise of reshaping artificial intelligence by enabling entirely new tasks and applications while eliminating the reliance on tools and complex pipelines previously necessary due to context length limitations [17, 26]. By consolidating complex pipelines into a unified model, LCLMs ameliorate issues like cascading errors [7] and cumbersome optimization [23, 48], offering a streamlined end-to-end approach to model development. Moreover, techniques such as adding instructions [21, 46, 11], incorporating few-shot examples [9], and leveraging demonstrations via chain-of-thought prompting [34, 47] can be seamlessly integrated to optimize LCLMs for the task at hand.

However, realizing the full potential of LCLMs necessitates rigorous evaluation on truly long-context tasks useful in real-world applications. Existing benchmarks fall short in this regard, relying on synthetic tasks like the popular "needle-in-haystack" [19, 25] or fixed-length datasets that fail to keep pace with the evolving definition of "long-context" [6]. Critically, existing evaluations do not adequately stress-test LCLMs on these paradigm-shifting tasks.

To address this, we introduce the **Lo**ng-Context **F**ron**t**iers (LOFT) benchmark, a suite of six tasks comprising over 35 datasets spanning text, visual, and audio modalities designed to push LCLMs to their limits and gauge their real-world impact. Unlike previous benchmarks, LOFT allows for

[1] We will publicly release our dataset and evaluation code upon acceptance.

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

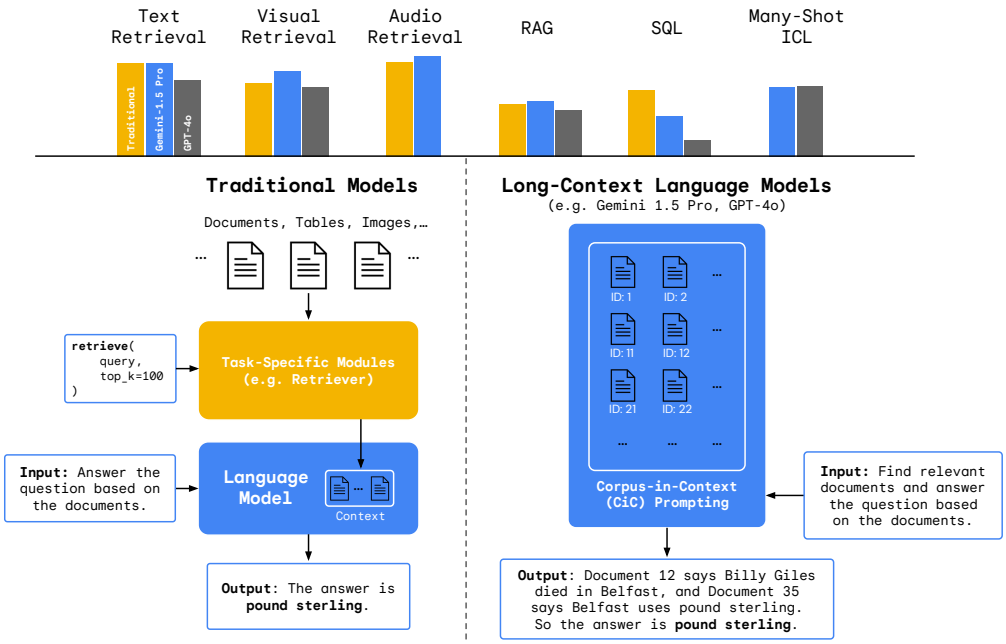

Figure 1: An overview of the LOFT benchmark, made of six tasks which measure LCLMs' ability to do in-context retrieval, reasoning, and many-shot learning on corpora up to millions of tokens. We compare the performance of LCLMs against traditional task-specific models (*e.g.,* CLIP for visual retrieval), which often rely on complex task-specific pipelines. Unlike traditional models, we show how LCLMs can simplify various tasks through Corpus-in-Context Prompting (Section 3).

automatic creation of varied context lengths, up to and exceeding 1 million tokens, ensuring rigorous evaluation as LCLMs continue to scale. Our benchmark focuses on the following areas where LCLMs have the potential for disruption:

- **Retrieval**: LCLMs can directly ingest and retrieve information from a corpus, eliminating the need for separate dual-encoder models [20, 33, 24, 37]. This addresses the information bottleneck found in retrievers [38] by enabling fine-grained interactions between query and corpus. We assess retrieval performance across text, visual, and audio modalities.
- **Retrieval-Augmented Generation (RAG)**: LCLMs simplify RAG pipelines by directly reasoning over a corpus, overcoming challenges like query decomposition [36] and mitigating cascading errors due to retrieval misses [7, 30].
- **SQL**: We explore LCLMs' capacity to process entire databases as text, enabling natural language database querying and bypassing conversion to a formal query language like SQL [53]. This potentially enables more expressive querying and handling of noisy or mixed-structured data.
- **Many-Shot ICL**: LCLMs can scale the number of examples from the tens in the traditional in-context learning setup to hundreds or thousands, removing the need to find the optimal set of few-shot examples to use [31].

The LOFT benchmark opens up a novel line of research on long-context prompting, which we introduce as Corpus-in-Context (CiC) Prompting (Section 3). Using this approach, we evaluate Gemini 1.5 Pro [Reid et al., 2024] and GPT-4o [Achiam et al., 2023] on LOFT. Figure 1 summarizes the performance of these LCLMs and traditional models on each task, showcasing how LCLMs can tackle LOFT tasks without specialized pipelines.

