Can LLMs Help Uncover Insights about LLMs? A Large-Scale, Evolving Literature Analysis of Frontier LLMs

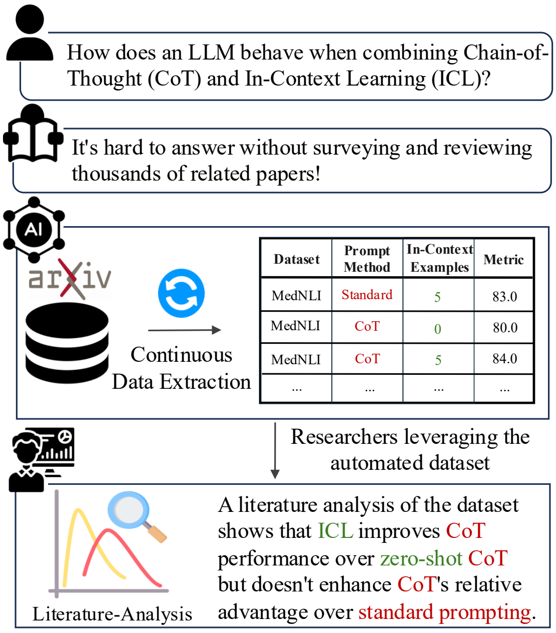
Jungsoo Park[1] **Junmo Kang**[1] **Gabriel Stanovsky**[2] **Alan Ritter**[1]
Georgia Institute of Technology[1] The Hebrew University of Jerusalem[2]
jpark3272@gatech.edu junmo.kang@gatech.edu
gabriel.stanovsky@mail.huji.ac.il alan.ritter@cc.gatech.edu

Abstract

The surge of LLM studies makes synthesizing their findings challenging. Analysis of experimental results from literature can uncover important trends across studies, but the time-consuming nature of manual data extraction limits its use. Our study presents a semi-automated approach for literature analysis that accelerates data extraction using LLMs. It automatically identifies relevant arXiv papers, extracts experimental results and related attributes, and organizes them into a structured dataset, LLMEVALDB. We then conduct an automated literature analysis of frontier LLMs, reducing the effort of paper surveying and data extraction by more than 93% compared to manual approaches. We validate LLMEVALDB by showing that it reproduces key findings from a recent manual analysis of Chain-of-Thought (CoT) reasoning and also uncovers new insights that go beyond it, showing, for example, that in-context examples benefit coding & multimodal tasks but offer limited gains in math reasoning tasks compared to zero-shot CoT. Our automatically updatable dataset enables continuous tracking of target models by extracting evaluation studies as new data becomes available. Through LLMEVALDB and empirical analysis, we provide insights into LLMs while facilitating ongoing literature analyses of their behavior.

1 Introduction

The rapid advancement of Large Language Models (LLMs) (Brown et al., 2020; Chowdhery et al., 2022; Achiam et al., 2023; Touvron et al., 2023; Team et al., 2023; Anthropic, 2024) has led to a proliferation of empirical studies. Recent *"surveys of surveys"* (Lv et al., 2024; Ruan et al., 2024) highlight the overwhelming growth of LLM publications, surpassing what individual researchers can manually review. The growing number of evaluations—each using different models, datasets, and prompting configurations—makes it challenging to analyze and synthesize findings across studies, underscoring the urgent need for automated analysis.


How does an LLM behave when combining Chain-of-Thought (CoT) and In-Context Learning (ICL)?
It's hard to answer without surveying and reviewing thousands of related papers!
AI
arXiv
Continuous Data Extraction
Researchers leveraging the automated dataset
Literature-Analysis
A literature analysis of the dataset shows that ICL improves CoT performance over zero-shot CoT but doesn't enhance CoT's relative advantage over standard prompting.

Dataset	Prompt Method	In-Context Examples	Metric
MedNLI	Standard	5	83.0
MedNLI	CoT	0	80.0
MedNLI	CoT	5	84.0
...	...	...	...

Figure 1: The diagram of our semi-automated literature analysis process and a key finding derived from data automatically extracted from the arXiv database.

A manual literature analysis that examines data from multiple independent studies to identify overall trends, offers a solution (Reiter, 2018; Hupkes et al., 2023). However, conducting such an analysis is time-consuming (Yun et al., 2023, 2024). For instance, as shown in Fig. 1, understanding the effects of combining Chain-of-Thought (CoT) with in-context examples requires identifying relevant papers, extracting specific experimental results, and aggregating the data. Furthermore, analyses based on existing studies are likely to become quickly outdated in a fast-moving field, motivating the need for automated analyses that can be continuously updated as new results are shared on preprint servers such as arXiv.org.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 20412–20433
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

In this study, we conduct the most extensive quantitative literature analysis of frontier LLMs to date. Our dataset, LLMEVALDB, comprises 18,127 experimental records extracted from 1,737 papers, along with 359 referenced papers describing datasets used in these experiments. LLMEVALDB can also be dynamically updated with minimal human effort to incorporate results from new models or studies as they are published. This comprehensive dataset enables us to present new insights into LLM performance, where previous studies on extracting leaderboards from machine learning papers have primarily focused on improving data extraction accuracy (Kardas et al., 2020; Yun et al., 2024).

To achieve this, we experiment with an automated approach to data extraction that efficiently processes research literature. This approach uses an LLM to scan the arXiv database, identify relevant papers with experimental results on target models, and extract experimental results along with pertinent attributes. This reduces the manual effort needed for surveying and extraction by more than 93% compared to expert manual data extraction. The approach employs schema-driven extraction (Bai et al., 2023) and context augmentation from paper contents to capture essential attributes for a comprehensive understanding of evaluation results. Additionally, the process creates dataset descriptions that summarize key characteristics of the dataset associated with the evaluation record, enhancing LLMEVALDB's potential utility in literature analysis.

Our analysis shows that LLMEVALDB can efficiently replicate a previous manual analysis (Sprague et al., 2024), which found that CoT reasoning primarily improves performance on mathematical and symbolic reasoning benchmarks. We also identify three novel insights from the dataset on prompting configurations. First, in-context learning (ICL) enhances coding & multimodal tasks but offers limited benefits for mathematical tasks compared to a zero-shot setup. Second, our qualitative analysis reveals specific characteristics of datasets that tend to reduce the effectiveness of CoT prompting and ICL. Third, we find that CoT prompting with demonstrations—commonly implemented through ICL—typically yields better performance than CoT without demonstrations. However, the relative improvement of CoT over standard direct prompting remains stable regardless of the number of demonstrations used (i.e., it holds true in both zero-shot and few-shot settings). We release our dataset and source code to support the ongoing automated integration of new findings in the literature. [1]

2 Data Extraction

Automated data extraction is essential for scaling and improving literature analysis efficiency. This section outlines our pipeline for extracting experimental results and model attributes from arXiv sources, which are used to construct LLMEVALDB. We define target models and attributes (§2.1) and introduce our LLM-powered extraction process, which includes three stages: preprocessing & filtering (§2.2), extraction & augmentation (§2.3), and description generation (§2.4). Unless stated otherwise, we utilize GPT-4o (Hurst et al., 2024) throughout the pipeline.

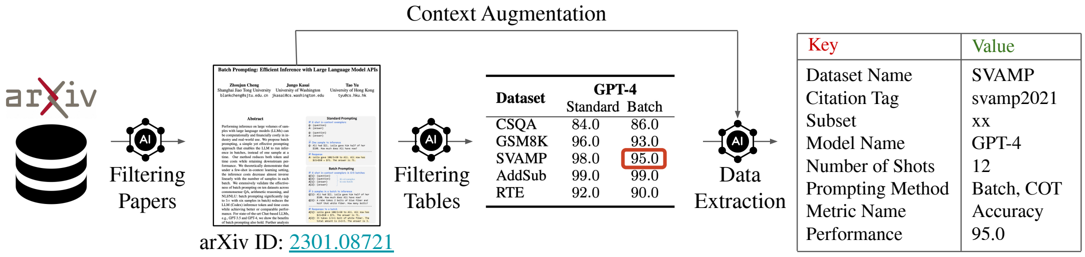
2.1 Defining Target Models and Attributes

Target Models We analyze four leading proprietary models in the NLP/AI field: GPT-4 (Achiam et al., 2023), GPT4-o (Hurst et al., 2024), Claude3 Opus (Anthropic, 2024), and Gemini 1.0 Pro (Team et al., 2023). Our analysis focuses on proprietary LLMs accessible via APIs that are not possible to fine-tune.[2] We chose to exclude fine-tuned models from our study because comparing diverse fine-tuning methods would make controlled cross-study comparisons much more challenging. We also exclude recently released advanced reasoning models (GPT4-o1 (OpenAI, 2024) and Deepseek-R1 (Guo et al., 2025)) due to very limited published results to date. In the future, our pipeline can readily extract experimental data for these models along with those of other target models, as more studies are published. For further discussion, see §7.

Target Attributes To enable thorough analysis across different studies, we extract various performance-related fields from proprietary models. The targeted attributes include: *Dataset Name*, *Subset*, *Model Name*, *Prompting Method*, *Number of Demonstrations*, *Metric Name*, and *Performance*. Note that the subset refers to any division of the dataset, such as a subtask, domain,

[1] The code is available at `https://github.com/JJumSSu/meta-analysis-frontier-LLMs`, and the dataset can be accessed at `https://huggingface.co/datasets/jungsoopark/LLMEvalDB`.

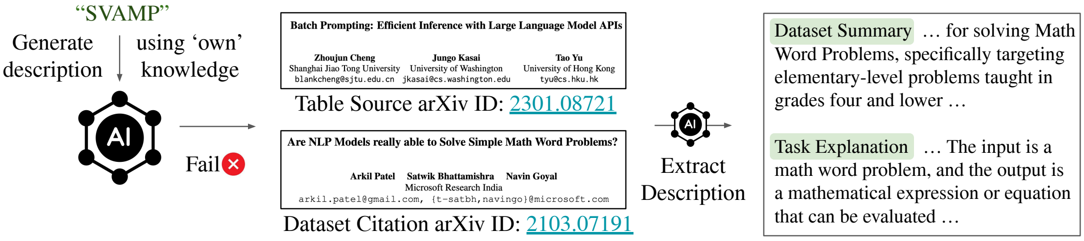
[2] GPT-4o and Gemini 1.0 Pro became available for fine-tuning during our study period, but we used rule-based heuristics to filter out limited fine-tuned results.



Dataset	GPT-4 Standard	GPT-4 Batch
CSQA	84.0	86.0
GSM8K	96.0	93.0
SVAMP	98.0	95.0
AddSub	99.0	99.0
RTE	92.0	90.0

Key	Value
Dataset Name	SVAMP
Citation Tag	svamp2021
Subset	xx
Model Name	GPT-4
Number of Shots	12
Prompting Method	Batch, COT
Metric Name	Accuracy
Performance	95.0

Figure 2: Data extraction pipeline overview with an extracted example from arXiv paper (Cheng et al., 2023). Target attributes are identified and extracted from the table, augmented with paper content.



Batch Prompting: Efficient Inference with Large Language Model APIs

Zhoujun Cheng Shanghai Jiao Tong University blankcheng@sjtu.edu.cn
Jungo Kasai University of Washington jkasai@cs.washington.edu
Tao Yu University of Hong Kong tyu@cs.hku.hk

Are NLP Models really able to Solve Simple Math Word Problems?

Arkil Patel **Satwik Bhattamishra** **Navin Goyal**
Microsoft Research India
arkil.patel@gmail.com, {t-satbh,navingo}@microsoft.com

Dataset Summary … for solving Math Word Problems, specifically targeting elementary-level problems taught in grades four and lower …

Task Explanation … The input is a math word problem, and the output is a mathematical expression or equation that can be evaluated …

Figure 3: Dataset description generation pipeline overview using the example of "SVAMP" dataset (Patel et al., 2021). Since LLM lacked knowledge of the given dataset, the model references the original dataset's arXiv source retrieved through extracted BibTex. If confident, however, the LLM's generated descriptions will be used.

split, or language pair. For instance, as shown in Fig.2, the arXiv paper with ID *2301.08721* (Cheng et al., 2023) presents experimental results for *GPT-4* (Achiam et al., 2023) on the *SVAMP* dataset (Patel et al., 2021). The evaluation used *Batch Prompting with CoT* (Wei et al., 2022; Cheng et al., 2023) and *12 in-context samples* (Brown et al., 2020), achieving an *accuracy* score of *95.0*.

2.2 Preprocessing & Filtering

To extract the experimental results of target models and attributes, we used arXiv sources from January 2023 to December 2024, focusing on machine learning papers (cs.CV, cs.AI, cs.CL, cs.LG). arXiv LaTeX sources preserve structure better than parsed PDFs, enabling easier dataset extraction—especially from tables—using simple regex, while PDFs require complex tools. Hence, following prior work (Kardas et al., 2020; Bai et al., 2023; Wang et al., 2024), we adopt arXiv LaTeX sources for data extraction, as over 90% of arXiv papers are submitted in LaTeX format (Frankston et al., 2024).

We downloaded LaTeX source files and applied regex-based methods to extract structured data like tables. Before extraction, we filtered tables to reduce time and API costs by selecting those with target model results. We also concentrated on leaderboard tables presenting benchmark results following Kardas et al. (2020) and used Llama3.1-70B-Instruct (Dubey et al., 2024) for the filtering.

2.3 Extraction & Augmentation

Based on the selected tables, we extract records from tables and paper contents. Using schema-driven extraction, we obtain records matching target attributes from tables, focusing only on relevant model data to save costs while maintaining accuracy. This selective approach significantly reduces API costs compared to extracting full results from each table (Bai et al., 2023). After table extraction, we augment records with context from the entire paper, adding experimental specifics. During this stage, we also gather BibTeX references for datasets to link and retrieve the original papers describing these datasets from arXiv.

2.4 Dataset Description Generation

In addition to extracting structured representations of experiments, we also generate summary descriptions of the relevant tasks and datasets (Fig. 3) that can aid in-depth literature analysis, such as classifying records by research subarea. We use a two-stage approach to create these descriptions. Initially, the LLM generates descriptions using its internal knowledge based on the dataset name and subset, which is cost-effective. When the LLM is uncertain and refuses to answer, we prompt it to extract descriptions using the full content of the

ArXiv ID	Table	Dataset	Subset	Dataset Description	Model	Shots	Prompt	Metric	Value
2301.08721	2	CommonsenseQA	xx	**Dataset Summary** This dataset is a multiple-choice question answering dataset focused on commonsense knowledge...	GPT-4	12	Batch CoT	Acc	86
2408.02718	3	MMIU	Overall	**Dataset Summary** This dataset is designed for the domain of multimodal understanding, specifically focusing on tasks that require the integration of information from multiple images...	GPT-4o	xx	xx	Accuracy	55.7
2405.02861	3	LexBench	IED	**Dataset Summary** This dataset is part of a comprehensive evaluation suite designed to test language models on various semantic phrase processing tasks...	Claude3	0	zero-shot	Accuracy	66.3
2409.19667	5	ProGraph	BGT	**Dataset Summary** This dataset is a benchmark designed to evaluate the capability of large language models (LLMs) in graph analysis tasks...	Gemini1.0	xx	xx	Accuracy	27.7

Table 1: Sampled dataset instances. 'xx' indicates missing values from the paper. Dataset descriptions, including summaries, tasks, and subsets, are abbreviated due to space limits. Full versions are in appendix D.

Metric	Value
General Statistics	
Total Records	18,127
Number of Unique Datasets	2,984
Number of Table Source Papers	1,737
Number of Unique Tables	2,694
Records per Model	
GPT-4 (Achiam et al., 2023)	12,475
GPT-4o (Hurst et al., 2024)	4,589
Claude3-Opus (Anthropic, 2024)	661
Gemini1.0-Pro (Team et al., 2023)	402
Missing Values ('xx')	
Records missing *Subset*	2,892
Records missing *Prompting Method*	5,489
Records missing *Number of Few-Shots*	9,081
Dataset Description Sources	
Source Paper of the Table	9,334
GPT-4o (Hurst et al., 2024)	7,199
Source Paper of the Linked Dataset	1,594

Table 2: Dataset Statistics Overview. Dataset description sources specify whether descriptions come from LLM internal knowledge or source papers.

source papers as reference. The source papers can be either the paper containing the table or the original dataset paper cited within it, which is retrieved using the BibTex references obtained from §2.3.

3 Comprehensive Analysis of the Dataset

Table 1 presents sampled instances of LLMEvalDB, while the dataset's statistics are summarized in Table 2. A total of 18,127 experimental records of four target models were extracted from scanning over 300k arXiv source papers. GPT-4 (Achiam et al., 2023) and GPT-4o (Achiam et al., 2023) dominate the results, while Claude 3 Opus (Anthropic, 2024) and Gemini 1.0 Pro (Team et al., 2023) have significantly fewer entries, highlighting a bias favoring certain proprietary models. Many prompting configurations have missing values, confirmed by human verification (§3.1) as unreported by the original authors, likely indicating the default setup. The majority of dataset descriptions are generated referencing the source paper of the table, as these papers often describe datasets when presenting benchmarks or detailing experimental sections.

3.1 Quality and Efficiency Assessment

Our human evaluation assessed extraction accuracy and description quality across 40 records of LLMEvalDB (280 total attributes) from different papers. Two NLP Ph.D. students verified field extraction correctness (Bai et al., 2023) and rated descriptions on a 5-point scale from (1) unrelated to (5) fully relevant and accurate (Amidei et al., 2019; Liu et al., 2023a). Missing values were marked correct if the information was genuinely unavailable in the source papers. More details are in appendix B.