Our evaluation of state-of-the-art LCLMs on LOFT reveals several notable findings. At the 128k token level, the largest size comparable across all models, all closely match the performance of specialized systems in textual retrieval, with Gemini also performing significantly better than specialized systems in visual and audio retrieval. On complex multi-hop compositional reasoning tasks, however, all LCLMs lag considerably, highlighting significant room for improvement. Furthermore, rigorous ablations on prompting strategies such as the format of the corpora, the incorporation of

	Dataset	Description	Avg. Cand. Length	# Cand. (128k)	Candidates	Input	Target
Text Retrieval	ArguAna	Argument Retrieval	196	531	Passages	Query	Passage ID(s)
	FEVER	Fact Checking	176	588			
	FIQA	Question Answering	196	531			
	MSMarco	Web Search	77	1,174			
	NQ	Question Answering	110	883			
	Quora	Duplication Detection	14	3,306			
	SciFact	Citation Prediction	301	357			
	Touché-2020	Argument Retrieval	330	329			
	TopiOCQA	Multi-turn QA	149	680			
	HotPotQA	Multi-hop QA	74	1,222			
	MuSiQue	Multi-hop QA	120	824			
	QAMPARI	Multi-target QA	132	755			
	QUEST	Multi-target QA	328	328			
Visual Retrieval	Flickr30k	Image Retrieval	258	440	Images	Text Query	Image ID
	MS COCO	Image Retrieval	258	440	Images	Text Query	Image ID
	OVEN	Image-text Retrieval	278	448	Images+Texts	Image+Text Query	Wikipedia ID
	MSR-VTT	Video Retrieval	774	140	Videos	Text Query	Video ID
Audio Retrieval	FLEURS-en	Audio Retrieval	249	428	Speech	Text Query	Speech ID
	FLEURS-es		315	343			
	FLEURS-fr		259	412			
	FLEURS-hi		292	369			
	FLEURS-zh		291	370			
RAG	NQ	Question Answering	110	883	Passages	Question	Answer(s)
	TopiOCQA	Multi-turn QA	149	680			
	HotPotQA	Multi-hop QA	74	1,222			
	MuSiQue	Multi-hop QA	120	824			
	QAMPARI	Multi-target QA	132	755			
	QUEST	Multi-target QA	328	328			
SQL	Spider	Single-turn SQL	111k	1	SQL Database	Question	Answer
	SParC	Multi-turn SQL	111k	1			
Many-Shot ICL	BBH-date	Multiple-choice QA	131	150	Training Examples	Question	Answer
	BBH-salient	Multiple-choice QA	246	104			
	BBH-tracking7	Multiple-choice QA	205	123			
	BBH-web	Multiple-choice QA	43	150			
	LIB-dialogue	Classification	266	284			

Table 1: Tasks and datasets in the LOFT benchmark. LOFT has 6 types of tasks, 4 modalities, and 35 datasets in total. For each dataset, we show the average length of the candidates (Avg. Cand. Length) as well as the number of candidates (# Cand) in the 128k version of LOFT.

chain-of-thought reasoning, and the location of the target information within the context, reveal large variance in performance, underscoring the need for further research to make LCLMs robust and instructable. Taken together, our results on LOFT demonstrate that LCLMs can match the performance of many specialized systems, while also revealing ample headroom for improvement in robust long-context reasoning as context windows continue to scale.

2 LOFT: A 1 Million+ Token Long-Context Benchmark

The LOFT benchmark aims to cover a wide range of real-world applications where LCLMs can be employed. These tasks span from retrieving relevant documents for a query to extracting compositional information from databases. Table 1 lists all tasks and their corresponding datasets.

For each dataset in all tasks, we sample up to 100 test queries, 5 few-shot queries, and 10 development queries. To test how LCLMs perform with a larger number of tokens in their context, we create LOFT with four different length limits, namely 32k[2], 128k, 200k, and 1M. To allow testing the same set of queries over different context lengths, we process each dataset to have the same evaluation queries across different context lengths (except for SQL, where we split queries by database size).

Retrieval & RAG We include diverse text retrieval and RAG datasets, covering heterougenous retrieval tasks from BEIR [43], multi-turn conversational QA, multi-hop QA [49, 44], as well as multi-target QA that require set-operations [3, 32]. For retrieval, we also include multimodal datasets, covering image, video, and audio.

[2]Since the gold documents of 100 test queries alone often exceed 32k tokens, we do not include test queries for the 32k version. We report the development set performance for 32k instead.

All queries in each retrieval and RAG dataset shares a single corpus, mimicking real retrieval applications. To create this shared corpus, we first include all gold passages from few-shot, development, and test queries, and then randomly add random passages until reaching the desired context size (Figure 2). This construction ensures smaller corpora (e.g., 128k) are subsets of larger ones (e.g., 200k). Gold and random passages are shuffled to avoid positional biases. For fair comparison, our results comparing traditional baselines to LCLMs are also done on this same corpora of data.

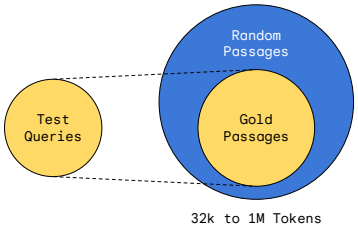

Figure 2: **Corpus creation** for retrieval and RAG. Given a set of test queries, we use their associated gold passages and other random passages to form the corpus.

Many-shot ICL We adapt datasets from Big Bench Hard (BBH) [40] and LongICLBench (LIB) [28] to evaluate LCLMs' many-shot in-context learning (ICL) capabilities. Similar to retrieval and RAG, we construct shared many-shot ICL contexts, ensuring training examples in smaller contexts are included in larger ones. Since all datasets are classification tasks, we guarantee that each classe is represented at least once.