Table 3 shows strong extraction accuracy and description quality of LLMEvalDB. GPT-4o (Hurst

Attribute	Accuracy (Score)
Extraction	
Dataset Name	95 %
Model Name	100 %
Prompting Method	86.3 %
Number of Few-Shot Examples	95 %
Metric	100 %
Metric Value	98.8 %
Description	
Dataset Description	4.55

Table 3: Results of Human Evaluation.

et al., 2024), which we used during the data extraction, demonstrated effective long-context information extraction, showcasing its potential for scientific literature synthesis. Moreover, high validation scores suggest missing values stem from unreported setups. Cohen's Kappa scores of 0.68 (extraction) and 0.57 (description) indicate substantial inter-annotator agreement (McHugh, 2012).

During the study, we recorded the time experts spent annotating target attribute information (excluding descriptions) from additional samples based on a list of papers and tables indexing experimental results of target models. On average, it took 7 minutes and 50 seconds per table, totaling approximately about 350 hours for LLMEVALDB of 2,694 tables. This does not include the initial effort of surveying and identifying papers and tables. In contrast, our pipeline identifies and extracts data from arXiv sources from 2023 to 2024 in a single day for under $500 using a batching API.

3.2 Categorization by Required Skills

We categorize experimental records from LLMEVALDB by required skills to enable comprehensive literature analyses and offer researchers a searchable resource. Expanding on Tulu3's core skills (Lambert et al., 2024), we defined 10 categories: *Knowledge*, *Reasoning*, *Math*, *Coding*, *Multimodality*, *Instruction Following*, *Safety*, *Multilinguality*, *Tool Use (Agent Framework)*, and *Other*. Records were classified using an LLM API (Hurst et al., 2024) based on dataset names, subsets, and descriptions. Since datasets can fit multiple categories, we applied multi-label classification. This categorized information is used throughout the analyses in §4.

Figure 4 shows the log-scale frequency of skill categories evaluated every three months based on arXiv publication dates. Over time, experimental results for all skills have increased due to rising interest in LLMs. *Reasoning* tasks are the most popular and continue to grow, while *Multimodality* has shown recent growth. *Knowledge*, *Multilinguality*, and *Math* have steadily increased in evaluation rates. Frequency is based on the unique count of dataset-subsets aggregated by paper. The most frequently used datasets for each category are listed in appendix E.

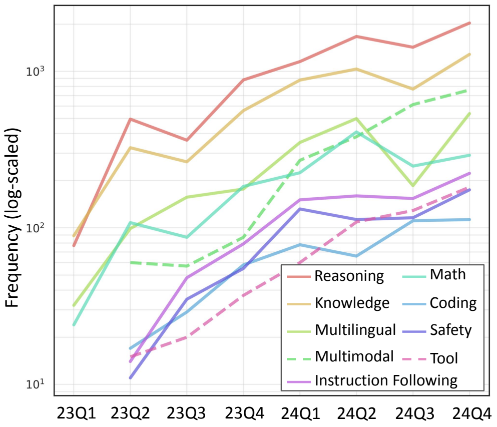


Figure 4: Log-scale frequency trends show a rapid increase of evaluation studies over successive quarters (Q). *Reasoning* remains the most popular evaluation category, while *Multimodality* exhibits recent rapid growth.

4 Prompting Behavior in Frontier LLMs: A Literature Review

LLMEVALDB enables partially automated literature analyses of LLM-prompting behaviors through its structured attributes. We validate our semi-automated approach to literature analysis by replicating Sprague et al. (2024)'s manual analysis, confirming CoT's advantages over direct prompting in mathematical and symbolic reasoning tasks (§4.1). We then show that LLMEVALDB enables a fine-grained analysis, showing that in-context examples boost performance in coding & multimodal tasks but not in mathematical ones (§4.2). We qualitatively analyze dataset characteristics, such as required expert knowledge, that may negatively affect the performance of CoT and ICL (§4.3). Moreover, we analyze CoT and ICL interactions, finding that while demonstrations enhance CoT performance, they do not affect CoT's relative improvement over standard direct prompting (§4.4). Finally, we analyze a subset of peer-reviewed papers published in major journals or conferences, revealing consistent trends that further support the robustness and generalizability of our findings (§4.5).

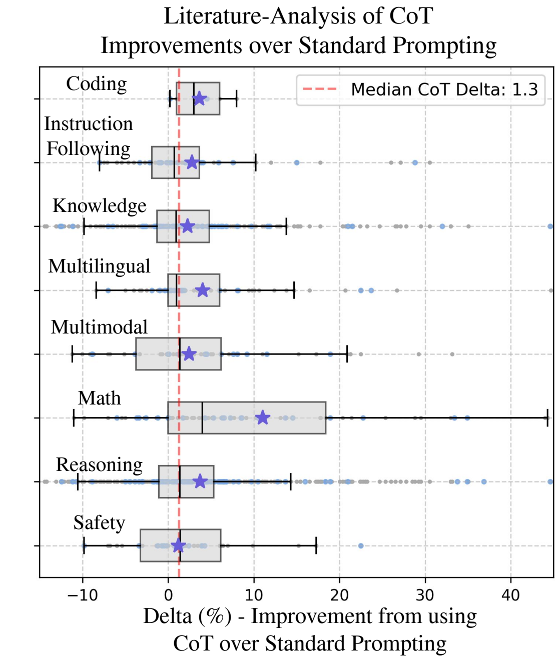


Figure 5: CoT shows significant performance improvement over direct prompting in mathematical tasks, whereas its impact on reasoning tasks is less distinct due to their complexity and diversity. Grey dots indicate individual deltas (improvements), blue dots represent the mean delta per paper, and a purple star marks the mean delta for each category.

4.1 Which Tasks Benefit from CoT?

Motivation and Setup CoT (Wei et al., 2022), a prompting technique for eliciting reasoning, has attracted considerable attention, with extensive literature on the subject (Wang et al., 2023; Yao et al., 2024) Recently, Sprague et al. (2024) conducted a manual analysis, concluding that *CoT improves over standard direct prompting primarily in mathematical and symbolic reasoning tasks*. We extract instances from LLMEvalDB that meet the criteria for replicating their investigation.

This identification is achieved through filtering using the dataset's attributes. We focus on instances from the same source paper and table, identifying experiments that feature both CoT and standard direct prompts under the same conditions (model, dataset, subset, metric, and few-shot setup). Human input is used only to identify whether a prompt is a CoT prompt or a standard direct prompt based on the *prompting method* attribute; this is the only manual effort required for the analysis aside from implementing the analysis itself. We exclude CoT variations like "xx of Thoughts" (Yao et al., 2024) and CoT-SC (Wang et al., 2022). We then calculate CoT's performance improvement over standard direct prompting, namely *delta*, where a positive delta indicates CoT outperforming direct prompting, and a negative delta indicates the reverse.

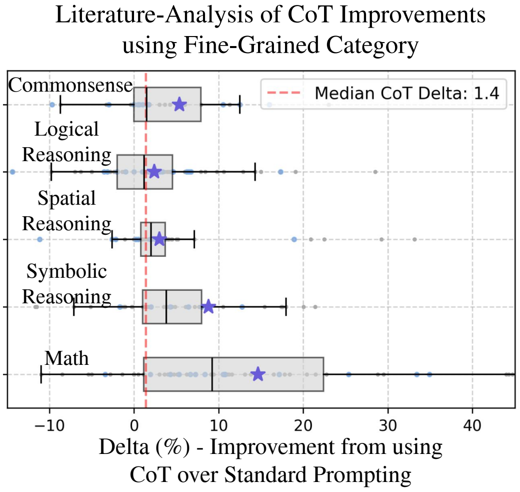


Figure 6: In the reasoning category, CoT shows notable improvement over direct prompting in *Symbolic Reasoning* tasks compared to other reasoning tasks. The figure displays an abbreviated version of our study, using categories defined by Sprague et al. (2024).

Analysis Our analysis, shown in Fig.5, reveals that CoT significantly enhances performance in mathematical tasks, with both median and mean improvements surpassing the overall median. However, we do not see clear improvements for reasoning tasks, likely due to their diversity, such as commonsense and logical reasoning.

To investigate further, we apply more detailed classification categories (but less general) from Sprague et al. (2024) and redo our study using LLM classification. The truncated results, presented in Fig. 6, clearly show that symbolic reasoning and mathematical tasks exhibit notable improvement over other reasoning and non-reasoning tasks. This confirms the reliability and efficiency of LLMEvalDB for literature analyses, as it provides consistent results with those obtained through manual curation in previous studies. Statistical tests, as shown in *Total Results* from Table. 7, confirm these patterns.

4.2 Which Tasks Benefit from ICL?

Motivation and Setup Previous research has systematically explored ICL behavior (Min et al., 2022; Agarwal et al., 2024; Wei et al., 2023; Bertsch et al., 2024). In line with our study in §4.1, we analyze the improvement of using in-context examples compared to not using any. To achieve this,

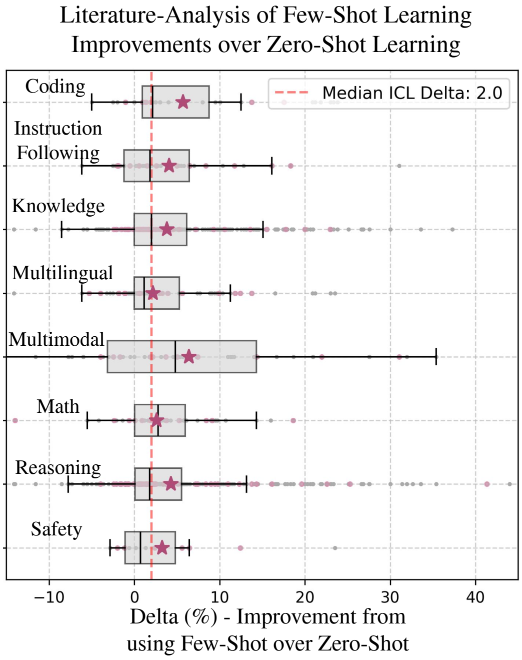


Figure 7: Few-shot learning significantly improves performance over zero-shot learning in *coding* and *multimodal* tasks, whereas its impact on math tasks is less pronounced compared to CoT, as shown in Fig. 5. Grey dots represent individual deltas (improvements), sky pink dots show the mean delta per paper, and a pink star marks the mean delta for each category.

we extract instances from LLMEVALDB that come from the same papers and tables comparing few-shot and zero-shot setups under identical conditions (i.e., model, dataset, subset, metric, and prompting method). No manual labor was needed beyond implementation, as the *number of in-context examples* is an integer value that can be easily computed for filtering. We measure performance deltas between few-shot and zero-shot setups, where positive values indicate few-shot advantages and negative values indicate the reverse.

Analysis Figure 7 presents the results. In contrast to CoT's strong performance in mathematical reasoning, ICL shows only modest benefits for math tasks. However, ICL demonstrates more substantial improvements in coding and multimodal applications despite considerable performance variability across different cases. This variance may be due to the broad scope of multimodal tasks, including speech and image processing. ICL shows more uniform improvements over zero-shot performance across different categories compared to the study in Fig. 5 (std: 8.3). Note that we did not account for the number of in-context examples used; we only distinguished between the use of in-context examples and zero-shot learning. We also plot the performance improvement distribution of more versus fewer in-context examples in appendix C and find a similar overall median. This suggests that the presence of demonstrations is more important than their quantity.

Key Characteristic	CoT Ratio	ICL Ratio
Expert Knowledge	31.6%	31.0%
Faithfulness	20.9%	4.6%
Complex Reasoning	17.9%	13.8%
Information Synthesis	9.7%	10.3%
Cognitive Tasks	8.7%	-
Affective Analysis	7.1%	13.8%
Structured Prediction	-	12.6%
Other	2.5%	13.8%

Table 4: A qualitative analysis of dataset characteristics that resulted in decreased performance with CoT or ICL, compared to direct prompting or zero-shot set-up. Key characteristics refer to the representative traits identified through qualitative analysis. The CoT and ICL ratios indicate the proportion of these traits among all negative cases for each method.

4.3 Which Dataset Characteristics Negatively Affect CoT and ICL?

Motivation and Setup To comprehend the patterns of performance degradation, we analyze cases where performance declined: specifically, where CoT resulted in worse outcomes than direct prompting and where few-shot learning performed worse than zero-shot learning (i.e. deltas below zero). Table 4 summarizes the key dataset characteristics associated with performance declines under both approaches based on analyzing the descriptions.

Analysis Out of the cases of performance decline, tasks requiring expert-level knowledge showed the highest ratio for both cases, approximately 31%, indicating that knowledge-intensive tasks do not benefit much from different prompting configurations alone. Sprague et al. (2024) notes a similar point that apparent performance gains in knowledge tasks mainly stem from "reasoning" or "math" components in datasets like MMLU (Hendrycks et al., 2020). This also aligns partly with Liu et al. (2024), who found that CoT can hinder LLM performance on tasks requiring minimal deliberative thinking.

The two approaches show different patterns in other aspects. CoT significantly degraded performance in faithfulness and factual verification

Performance comparison	Median	Q1	Q3
Few-shot CoT − Zero-shot CoT	3.0	0.4	9.2

Table 5: Effect of in-context demonstrations *within* Chain-of-Thought prompting.

Demonstration level	Δ(Median)	Δ(Q1)	Δ(Q3)
Zero-shot	+1.3	−0.4	4.7
Few-shot	+0.9	−1.2	3.7

Table 6: The performance comparison (Δ) between CoT and standard prompting under matched demonstration settings. A positive Δ indicates that CoT outperforms standard prompting.

tasks, possibly due to introduced reasoning biases (Shaikh et al., 2023; Chua et al., 2024). In contrast, demonstrations led to notable declines in affective analysis and structured prediction tasks, where examples may bias the model toward imitating specific patterns rather than genuine reasoning. This also partially aligns with the finding from Zhang et al. (2022a) that demonstrations increase the confidence of model predictions on captured superficial patterns, possibly incurring biases for structure prediction tasks such as NER.

4.4 How do CoT and ICL Interact?

Motivation and Setup Building on previous work sections 4.1 and 4.2, we ask two related questions about the joint behavior of CoT and ICL:

1. **Do demonstrations boost performance inside the CoT framework?** We compare few-shot and zero-shot settings while keeping CoT prompting constant (Table 5), replicating the manual study from Wei et al. (2022) through our literature analysis.

2. **Do demonstrations enable CoT to beat standard prompting?** We compare CoT and standard prompting under the same demonstration level (Table 6).

Analysis

1. **Demonstrations within CoT.** Table 5 shows that adding demonstrations yields a median gain of **+3.0 pts** which is significant. This replicates the uplift reported by (Wei et al., 2022; Zhang et al., 2022b).

2. **Demonstrations do *not* "rescue" CoT.** Table 6 shows that, whether we include demonstrations or not, CoT fails to open a meaningful lead over standard prompting. The median gains are modest— **+0.9 pts** in the few-shot setting and **+1.3 pts** in zero-shot. In other words, demonstrations lift both prompting styles in parallel but do not turn CoT into the clearly better option.

4.5 How Do Trends in Peer-Reviewed Papers Compare to Those in arXiv Publications?

Motivation and Setup LLMEVALDB includes information from papers published on arXiv. To examine whether the trends observed in our general analysis align with those in more selective venues, we conducted a focused analysis on a subset of peer-reviewed papers accepted at journals or conferences, using metadata from DBLP.[3] We report statistical significant test results from 4.1 using a subset of peer-reviewed papers. Following Sprague et al. (2024), we perform a one-sided bootstrap test to assess whether each category shows performance gains (mean improvement > 0).

Analysis Table 7 represents the results. The statistically significant test results from our **Total Results** and **Filtered Results** indicate that math and symbolic reasoning tasks derive substantial benefit from using Chain-of-Thought (CoT) prompting, which aligns with both manual analysis (Sprague et al., 2024) and our examination in §4.1. We can see that the core finding remains consistent even if we used a subset of peer-reviewed papers. This invariance in analysis patterns extends to other examinations like those in §4.4, with additional details provided in Appendix C. These consistent results across both the complete dataset and the filtered selection of high-quality published papers serve to corroborate our findings and implicitly confirm our hypothesis.

5 Related Work

Information Extraction Previous works have focused on extracting basic result tuples (e.g., task, dataset, metric) from scientific literature (Singh et al., 2019; Hou et al., 2019; Kardas et al., 2020; Yang et al., 2022; Bai et al., 2023; Singh et al., 2024; Şahinuç et al., 2024; Kabongo et al., 2024). Our extraction pipeline improves upon this approach in two significant ways: it extracts enriched tuples that include prompting-related attributes and

[3] https://dblp.org/

Category	Total Results			Filtered Results		
	Mean Δ	p-value	Significant	Mean Δ	p-value	Significant
Math	14.61	0.0000	Yes	13.53	0.0000	Yes
Symbolic and algorithmic	8.85	0.0002	Yes	9.13	0.0000	Yes
Spatial and temporal reasoning	3.03	0.0166	No	2.07	0.0056	No
Logical reasoning	2.39	0.0084	No	1.18	0.3776	No
Commonsense reasoning	5.41	0.0450	No	5.61	0.0748	No
Multi-hop QA	2.05	0.0000	Yes	1.21	0.0064	No
Context-aware QA	2.45	0.0014	Yes	0.28	0.7060	No
Encyclopedic knowledge	2.18	0.0076	No	3.31	0.0138	No
Generation	4.24	0.0280	No	1.92	0.3920	No
Text classification	1.01	0.4464	No	0.27	0.8644	No
Entailment	0.81	0.2070	No	0.93	0.1666	No

Table 7: Statistical test results across different categories for replicating the study from Sprague et al. (2024). **Total Results** refers to the results without filtering any papers. **Filtered Results** refer to the results from the filtered papers that are published in peer-reviewed journals & conferences. Mean Δ shows the average improvement when using CoT over standard prompting, and significance is determined at $p = 0.00227$ after applying a Bonferroni correction.

generates detailed dataset descriptions by leveraging LLM and automatically linked source papers. Hence, unlike previous works that primarily compiled leaderboard tables, our enhanced extraction pipeline enables deeper review analysis, contributing to the broader goal of AI-driven scientific discovery (Xu et al., 2021; Majumder et al., 2024; M. Bran et al., 2024).