SQL We evaluate SQL-like reasoning on Spider, a single-turn text-to-SQL dataset [51], and SparC, its multi-turn variant [52]. The corpus for each query is its associated database of one or more tables. For a maximum corpus size of N, we select queries with the largest databases still under N. Therefore, unlike shared corpus tasks, the query sets differ across LOFT sizes.

Given a maximum context length of $N \in \{32\text{k}, 128\text{k}, 200\text{k}, 1\text{M}\}$, we create a corpus up to a size of $0.9N$, to account for differences in tokenizers and reserving room for for instructions and formatting as we will see in Figure 3. Please refer to Appendix A for more details about dataset selection.

3 Corpus-in-Context Prompting

Traditionally, utilizing large corpora of passages, data tables, or training examples required specialized recipes or systems. Long-context language models (LCLMs) now enable direct ingestion and processing of entire corpora within their context window. This unlocks a novel prompting-based approach for solving , which we call **Corpus-in-Context** prompting (**CiC**, pronounced "sick").

3.1 Prompt Design

CiC prompting effectively combines established prompting strategies, tailoring them to leverage the unique capabilities of LCLMs for learning, retrieving and reasoning over in-context corpora. Figure 3 illustrates our key design choices, whose effectiveness is rigorously evaluated through extensive ablation studies in Section 5.

Instructions We first provide task-specific instructions to guide the LCLM's behaviors [21, 46, 11]. As an example for the retrieval task in Figure 3, we ask the model to read the corpus carefully and find relevant documents to answer the question.

Corpus Formatting We then insert the entire corpus into the prompt. The structure of the corpus significantly impacts retrieval performance. We find that careful formatting, such as repeating document IDs after passage text in retrieval, mitigates the effects of causal attention in decoder-only LCLMs, enhancing retrieval accuracy.

Few-Shot Examples Providing a limited number of demonstrations helps the LCLM grasp the desired response format and improves task accuracy [9]. Unlike common approaches where few-shot examples are independent, we ground all examples to the same corpus, aiming to teach the model understand the specific corpus. As we will see, positioning these examples can guide the model's attention to areas where it is typically weaker, mitigating "dead zones" in attention distribution.

Each few-shot example is accompanied by a Chain-of-Thought reasoning [34, 47]. We find adding Chain-of-Thought reasoning chains leads to the greatest benefits on tasks requiring complex multi-hop compositional reasoning.

Query Formatting The final evaluation query is formatted similar to each few-shot example (if any). Based on our query formatting, LCLMs complete the generation and provide answers.

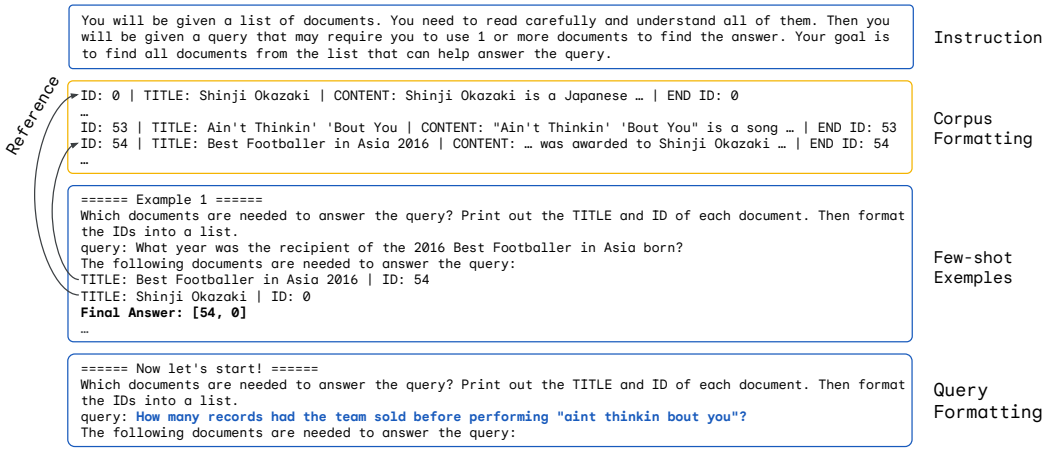

Figure 3: **Example of Corpus-in-Context Prompting** for retrieval. CiC prompting leverages large language models' capacity to follow instructions, leverage few-shot exemplars, and benefit from reasoning demonstrations to retrieve and reason over large corpora provided in context.

3.2 Discussion on Efficiency

Encoding a one million token context can be slow and computationally expensive. One key advantage of CiC prompting is its compatibility with prefix-caching in autoregressive language models as the query appears at the end of the prompt. This means *the corpus only needs to be encoded once*, similar to the indexing process in traditional information retrieval or database systems.

4 LOFT Tasks and Primary Results

Our evaluation on LOFT employs two state-of-the-art LCLMs: Google's **Gemini-1.5-Pro** [42] and OpenAI's **GPT-4o** [35]. These models were selected because their APIs support the most modalities in the benchmark. Their maximum context lengths are 2 million and 128k tokens, respectively. We use their official APIs[3,4] for the evaluation. A small number of API calls were blocked due to various reasons, which were treated as incorrect.

4.1 Text Retrieval

We adopt Gecko [24], a state-of-the-art retriever as the traditional task-specific baseline. Gecko is a dual-encoder model fine-tuned on extensive text retrieval and similarity tasks. To ensure fair comparison, we use the same corpus used to test the LCLMsto evaluate Gecko.