LLM & Prompting Our study focuses on extracting experimental results of frontier proprietary LLMs (Achiam et al., 2023; Anthropic, 2024; Team et al., 2023), with a specific emphasis on target attributes that incorporate information about prompting methods (Brown et al., 2020; Wei et al., 2022). In the context of prompting, prior studies have analyzed the mechanisms behind prompting methods, focusing either on the use of in-context examples (Min et al., 2022; Lampinen et al., 2022; Weber et al., 2023; Zhang et al., 2022a) or techniques that elicit reasoning, such as CoT prompting (Wei et al., 2022; Shaikh et al., 2023; Wang et al., 2023; Turpin et al., 2024). Conversely, we examine the model's behavior by conducting a literature analysis, which compiles data from scientific sources to reveal insights.

Literature Analysis Literature analysis systematically aggregates and examines data from multiple independent studies on a given topic to derive more precise and reliable conclusions. It has been widely applied in the biomedical domain for identifying target materials or clinical records (Bao et al., 2019; Yun et al., 2024). In the NLP domain, review analysis has been used for metric standardization (Reiter, 2018), literature review (Santu et al., 2024; Du et al., 2024), and assessing evaluation criteria for specific domains (Ostheimer et al., 2023). In contrast, our work employs a review analysis approach to evaluate the behavior of LLMs. In the context of LLMs, Asai et al. (2024) utilizes retrieval-augmented language models to synthesize scientific literature. However, this approach can only process a limited number of documents during retrieval to synthesize. The work by Sprague et al. (2024) is perhaps the most closely related to ours. They conducted a review analysis through a literature survey to examine the effectiveness of CoT prompting for LLMs. However, their study is focused on CoT prompting, conducted on a limited scale, and relies on manual extraction methods.

6 Conclusion

Our study streamlines literature analysis by using an LLM for dataset extraction, demonstrating that the automatically extracted dataset, LLMEVALDB, can yield novel findings and replicate manual analyses. We confirm the dataset's quality by replicating a key finding from Sprague et al. (2024). Our analysis provides insights into prompting configurations, such as the benefits of ICL, the combined effects of CoT and ICL, and the dataset characteristics on performance declines with CoT or ICL. Overall, our resources support ongoing literature analyses, enhancing the understanding of LLM behaviors.

7 Limitations

Target Model Scope Our study focuses on four leading proprietary LLMs selected for their widespread adoption and extensive documentation in existing literature. We excluded newer models like GPT4-o1 and Deepseek-R1 due to limited published results, though our analysis pipeline can easily accommodate them as more experimental data becomes available. While most analyzed models are not fine-tunable, this limitation presents an opportunity for future research combining fine-tuning and prompting methods analysis.

Further Validation Our literature analysis aims to identify trends by aggregating information across multiple studies, which generates potential hypotheses but requires systematic validation. While we report findings and patterns observed in the aggregated scientific literature, we did not independently validate each claim or finding. This limitation suggests the need for future work to rigorously test the hypotheses emerging from our analysis.

Attributes' Descriptiveness We frequently observed that the extracted attributes were not descriptive enough, which can hinder the dataset's utility for further analysis. Techniques like *Batch COT*, for example, would benefit from more detailed descriptions. Additionally, the dataset descriptions could be enhanced to better differentiate between various dataset characteristics. Our attempts to further generate the "collection process" of the dataset resulted in numerous inaccuracies. Moreover, efforts to automatically link descriptions to actual dataset instances also encountered technical challenges, necessitating extensive manual intervention. Future work should aim to develop more effective methods for comprehensive dataset characterization.

Dataset Canonicalization Cross-study analysis requires standardizing dataset names and formats. While we implemented strict rules for dataset canonicalization, our approach likely missed potential matches. Alternative matching techniques we explored using LLM produced too many false negatives, whereas linking to the PaperswithCode dataset ontology[4] was limited by its incomplete coverage of datasets.

[4] https://paperswithcode.com/datasets

Acknowledgments

We would like to thank Microsoft's Azure Accelerate Foundation Models Research Program and NVIDIA's Academic Grant Program for providing computational resources to support this work. This research is supported in part by the NSF under grant numbers IIS-2052498 and SMA-2418946. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. We also appreciate Ethan Mendes, Duong Minh Le, and Seongeun Park for their valuable discussions and feedback on the paper.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Rishabh Agarwal, Avi Singh, Lei M Zhang, Bernd Bohnet, Luis Rosias, Stephanie Chan, Biao Zhang, Ankesh Anand, Zaheer Abbas, Azade Nova, et al. 2024. Many-shot in-context learning. *arXiv preprint arXiv:2404.11018*.

Jacopo Amidei, Paul Piwek, and Alistair Willis. 2019. The use of rating and likert scales in natural language generation human evaluation tasks: A review and some recommendations. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 397–402.

Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku. *Semantic Scholar (CorpusID: 268232499)*.

Akari Asai, Jacqueline He, Rulin Shao, Weijia Shi, Amanpreet Singh, Joseph Chee Chang, Kyle Lo, Luca Soldaini, Sergey Feldman, Mike D'arcy, et al. 2024. Openscholar: Synthesizing scientific literature with retrieval-augmented lms. *arXiv preprint arXiv:2411.14199*.

Fan Bai, Junmo Kang, Gabriel Stanovsky, Dayne Freitag, Mark Dredze, and Alan Ritter. 2023. Schema-driven information extraction from heterogeneous tables. *arXiv preprint arXiv:2305.14336*.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Yujia Bao, Zhengyi Deng, Yan Wang, Heeyoon Kim, Victor Diego Armengol, Francisco Acevedo, Nofal Ouardaoui, Cathy Wang, Giovanni Parmigiani, Regina Barzilay, et al. 2019. Using machine learning and natural language processing to review and classify the medical literature on cancer susceptibility genes. *JCO Clinical Cancer Informatics*, 1:1–9.

Amanda Bertsch, Maor Ivgi, Uri Alon, Jonathan Berant, Matthew R Gormley, and Graham Neubig. 2024. In-context learning with long-context models: An in-depth exploration. *arXiv preprint arXiv:2405.00200*.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*. Introduces GPT-3 and demonstrates few-shot learning capabilities in language models.

Zhoujun Cheng, Jungo Kasai, and Tao Yu. 2023. Batch prompting: Efficient inference with large language model apis. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 792–810.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*. Introduces PaLM and demonstrates benefits of extreme scale in language models.

James Chua, Edward Rees, Hunar Batra, Samuel R Bowman, Julian Michael, Ethan Perez, and Miles Turpin. 2024. Bias-augmented consistency training reduces biased reasoning in chain-of-thought. *arXiv preprint arXiv:2403.05518*.

Jiangshu Du, Yibo Wang, Wenting Zhao, Zhongfen Deng, Shuaiqi Liu, Renze Lou, Henry Peng Zou, Pranav Narayanan Venkit, Nan Zhang, Mukund Srinath, et al. 2024. Llms assist nlp researchers: Critique paper (meta-) reviewing. *arXiv preprint arXiv:2406.16253*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Charles Frankston, Jonathan Godfrey, Shamsi Brinn, Alison Hofer, and Mark Nazzaro. 2024. Html papers on arxiv–why it is important, and how we made it happen. *arXiv preprint arXiv:2402.08954*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Yufang Hou, Charles Jochim, Martin Gleize, Francesca Bonin, and Debasis Ganguly. 2019. Identification of tasks, datasets, evaluation metrics, and numeric scores for scientific leaderboards construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5203–5213.

Dieuwke Hupkes, Mario Giulianelli, Verna Dankers, Mikel Artetxe, Yanai Elazar, Tiago Pimentel, Christos Christodoulopoulos, Karim Lasri, Naomi Saphra, Arabella Sinclair, et al. 2023. A taxonomy and review of generalization research in nlp. *Nature Machine Intelligence*, 5(10):1161–1174.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Salomon Kabongo, Jennifer D'Souza, and Sören Auer. 2024. Orkg-leaderboards: a systematic workflow for mining leaderboards as a knowledge graph. *International Journal on Digital Libraries*, 25(1):41–54.

Marcin Kardas, Piotr Czapla, Pontus Stenetorp, Sebastian Ruder, Sebastian Riedel, Ross Taylor, and Robert Stojnic. 2020. Axcell: Automatic extraction of results from machine learning papers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8580–8594.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, et al. 2023. Prometheus: Inducing fine-grained evaluation capability in language models. In *The Twelfth International Conference on Learning Representations*.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. 2024. T\" ulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*.

Andrew Lampinen, Ishita Dasgupta, Stephanie Chan, Kory Mathewson, Mh Tessler, Antonia Creswell, James McClelland, Jane Wang, and Felix Hill. 2022. Can language models learn from explanations in context? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 537–563.

Ryan Liu, Jiayi Geng, Addison J Wu, Ilia Sucholutsky, Tania Lombrozo, and Thomas L Griffiths. 2024. Mind your step (by step): Chain-of-thought can reduce performance on tasks where thinking makes humans worse. *arXiv preprint arXiv:2410.21333*.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023a. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023b. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522.