Results Results in Table 2 demonstrate that at 128k context, Gemini-1.5-Pro perform comparably to Gecko. This is notable, as LCLMs have not undergone specialized contrastive learning for retrieval. While LCLMs's performance does degrade when scaling the corpus to millions of tokens (Figure 5), this initial parity suggests the potential of LCLMs for retrieval tasks.

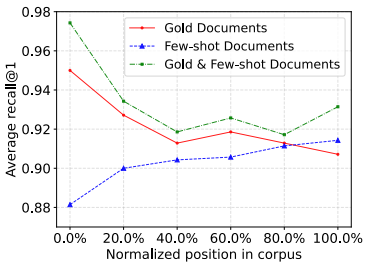

Figure 4: **Positional Analysis**. We vary the position of gold and few-shot documents within the corpus (0% = beginning, 100% = end).

Positional Analysis To better understand the cause of performance degradation of LCLMs on larger context length datasets, we investigate how the position of gold and few-shot documents in the corpus influences retrieval [29].

Figure 4 reveals that performance drops as gold documents move towards the end of the corpus, suggesting reduced attention in later sections. Conversely, placing few-shot examples at the end improves recall, indicating their ability to mitigate attention weaknesses in this region. Co-locating gold and few-shot documents consistently boosts performance. This demonstrates how few-shot

[3] https://ai.google.dev/gemini-api
[4] https://platform.openai.com/docs/models/gpt-4o

	Dataset	Gemini-1.5$_{\text{Pro}}$	GPT-4o	Traditional
Text Retrieval	ArguAna	0.84	0.54	0.75
	FEVER	0.98	0.92	0.97
	FIQA	0.79	0.20	0.83
	MSMarco	0.95	0.89	0.97
	NQ	1.00	0.92	0.99
	Quora	0.93	0.95	1.00
	SciFact	0.88	0.80	0.85
	TopiOCQA	0.31	0.29	-
	Webis-2020	0.91	0.71	0.88
	HotPotQA†	0.90	0.70	0.92
	MuSiQue†	0.42	0.18	0.29
	QAMPARI†	0.61	0.21	0.57
	QUEST†	0.30	0.19	0.54
	Average‡	**0.91**	0.74	**0.91**
Visual Retrieval	Flickr30k	0.84	0.65	0.75
	MS COCO	0.77	0.44	0.66
	MSR-VTT	0.76	0.72	0.64
	OVEN	0.93	0.89	0.79
	Average	**0.83**	0.68	0.71
Audio Retrieval	FLEURS-en	1.00	-	0.98
	FLEURS-es	0.99	-	0.99
	FLEURS-fr	1.00	-	1.00
	FLEURS-hi	1.00	-	0.74
	FLEURS-zh	1.00	-	1.00
	Average	**1.00**	-	0.94
RAG	HotPotQA	0.72	0.76	0.61
	MuSiQue	0.53	0.48	0.45
	NQ	0.81	0.76	0.70
	QAMPARI	0.39	0.20	0.51
	QUEST	0.28	0.12	0.31
	TopiOCQA	0.34	0.28	-
	Average‡	**0.55**	0.46	0.51
SQL	Spider	0.40	0.14	0.74
	SParC	0.36	0.13	0.55
	Average	0.38	0.14	**0.65**
Many-Shot ICL	BBH-date	0.88	0.81	-
	BBH-salient	0.78	0.64	-
	BBH-tracking7	0.33	0.81	-
	BBH-web	0.67	0.57	-
	LIB-dialogue	0.76	0.67	-
	Average	0.68	**0.70**	-

Table 2: **Main Results on LOFT 128k context test set**. We show performances of two LCLMs (Gemini-1.5$_{\text{Pro}}$ and GPT-4o) as well as baselines that are traditionally used to solve these tasks. For the evaluation metrics: text, visual, and audio retrieval use Recall@1; RAG uses span-level exact match; SQL uses execution accuracy; and many-shot prompting uses accuracy. †: retrieval datasets with multiple gold targets use mRecall@k (Appendix A). ‡: The average text retrieval and RAG performance excludes TopiOCQA as the traditional baseline does not support multi-turn queries.

examples can strategically counterbalance areas of weak attention, offering a promising approach to overcome performance degradation in large corpora. Per-dataset analysis is provided in Appendix C.

4.2 Visual Retrieval

We employ CLIP-L/14, a widely used text-to-image retrieval model, as our traditional task-specific baseline [37]. For Flickr30k and MSCOCO, CLIP performs text-to-image retrieval. For MSR-VTT, it performs text-to-video retrieval by averaging scores across frames. For OVEN, due to the lack of suitable open-source image-to-text models, we approximate image-to-text retrieval also using CLIP's text-to-image retrieval.

Results Gemini 1.5 Pro outperforms GPT-4o across all four visual benchmarks (Table 2). Notably, as shown in Figure 5, Gemini 1.5 Pro maintains a performance advantage over the CLIP across all visual benchmarks and context lengths.

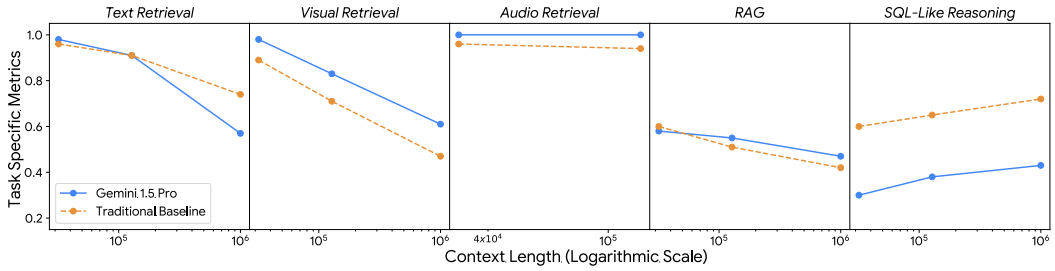

Figure 5: **Scaling results** of LCLMs compared to a traditional baseline by scaling the corpus size from 32k to 1 million tokens. Results are averaged over all constituent datasets in each task.