Chuanhao Lv, Kaiyan Chang, Ziyang Wang, Shuhan Zhou, Nuo Xu, et al. 2024. Abigsurvey. https://github.com/NiuTrans/ABigSurvey. GitHub repository.

Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. 2024. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, pages 1–11.

Bodhisattwa Prasad Majumder, Harshit Surana, Dhruv Agarwal, Bhavana Dalvi Mishra, Abhijeetsingh Meena, Aryan Prakhar, Tirth Vora, Tushar Khot, Ashish Sabharwal, and Peter Clark. 2024. Discoverybench: Towards data-driven discovery with large language models. *arXiv preprint arXiv:2407.01725*.

Mary L McHugh. 2012. Interrater reliability: the kappa statistic. *Biochemia medica*, 22(3):276–282.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064.

OpenAI. 2024. Openai o1-preview. https://openai.com/index/introducing-openai-o1-preview/. Model Webiste.

Phil Ostheimer, Mayank Nagda, Marius Kloft, and Sophie Fellenz. 2023. A call for standardization and validation of text style transfer evaluation. *arXiv preprint arXiv:2306.00539*.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are nlp models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094.

Ehud Reiter. 2018. A structured review of the validity of bleu. *Computational Linguistics*, 44(3):393–401.

Junhao Ruan, Long Meng, Weiqiao Shan, Tong Xiao, and Jingbo Zhu. 2024. Abigsurveyofllms. https://github.com/NiuTrans/ABigSurveyOfLLMs. GitHub repository.

Furkan Şahinuç, Thy Thy Tran, Yulia Grishina, Yufang Hou, Bei Chen, and Iryna Gurevych. 2024. Efficient performance tracking: Leveraging large language models for automated construction of scientific leaderboards. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7963–7977.

Shubhra Kanti Karmaker Santu, Sanjeev Kumar Sinha, Naman Bansal, Alex Knipper, Souvika Sarkar, John Salvador, Yash Mahajan, Sri Guttikonda, Mousumi Akter, Matthew Freestone, et al. 2024. Prompting llms to compose meta-review drafts from peer-review narratives of scholarly manuscripts. *arXiv preprint arXiv:2402.15589*.

Omar Shaikh, Hongxin Zhang, William Held, Michael Bernstein, and Diyi Yang. 2023. On second thought, let's not think step by step! bias and toxicity in zero-shot reasoning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4454–4470.

Mayank Singh, Rajdeep Sarkar, Atharva Vyas, Pawan Goyal, Animesh Mukherjee, and Soumen Chakrabarti. 2019. Automated early leaderboard generation from comparative tables. In *Advances in Information Retrieval: 41st European Conference on IR Research, ECIR 2019, Cologne, Germany, April 14–18, 2019, Proceedings, Part I 41*, pages 244–257. Springer.

Shruti Singh, Shoaib Alam, and Mayank Singh. 2024. Legobench: Leaderboard generation benchmark for scientific models. *arXiv preprint arXiv:2401.06233*.

Zayne Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. 2024. To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning. *arXiv preprint arXiv:2409.12183*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. 2024. Language models don't always say what they think: unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36.

Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. 2023. Towards understanding chain-of-thought prompting:

An empirical study of what matters. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Zirui Wang, Mengzhou Xia, Luxi He, Howard Chen, Yitao Liu, Richard Zhu, Kaiqu Liang, Xindi Wu, Haotian Liu, Sadhika Malladi, et al. 2024. Charxiv: Charting gaps in realistic chart understanding in multimodal llms. *Advances in Neural Information Processing Systems*, 37:113569–113697.

Lucas Weber, Elia Bruni, and Dieuwke Hupkes. 2023. Mind the instructions: a holistic evaluation of consistency and interactions in prompt-based learning. In *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*, pages 294–313.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, et al. 2023. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846*.

Yongjun Xu, Xin Liu, Xin Cao, Changping Huang, Enke Liu, Sen Qian, Xingchen Liu, Yanjun Wu, Fengliang Dong, Cheng-Wei Qiu, et al. 2021. Artificial intelligence: A powerful paradigm for scientific research. *The Innovation*, 2(4).

Sean Yang, Chris Tensmeyer, and Curtis Wigington. 2022. Telin: Table entity linker for extracting leaderboards from machine learning publications. In *Proceedings of the first Workshop on Information Extraction from Scientific Publications*, pages 20–25.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

Hye Sun Yun, Iain J Marshall, Thomas A Trikalinos, and Byron C Wallace. 2023. Appraising the potential uses and harms of llms for medical systematic reviews. *arXiv preprint arXiv:2305.11828*.

Hye Sun Yun, David Pogrebitskiy, Iain J Marshall, and Byron C Wallace. 2024. Automatically extracting numerical results from randomized controlled trials with large language models. *arXiv preprint arXiv:2405.01686*.

Hongxin Zhang, Yanzhe Zhang, Ruiyi Zhang, and Diyi Yang. 2022a. Robustness of demonstration-based learning under limited data scenario. *arXiv preprint arXiv:2210.10693*.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022b. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*.

A Details and Trials During Extraction

We provide the details and trials during the data extraction process.

Hyperparameters and Prompt Tuning We did not extensively tune any LLM hyperparameters, instead, we used OpenAI's default generation setup with greedy decoding. For prompting, we initially built upon previous work (Bai et al., 2023) as a foundation. Still, we adapted the prompts to better suit our project's objectives, such as (1) determining whether the table includes experimental results for our target model or (2) enhancing records using contextual information from the paper.

Preprocessing To extract the experimental results of target models and attributes, we utilize arXiv sources[5] published between January 2023 and December 2024, as this timeframe aligns with the release of proprietary models, ensuring relevance to the latest advancements in the field. We specifically targeted papers in the machine learning (ML) domain (cs.CV, cs.AI, cs.CL, cs.LG) and downloaded their LaTeX source files for detailed analysis. To extract structured data, such as tables, we utilized a regex-based rule extraction method to retrieve the LaTeX source of tables along with their indices (Bai et al., 2023).

Filtering Before the extraction stage, we filter tables to reduce extraction time and API costs. To cut computational expenses, we pre-filter tables for those containing results of target models. Using simple heuristics, we filter tables based on keywords related to the models (e.g., "gemini," "gpt"), which significantly reduces irrelevant data and minimizes LLM API usage. Moreover, we focus on leaderboard tables, which present the main results on specific benchmarks (Kardas et al., 2020), to assess LLM performance. We employ an LLAMA3.1-70B-Instruct (Dubey et al., 2024) to filter out non-leaderboard tables.

Extraction and Augmentation We employ schema-driven extraction to obtain records matching our target attributes from tables (detailed in §2.1). The LLM first determines if a table contains target model records and only proceeds with extraction when relevant models are present, skipping tables without target model data. While our approach resembles Bai et al. (2023), it achieves significant cost savings while maintaining accuracy (described in §3.1) by focusing solely on target model records rather than processing entire tables. This selective approach reduces API calls by up to 1/200 per table.

After extracting information from the table, we further augment the extracted records by incorporating additional context from the entire paper. This is important because valuable details, such as experimental specifics, are often found in various sections of the paper. We heuristically filter out irrelevant sections from the full latex source and provide the LLM with the extracted information to enhance the attribute details using the context. During this process, we also gathered BibTeX references for the datasets associated with each record, allowing us to link to and retrieve the original papers describing these datasets.

Dataset Description Generation Initially, we asked the LLM to generate descriptions using only the dataset name and subset information from its internal knowledge. This approach was cost-effective as it didn't require processing additional context. However, the LLM may be uncertain—particularly with lesser-known datasets or those beyond its knowledge cutoff. To address this, we instructed the model to refuse to answer when unsure about dataset information (Bai et al., 2022). In such cases, a second stage is triggered, where the model is prompted to extract the necessary information directly from the source papers. When LLM knowledge was insufficient, we directly extracted dataset descriptions from source papers. These source papers could be either the paper containing the table being processed or the original dataset paper cited by it. We used citation tags from §2.3 and rule-based heuristics to link to external arXiv papers associated with the original dataset. We then used the full content of these source papers to prompt the LLM to extract the dataset descriptions again, using the paper's contents as a reference.

We combined the dataset name and subset name to facilitate information generation or extraction. The schema for generation includes *Dataset Summary*, *Task Explanation*, and *Subset Description*. Initially, we attempted to generate a *Collection Process* section, detailing how the dataset was sourced and curated, but this led to excessive inaccuracies, so we decided to omit this part.

Models We utilize GPT-4o (Hurst et al., 2024) as the LLM for the pipeline and employ Llama-3.1 70B (Dubey et al., 2024) for filtering tables.

[5] https://info.arxiv.org/help/bulk_data_s3.html

Valid Metrics To ensure consistent analysis, we established a standardized set of metrics and excluded records using metrics outside this set. We also normalized all metric values to maintain uniformity across the analysis. The approved metrics are: Accuracy, Exact Match, F1, BLEU, Rouge, MRR (Mean Reciprocal Rank), Precision, Recall, Pearson Correlation Coefficient, MAE (Mean Absolute Error), and MSE (Mean Square Error). For all metrics except Pearson correlation coefficient, MAE, and MSE, we standardized the values to range from a minimum of 0 to a maximum of 100 (e.g., 0.63% was scaled to 63).

Missing Values If certain target attributes were unavailable from the paper or the LLM couldn't locate specific information, we instructed it to mark these missing values with "xx" as a placeholder.

Canonicalization Using rule-based heuristics, we canonicalized the *Dataset name* by grouping identical names or methods under different labels. We took care to avoid merging similar but distinct names representing different versions, variations, or separate entities by applying strict character and abbreviation matching rules. We initially explored linking datasets to the PaperswithCode ontology[6]. However, the outdated and incomplete nature of its dataset list made this approach infeasible. We then experimented with clustering algorithms and large language models (LLMs) for grouping, but these methods resulted in an excessive number of false positives. Ultimately, we adopted a rule-based grouping algorithm as our approach.

Postprocessing To ensure dataset quality, we removed duplicate records with identical values for *Dataset Name*, *Subset*, *Number of FewShots*, *Prompting Method*, *Metric Name* but differing *Performance*. Additionally, we filtered out records with invalid dataset descriptions.

B Details in Human Evaluation

We detail the human evaluation process.

Extraction Evaluation The objective is to assess whether the fields within records have been accurately extracted. This evaluation involves checking each field in a record against the original source paper to confirm its correctness. Fields are marked with 'o' for success or 'x' for failure, depending on the accuracy of the extraction. Annotators were instructed to use the *original_extracted_dictionary* for verification, as metric names have been standardized and canonicalized. A table index was provided to help locate records and tables more quickly for annotation. Additionally, it is permissible for prompting methods to include few-shot examples if relevant information is not found in the paper.

Description Evaluation This step evaluates whether the description aligns appropriately with the dataset-subset pair. The evaluation protocol (5-Point Likert Scale) involves assessing the quality of the dataset description based on the following rubric:

- Score 1: The description is completely unrelated to the dataset-subset pair.
- Score 2: The description has minimal relevance but lacks alignment or context.
- Score 3: The description is moderately relevant, capturing the essence of the dataset-subset pair but includes noticeable inaccuracies.
- Score 4: The description is highly relevant with only minor inaccuracies.
- Score 5: The description is fully relevant and entirely accurate, with no errors.

When scoring, annotators used references from the web or literature searches to ensure the evaluation was well-informed. Our scoring rubric is designed to approximate an interval scale, allowing us to compute average scores. This approach aligns with standard practices in the machine learning field for evaluating response quality, whether through model-based or human-based assessments (Liu et al., 2023b; Kim et al., 2023).

C Details in Literature Analysis of Prompting Behavior in Frontier LLMs

C.1 Which Categories Benefit from CoT?

To replicate the Chain-of-Thought (CoT) improvements over standard direct prompting, as demonstrated by Sprague et al. (2024), we adopt their category schemas, which include entailment, text classification, generation, knowledge, contextQA, multihopQA, commonsense reasoning, logical reasoning, spatial reasoning, and math. While these

[6] https://paperswithcode.com/datasets

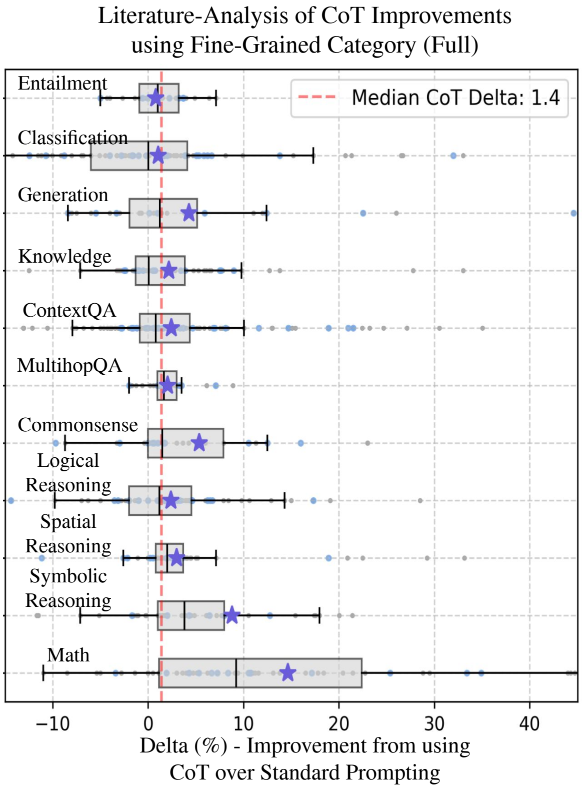


Figure 8: Full version of box plots showing performance improvements with CoT compared to standard prompting, categorized according to Sprague et al. (2024) for a fine-grained investigation in various reasoning tasks.

categories provide a fine-grained definition of traditional NLP tasks, they lack generalizability in encompassing the broader capabilities of modern LLMs, such as multimodality, safety, and tool use. The complete version of Fig. 6 is presented below in Fig. 8.

Extracting records for this type of analysis is challenging due to the strict criteria for sample selection. While Sprague et al. (2024)'s dataset contains a larger overall sample size, as it is not limited to a single model and LLMEVALDB is not specifically oriented to replicating their study. For the overlapping model (GPT-4), their dataset includes 168 instances, whereas ours contains 553. This demonstrates the scalability of our pipeline.

C.2 Which Categories Benefit from ICL?

We visualize the distribution of in-context examples, comparing cases with more versus fewer demonstrations, in Fig. 9. The results show that the overall median and distribution are similar to those in Fig. 7, suggesting that the presence of demonstrations is more crucial than their quantity.

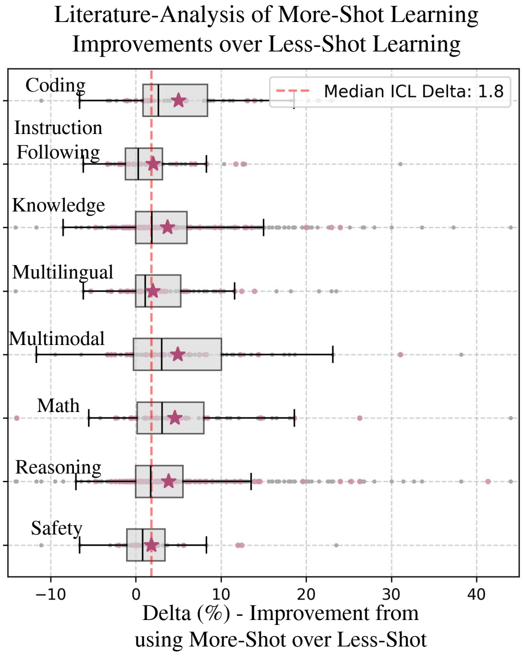


Figure 9: Box plot distributions showcasing performance enhancements using more in-context examples compared to less in-context examples setup, categorized by skill sets in §3.2. Grey dots represent individual deltas, while sky pink dots show the mean delta aggregated for each paper. A pink star indicates the mean delta for each category.

C.3 How Do Trends in Peer-Reviewed Papers Compare to Those in arXiv Publications?

We reanalyzed the joint behavior findings from §4.4 using only peer-reviewed papers published in journals or conferences. Table 8 and Table 9 present these results. The filtered data confirms our original finding regarding the interaction between CoT and ICL: CoT with demonstrations (ICL) consistently outperforms CoT without demonstrations. Additionally, CoT's improvement over standard prompting remains consistent regardless of demonstration count (zero-shot or few-shot).

D Dataset Examples

We provide a curated selection of examples from LLMEVALDB in Table 10 and Table 11.

E Frequent Datasets by Skill Category

We present a list of the top 10 most frequently used datasets in Table 12, aggregated according to our defined skill categories, in §3.2. Frequency is measured by the number of unique papers that evaluate a dataset, without counting multiple experiments within the same paper. Some datasets appear in

Performance comparison	Original			Filtered		
	Median	Q1	Q3	Median	Q1	Q3
Few-shot CoT − Zero-shot CoT	3.0	0.4	9.2	5.0	2.3	10.4

Table 8: Effect of in-context demonstrations *within* Chain-of-Thought prompting. **Original** uses all papers; **Filtered** keeps only top-venue papers.

Demonstration level	Original Δ			Filtered Δ		
	Median	Q1	Q3	Median	Q1	Q3
Zero-shot	+1.3	−0.4	+4.7	+1.3	0.0	+3.6
Few-shot	+0.9	−1.2	+3.7	+1.3	0.0	+3.0

Table 9: CoT versus standard prompting when the demonstration setting is fixed. Positive Δ means CoT outperforms Direct prompting.**Original** Δ uses all papers; **Filtered** Δ keeps only top-venue papers.

multiple categories due to multi-label categorization and the use of different subsets focusing on distinct skill areas.

F Prompts Used during Data Extraction

We provide the prompts used during the data extraction in Table 13 and Table 14.