4.3 Audio Retrieval

Audio retrieval baseline used PALM 2 DE from [15], a dual-encoder model trained to maximize the similarity between audio and their transcription, and has achieved previous state-of-the-art on the FLEURS datasets. At present, GPT-4o does not support audio input.

Results Gemini-1.5-Pro demonstrates comparable performance to PALM 2 DE across all 5 languages (Table 2). We notice that Gemini-1.5 Pro notably surpasses PALM 2 DE in Hindi; this advantage likely stems from variations in pre-training data. Figure 5 further confirms Gemini-1.5-Pro's robust performance across various context length, highlighting the current capabilities of LCLMs while also indicating the need for more challenging audio datasets.

4.4 RAG

We set up a retrieve-and-read RAG pipeline as the baseline, using Gecko [24] for top-40 document retrieval, followed by Gemini-1.5-Pro for generating the answering conditioned on the question and the top documents.

Results Table 2 demonstrates that Gemini-1.5-Pro, with the entire corpus in context, outperforms the RAG pipeline on multi-hop datasets (HotpotQA and MusiQue). This is because long-context model can reason over multiple passages in the context window using Chain-of-Thoughts [47], a capability that RAG pipelines typically lack without sophisticated planning and iterative retrieval mechanisms.

However, a specialized retriever like Gecko excels at ranking all topically relevant passages from a corpus, enabling it to identify a comprehensive set of passages covering all answers. This proves particularly beneficial for multi-target datasets, such as QUEST and QAMPARI.

Interestingly, Figure 5 reveals that LCLMs also demonstrate superior RAG performance at 200k and 1M context lengths compared to the RAG pipeline, even though their retrieval performance on the corresponding retrieval datasets is inferior to Gecko.

Dataset	32k	128k/200k/1M
HotPotQA	0.60 (-0.30)	0.31 (-0.41)
MuSiQue	0.20 (-0.60)	0.10 (-0.43)
NQ	0.60 (-0.10)	0.37 (-0.44)

Table 3: **Gemini's closed-book performance on RAG** (32k = development, rest = test queries). Red indicates the performance difference compared to the CiC prompting.

Closed-Book Ablations To further probe capabilities, we conduct closed-book ablations on Gemini 1.5 Pro, removing the corpus to assess LCLM performance based solely on parametric knowledge [27, 30]. Table 3 presents the results, revealing that the closed-book performance significantly lags behind our long-context and traditional model. This underscores the tested models' effectiveness in leveraging external information from the corpus to enhance its reasoning capabilities.

4.5 SQL-Like Compositional Reasoning

SQL baseline uses a semantic parser to tranlsate the natural langauge input into SQL query, then excute the SQL query over the database. Specifically, we use DAIL-SQL [14], a state-of-the-art semantic parser that prompts an LLM. We adapt DAIL-SQL by replacing its LLM with Gemini 1.5 Pro and using the fixed set of few-shot examples.

Task (Metric)	Dataset	Best Prompt	Generic Instruction	Query at Beginning	Alphanumeric IDs	Titles Only	Without ID Echo	Corpus in Each Few-shot	Without CoT
Text Retrieval (Recall@1)	ArguAna	0.84	0.76	0.72	0.81	-	0.78	0.62	0.79
	FIQA	0.79	0.77	0.58	0.75	-	0.76	0.78	0.85
	NQ	1.00	0.98	0.98	0.99	0.91	1.00	1.00	1.00
	SciFact	0.88	0.88	0.81	0.90	0.84	0.87	0.78	0.90
Text Set (mRecall@k)	MuSiQue	0.49	0.44	0.19	0.44	0.10	0.36	0.35	0.43
	QAMPARI	0.61	0.61	0.49	0.54	0.09	0.49	0.35	0.43
	QUEST	0.28	0.28	0.22	0.30	0.05	0.27	0.22	0.30
RAG (Span EM)	MuSiQue	0.53	0.55	0.39	0.50	0.23	0.54	0.48	0.50
	NQ	0.81	0.78	0.73	0.80	0.40	0.80	0.81	0.81
	QAMPARI	0.39	0.30	0.30	0.33	0.08	0.26	0.30	0.25
	QUEST	0.28	0.31	0.16	0.25	0.02	0.24	0.26	0.29
	Average†	0.59	0.57	0.47	0.56	0.30	0.54	0.54	0.55
	(Δ)	-	(-0.02)	(-0.12)	(-0.03)	(-0.29)	(-0.05)	(-0.05)	(-0.04)

Table 4: Ablation results of Gemini-1.5-Pro on different tasks of LOFT at 128k context length. Starting from our best prompt format (used in the rest of the experiments), individual facets of the corpus, query, and instruction are ablated to surface their relative effect on quality. †: The average is computed without ArguAna and FIQA, as not all ablations apply to them (they do not contain titles).

Results Results in Table 2 show that LCLMs achieve non-trivial performance, though they are significantly behind the text-to-SQL baseline. This reveals substantial headrooms to enhance the compositional reasoning capabilities of LCLMs.

Reasoning Analysis To gain insights into the short-comings of LCLMs in complex compositional reasoning, we categorize queries based on the operators in the gold SQL queries and measure Gemini-1.5-Pro's performance for each operator. Figure 6 shows that averaging is the most difficult operation while counting is relatively easy. Moreover, we find that reasoning over equality is considerably easier than reasoning over inequality.

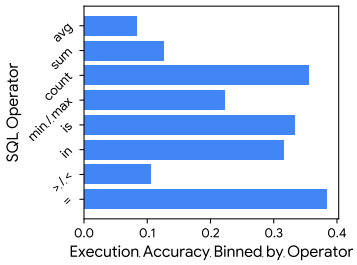

Figure 6: **SQL Reasoning Analysis**. We bin Spider queries by operators in their SQL query and report binned Gemini performance.

4.6 Many-Shot ICL

Results Table 2 compares accuracy for Gemini 1.5 Pro and GPT-4o on all ICL benchmarks. For BBH, we report the accuracy on 32k, which is the maximum context length available. Gemini 1.5 Pro outperforms GPT-4o on all benchmarks, except for BBH-tracking7 where Gemini performs surprisingly poorly.

Scaling Many Shot ICL Fig. 7 illustrates the impact of increasing the number of examples on performance. In LIB-dialog, accuracy improves monotonically with more examples. In contrast, results on BBH are mixed. Knowledge-intensive tasks like BBH-date and BBH-salient see monotonic improvements similar to LIB-dialog, while reasoning-intensive tasks like BBH-tracking7 and BBH-web do not benefit. These results suggests that building and updating mental models is harder to learn from scaling the number of in-context examples.

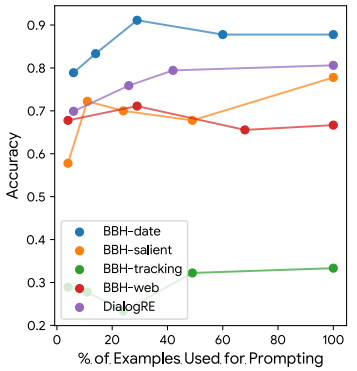

Figure 7: **ICL Performance** as we scale the percentage of examples used up to 100%.

5 CiC Prompt Ablations

We conduct ablations over the different facets of the CiC Prompt with ablated prompt examples in Appendix D. For the ablations, we evaluate Gemini-1.5-Pro at 128k context length.

The ablations show the effectiveness of our CiC prompting design. Removing tasks-specific instructions (`Generic Instruction`) or Chain-of-Thoughts reasoning (`Without CoT`) both lead to worse performance. We also observe performance decrease for `Corpus in Each Few-Shot`, where a small corpus (10 oracle and randomly passage) is added for each few shot example instead of using one shared corpus. Placing the query at the beginning of the prompt instead of the end (`Query at Beginning`) led to a significant and consistent performance decrease. This allows us to perform prefix-caching as we do not need to encode the corpus conditioned on the specific query.

For the document ID formatting, replacing monotonic numerical IDs with random (`Alphanumeric IDs`) negatively impacted performance in most datasets, possibly due to tokenizer being optimized for numerical values. Not repeating the ID at the end of the document (`Without ID Echo`) resulted in a 5% performance drop, confirming [39] that repeating text can compensate for missing context in autoregressive language models.

To test if model uses parametric knowledge instead of grounding on the context, we remove the document content and simply keep the document title and ID in the corpus (`Title Only`). Across all experiments, this ablation significantly degraded performance, indicating the model indeed relies provided context.

Finally, we study how the number of few-shot examples in the prompt affect quality in Figure 8. Increasing the number of examples increase quality overall on the retrieval task, from 0.76 at zero-shot to 0.81 at 5-shots.

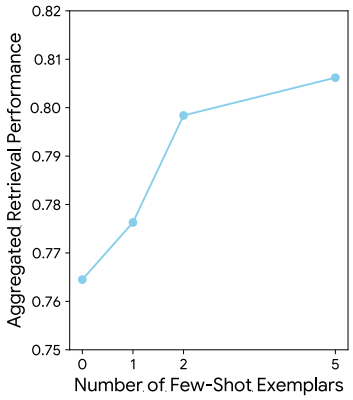

Figure 8: **Effect of the number of few-shot examples**. The performance increases with the number of few-shot examples.

6 Related Work

Evaluating long-context language models (LCLMs) remains a challenge due to the limitations of existing benchmarks. Many popular datasets and methods rely on synthetic tasks [41] such as the popular "Needle-in-A-Haystack" retrieval [19] or its extension to multi-hop QA [25]. While scalable to arbitrary lengths, these approaches do not fully capture the nuances of real-world retrieval or reasoning tasks [18]. Conversely, some recent benchmarks leverage existing NLP datasets for tasks such as extreme summarization and multi-document QA [6]. However, these lack the dynamic scaling capabilities of synthetic benchmarks.

LongAlpaca [10] and LongBench-Chat [5] evaluate instruction-following under long-text settings, while Ada-LEval [45] tests LCLMs on 100k+ tokens but with limited task diversity.

Closest to our work is [29], which applies LCLMs to long-context QA using top retrieved documents from MSMarco, similar to our RAG setup in LOFT. They find that LCLMs lose recall when relevant information is placed in the middle of the context (*i.e.,* lost-in-the-middle). However, their analysis is limited to contexts under 10k tokens. We extend the evaluation of LCLMs to up to 1M tokens context length and multiple modalities.

7 Conclusion

As language models improve and scale, their ability to retrieve and reason over increasing context lengths will unlock unprecedented use-cases. To measure this progress, we introduce LOFT, the Long Context Frontiers benchmark. LOFT is a suite of tasks that rigorously assesses LCLMs on tasks ripe for a paradigm shift: retrieval, retrieval-augmented generation, and SQL-like reasoning. LOFT provides dynamic scaling of context lengths, up to 1 million tokens, ensuring that evaluations remain relevant as LCLMs continue to evolve. Initial findings showcase that despite never trained to do retrieval, LCLMs have retrieval capabilities rivaling dedicated SOTA retrieval systems. Nevertheless, there remains considerable room for advancement in long-context reasoning, particularly as models gain access to even longer context windows. We believe that LOFT provides fertile testing ground for measuring progress in long-context modeling.

Limitations Our experiments were constrained by the speed, computational resources and financial costs associated with utilizing the long context language models. We were not able to measure the efficiency improvements from prefix caching [16] at the time of the experiments due to API constraints; without caching, Gemini-1.5-Pro API's median lantency is roughly 4 seconds on 32k token input, 12 seconds on 128k token input, and 100 seconds on 1m token input. Additionally, the scope of our retrieval and RAG tasks was limited to 1 million tokens, which still has a large gap towards real-world applications that may involve millions or even billions of documents.

References

[1] Vaibhav Adlakha, Parishad BehnamGhader, Xing Han Lu, Nicholas Meade, and Siva Reddy. Evaluating Correctness and Faithfulness of Instruction-Following Models for Question Answering. *Transactions of the Association for Computational Linguistics*, 12:681–699, 05 2024.

[2] Vaibhav Adlakha, Shehzaad Dhuliawala, Kaheer Suleman, Harm de Vries, and Siva Reddy. Topiocqa: Open-domain conversational question answering with topic switching. *Transactions of the Association for Computational Linguistics*, 10:468–483, 2021.

[3] Samuel Joseph Amouyal, Tomer Wolfson, Ohad Rubin, Ori Yoran, Jonathan Herzig, and Jonathan Berant. Qampari: A benchmark for open-domain questions with many answers. In *IEEE Games Entertainment Media Conference*, 2022.

[4] Anthropic. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*, 2024.

[5] Yushi Bai, Xin Lv, Jiajie Zhang, Yuze He, Ji Qi, Lei Hou, Jie Tang, Yuxiao Dong, and Juanzi Li. Longalign: A recipe for long context alignment of large language models, 2024.

[6] Yushi Bai, Xin Lv, Jiajie Zhang, Hong Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. Longbench: A bilingual, multitask benchmark for long context understanding. *ArXiv*, abs/2308.14508, 2023.

[7] Scott Barnett, Stefanus Kurniawan, Srikanth Thudumu, Zach Brannelly, and Mohamed Abdelrazek. Seven failure points when engineering a retrieval augmented generation system. *ArXiv*, abs/2401.05856, 2024.

[8] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[9] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *ArXiv*, abs/2005.14165, 2020.

[10] Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. Longlora: Efficient fine-tuning of long-context large language models, 2024.

[11] Hyung Won Chung, Le Hou, S. Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Wei Yu, Vincent Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed Huai hsin Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models. *ArXiv*, abs/2210.11416, 2022.

[12] Alexis Conneau, Min Ma, Simran Khanuja, Yu Zhang, Vera Axelrod, Siddharth Dalmia, Jason Riesa, Clara Rivera, and Ankur Bapna. Fleurs: Few-shot learning evaluation of universal representations of speech. *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 798–805, 2022.

[13] Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[14] Dawei Gao, Haibin Wang, Yaliang Li, Xiuyu Sun, Yichen Qian, Bolin Ding, and Jingren Zhou. Text-to-sql empowered by large language models: A benchmark evaluation. *ArXiv*, 2023.

[15] Frank Palma Gomez, Ramon Sanabria, Yun-hsuan Sung, Daniel Cer, Siddharth Dalmia, and Gustavo Hernandez Abrego. Transforming llms into cross-modal and cross-lingual retrievalsystems. In *Proceedings of the 21st International Conference on Spoken Language Translation (IWSLT 2024)*, 2024.

[16] Google. Context caching guide. https://ai.google.dev/gemini-api/docs/caching, 2024. Accessed: 2024-06-05.

[17] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Realm: Retrieval-augmented language model pre-training. *ArXiv*, abs/2002.08909, 2020.

[18] Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. Ruler: What's the real context size of your long-context language models? 2024.

[19] Greg Kamradt. Needle in a haystack - pressure testing llms, 2023.

[20] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Yu Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. Dense passage retrieval for open-domain question answering. *ArXiv*, abs/2004.04906, 2020.

[21] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *ArXiv*, abs/2205.11916, 2022.

[22] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, 2018.

[23] Haejun Lee, Akhil Kedia, Jongwon Lee, Ashwin Paranjape, Christopher D. Manning, and Kyoung-Gu Woo. You only need one model for open-domain question answering. In *Conference on Empirical Methods in Natural Language Processing*, 2021.

[24] Jinhyuk Lee, Zhuyun Dai, Xiaoqi Ren, Blair Chen, Daniel Cer, Jeremy R. Cole, Kai Hui, Michael Boratko, Rajvi Kapadia, Wen Ding, Yi Luan, Sai Meher Karthik Duddu, Gustavo Hernández Abrego, Weiqiang Shi, Nithi Gupta, Aditya Kusupati, Prateek Jain, Siddhartha R. Jonnalagadda, Ming-Wei Chang, and Iftekhar Naim. Gecko: Versatile text embeddings distilled from large language models. *ArXiv*, abs/2403.20327, 2024.

[25] Mosh Levy, Alon Jacoby, and Yoav Goldberg. Same task, more tokens: the impact of input length on the reasoning performance of large language models. *ArXiv*, abs/2402.14848, 2024.

[26] Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. *ArXiv*, abs/2005.11401, 2020.

[27] Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. Question and answer test-train overlap in open-domain question answering datasets. In *Conference of the European Chapter of the Association for Computational Linguistics*, 2020.

[28] Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. Long-context llms struggle with long in-context learning. *arXiv preprint arXiv:2404.02060*, 2024.

[29] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2023.

[30] S. Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. Entity-based knowledge conflicts in question answering. *ArXiv*, abs/2109.05052, 2021.

[31] Man Luo, Xin Xu, Zhuyun Dai, Panupong Pasupat, Mehran Kazemi, Chitta Baral, Vaiva Imbrasaite, and Vincent Zhao. Dr.icl: Demonstration-retrieved in-context learning. *ArXiv*, abs/2305.14128, 2023.

[32] Chaitanya Malaviya, Peter Shaw, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Quest: A retrieval dataset of entity-seeking queries with implicit set operations. *ArXiv*, abs/2305.11694, 2023.

[33] Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, et al. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9844–9855, 2022.

[34] Maxwell Nye, Anders Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. Show your work: Scratchpads for intermediate computation with language models. *ArXiv*, abs/2112.00114, 2021.

[35] OpenAI. Gpt-4 technical report. *ArXiv*, 2023.

[36] Ethan Perez, Patrick Lewis, Wen tau Yih, Kyunghyun Cho, and Douwe Kiela. Unsupervised question decomposition for question answering. In *Conference on Empirical Methods in Natural Language Processing*, 2020.

[37] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021.

[38] Nils Reimers and Iryna Gurevych. The curse of dense low-dimensional information retrieval for large index sizes. In *Annual Meeting of the Association for Computational Linguistics*, 2020.

[39] Jacob Mitchell Springer, Suhas Kotha, Daniel Fried, Graham Neubig, and Aditi Raghunathan. Repetition improves language model embeddings. *arXiv preprint arXiv:2402.15449*, 2024.

[40] Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, and Adrià Garriga-Alonso et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *ArXiv*, abs/2206.04615, 2022.

[41] Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. Long range arena: A benchmark for efficient transformers. *ArXiv*, abs/2011.04006, 2020.

[42] Gemini Team. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *ArXiv*, abs/2403.05530, 2024.

[43] Nandan Thakur, Nils Reimers, Andreas Ruckl'e, Abhishek Srivastava, and Iryna Gurevych. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *ArXiv*, abs/2104.08663, 2021.

[44] H. Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2021.

[45] Chonghua Wang, Haodong Duan, Songyang Zhang, Dahua Lin, and Kai Chen. Ada-leval: Evaluating long-context llms with length-adaptable benchmarks, 2024.

[46] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. *ArXiv*, abs/2109.01652, 2021.

[47] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Huai hsin Chi, F. Xia, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903, 2022.

[48] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *ArXiv*, abs/2007.00808, 2020.

[49] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing*, 2018.

[50] Dian Yu, Kai Sun, Claire Cardie, and Dong Yu. Dialogue-based relation extraction. *ArXiv*, abs/2004.08056, 2020.

[51] Tao Yu, Rui Zhang, Kai-Chou Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Z Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir R. Radev. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. *ArXiv*, abs/1809.08887, 2018.

[52] Tao Yu, Rui Zhang, Michihiro Yasunaga, Yi Chern Tan, Xi Victoria Lin, Suyi Li, He Yang Er, Irene Z Li, Bo Pang, Tao Chen, Emily Ji, Shreya Dixit, David Proctor, Sungrok Shim, Jonathan Kraft, Vincent Zhang, Caiming Xiong, Richard Socher, and Dragomir R. Radev. Sparc: Cross-domain semantic parsing in context. In *Annual Meeting of the Association for Computational Linguistics*, 2019.

[53] Victor Zhong, Caiming Xiong, and Richard Socher. Seq2SQL: Generating structured queries from natural language using reinforcement learning. *ArXiv*, 2017.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes. Our results in Section 4 back up the claims made in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Yes. We dedicate an entire section to the limitations of our data and the methodology we test.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: There are no theoretical results in the paper.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Yes. Section 2 describes our dataset creation process at a high level and Appendix A delves into more details on how we selected the individual datasets to be a part of LOFT. We also plan to release the code reproduce the data in LOFT.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The datasets that make up LOFT are all open-source already, therefore it is possible to reproduce LOFT approximately using the details in the paper. At the moment, we are cleaning up our data generation pipeline. We will soon open-source our data-generation pipeline so that the data in LOFT is exactly reproducible.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: There is no training and all testing is done via API through prompting which we detail in Section 3 with additional prompting details in the Appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Given the fact that our evaluation of baselines on LOFT were done by using the APIs of several companies hosting large language models, we were constrained via time and budget, thus making doing multiple runs to get error bars prohibitively expensive.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Because our experiments are done via LLM APIs, we do not report information on compute resources for these models as this is proprietary information. We do provide execution times for our evaluations.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The paper conforms with the NeurIPS Code of Ethics

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: Our paper, while being a dataset paper, does not introduce any new data itself, rather repackages existing data to explore a new paradigm of prompting with models that already exist. Therefore, we do not introduce any new data itself or any new models, and thus we feel that the potential for harm from our work is low.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: This paper does not pose a safety risk as it does not introduce a new model new does it create brand new data. Rather it packages existing datasets that are well-established in the machine learning community to test a new paradigm of long-context modeling.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: Mostly [No] at time of submission but shortly will be Fully [Yes]

Justification: We cite the papers associated with all datasets used in LOFT. We have compiled licenses for all datasets, and will update the paper to include these licenses in the appendix shortly.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: This paper does not introduce any new assests, as it is a reformulation of existing data.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.