ArXiv ID	Table	Dataset	Subset	Dataset Description	Model	Shots	Prompt	Metric	Value
2301.08721	2	CommonsenseQA	xx	**Dataset Summary**: This dataset is a multiple-choice question answering dataset focused on commonsense knowledge. It is designed to evaluate a model's ability to understand and apply commonsense reasoning to answer questions correctly. The dataset is widely used in the natural language processing domain to benchmark the performance of models on tasks requiring commonsense understanding. **Task Explanation**: The task involves providing a question along with multiple answer choices, where the input is the question and the list of possible answers. The output is the correct answer choice. The task is evaluated based on the accuracy of the model's predictions compared to the correct answers provided in the dataset.	GPT-4	12	Batch CoT	Acc	86
2301.08745	8	Flores-101	De → En	**Dataset Summary**: This dataset is a multilingual translation benchmark designed to evaluate machine translation systems across a wide range of languages. It covers 101 languages, providing a comprehensive resource for assessing translation quality and performance. The dataset is intended for tasks involving language translation, with a focus on both high-resource and low-resource languages. Key characteristics include its diverse language pairings and standardized evaluation metrics, which aim to facilitate consistent and fair comparisons of translation models. **Task Explanation**: The primary task associated with this dataset is machine translation. The input consists of text in one language, and the output is the translated text in another language. For this specific subset, the task involves translating text from German (De) to English (En). Evaluation of the task is typically conducted using metrics such as BLEU, which measures the accuracy and fluency of the translated output compared to reference translations. **Subset Description**: The De→En subset specifically focuses on translations from German to English. This subset is used to evaluate the performance of translation models in converting German text into English, providing insights into the model's ability to handle this particular language pair.	GPT-4	0	Direct	BLEU	46
2303.07992	3	KQApro	xx	**Dataset Summary**: This dataset is a large-scale knowledge-based question answering dataset designed to evaluate the ability of models to understand and reason over complex questions. It is primarily used in the domain of natural language processing and artificial intelligence, focusing on tasks that require comprehension of structured knowledge bases. The dataset features a diverse set of questions that test various reasoning skills, such as logical reasoning, comparison, and temporal reasoning. The evaluation goals are to assess the accuracy and reasoning capabilities of models in answering these questions correctly. **Task Explanation** The primary task associated with this dataset is knowledge-based question answering. The input consists of a natural language question, and the output is the correct answer derived from a structured knowledge base. Models are expected to interpret the question, retrieve relevant information from the knowledge base, and perform the necessary reasoning to arrive at the correct answer. The task evaluation method typically involves measuring the accuracy of the model's answers compared to a set of ground truth answers.	GPT-4	0	xx	Accuracy	57.2
2304.08244	3	API-Bank	Call	**Dataset Summary**: This dataset is a comprehensive benchmark designed to evaluate and enhance the capabilities of Large Language Models (LLMs) in utilizing external API tools. It focuses on tool-augmented LLMs, assessing their abilities in planning, retrieving, and calling APIs. The dataset includes a wide range of domains and APIs, aiming to provide a realistic and diverse evaluation environment. The primary goal is to understand the effectiveness of current LLMs in tool usage, improve their capabilities, and identify challenges in leveraging tools. **Task Explanation**: The task involves evaluating LLMs on their ability to interact with APIs based on user queries. The input consists of user queries and API documentation, while the output is the correct API call and response. The task is evaluated based on the correctness of API calls and the quality of responses, using metrics like accuracy and ROUGE-L. **Subset Description**: The ""Call"" subset specifically focuses on evaluating the ability of LLMs to make API calls based on given queries when the APIs are known. This subset tests the basic capability of LLMs to interact with APIs directly without the need for retrieval or planning.	GPT-4	0	Zero-shot	Rouge	36.91
2310.05915	1	HotpotQA	xx	**Dataset Summary**: This dataset is a large-scale, high-quality question answering dataset designed to facilitate research in the domain of natural language processing, specifically in multi-hop question answering tasks. It contains questions that require reasoning over multiple documents to arrive at the correct answer. The dataset is intended to evaluate the ability of models to perform complex reasoning and synthesis of information across different contexts. **Task Explanation**: The primary task involves providing a natural language question as input and expecting a concise answer as output. The questions are designed to require multi-hop reasoning, meaning that the answer cannot be found in a single document but requires synthesizing information from multiple sources. The task is evaluated based on the accuracy of the answers provided by the model, often using metrics such as exact match and F1 score to assess performance.	GPT-4	xx	IO	Exact Match	37.2

Table 10: Curated list of examples from LLMEvalDB (Target Model: GPT-4 (Achiam et al., 2023)).

ArXiv ID	Table	Dataset	Subset	Dataset Description	Model	Shots	Prompt	Metric	Value
2407.19619	1	HPC	Fortran2C++	**Dataset Summary**: This dataset is designed for the domain of high-performance computing (HPC) code translation, specifically focusing on translating between OpenMP Fortran and C++ code. It aims to train and evaluate large language models (LLMs) to enhance their translation capabilities between these two languages. The dataset is characterized by its diverse and representative collection of open-source OpenMP benchmarks, refined through code similarity tests. The evaluation goals include improving translation accuracy, as measured by CodeBLEU scores, and achieving human-level translation proficiency. **Task Explanation**: The primary task is to translate code from OpenMP Fortran to C++. The input is a Fortran code snippet, and the output is its equivalent C++ translation. The task is evaluated using CodeBLEU, a metric that assesses the quality of code translation by considering both syntactic and semantic elements. Human evaluation is also employed to ensure the translations are correct, readable, and maintain the original code's functionality. **Subset Description**: The Fortran to C++ translation subset focuses specifically on translating code from Fortran to C++. It serves the purpose of training models to understand and convert Fortran code into its C++ equivalent, addressing the challenges of translating between these two prominent HPC languages.	GPT-4o	0	Zero-Shot	BLEU	37.1
2408.02718	3	MMIU	Overall	**Dataset Summary**: This dataset is designed for the domain of multimodal understanding, specifically focusing on tasks that require the integration of information from multiple images. It aims to facilitate research in areas such as image comparison, visual storytelling, and cross-image reasoning. The key characteristics of the dataset include a diverse set of image pairs or groups, each accompanied by annotations or questions that require understanding the relationships or narratives across the images. The evaluation goals are to assess the ability of models to comprehend and reason about multiple images simultaneously. **Task Explanation**: The primary task involves taking multiple images as input and producing an output that demonstrates understanding of the relationships or narratives between them. This could involve answering questions, generating descriptive text, or identifying similarities and differences. The task evaluation method typically involves comparing the model's output to human-generated annotations or answers, using metrics such as accuracy, BLEU score, or human judgment for qualitative assessments. **Subset Description**: The Overall subset encompasses the entire dataset, providing a comprehensive collection of all image pairs or groups and their associated annotations. This subset is intended for general evaluation and benchmarking of models on the full range of tasks supported by the dataset.	GPT-4o	xx	xx	Accuracy	55.7
2405.02861	3	LexBench	IED	**Dataset Summary**: This dataset is part of a comprehensive evaluation suite designed to test language models on various semantic phrase processing tasks. It focuses on idiomatic expression detection, which is one of the ten tasks included in the suite. The dataset aims to evaluate the ability of language models to understand and process idiomatic expressions, which are non-compositional phrases whose meanings cannot be deduced from their individual components. The evaluation goals include assessing model performance in classification, extraction, and interpretation tasks related to idiomatic expressions. **Task Explanation**: The task involves detecting idiomatic expressions within a given context. The input consists of a sentence containing an idiomatic expression, and the output is a choice from multiple options that best describes the idiomatic expression's meaning. The task is evaluated using exact match accuracy, where the model's prediction is compared against the correct option. **Subset Description**: The Idiomatic Expression Detection (IED) subset specifically focuses on identifying idiomatic expressions within sentences and selecting the correct interpretation from multiple options. This subset is designed to challenge language models in understanding the non-compositional nature of idiomatic expressions.	Claude3	0	zero-shot	Accuracy	66.3
2409.19667	5	ProGraph	BGT	**Dataset Summary**: This dataset is a benchmark designed to evaluate the capability of large language models (LLMs) in graph analysis tasks. It focuses on enabling LLMs to process and analyze graphs using external APIs, similar to how human experts would approach such tasks. The dataset is crafted to assess the models' ability to generate code solutions for graph-related problems, rather than relying solely on reasoning over raw inputs. The benchmark includes 512 problems across three categories: basic graph theory, graph statistical learning, and graph embedding. The evaluation aims to measure the models' performance in leveraging programming libraries to solve graph tasks. **Task Explanation**: The primary task involves generating Python code to solve graph-related problems using specified APIs. The input consists of a problem statement describing a graph task, and the output is the corresponding Python code that utilizes relevant APIs to solve the task. The evaluation method involves assessing the pass rate (the ratio of executable code) and accuracy (the ratio of correct answers from the executable code) of the generated solutions. **Subset Description**: The Basic Graph Theory (BGT) subset focuses on fundamental concepts of graph theory, including types, properties, classical algorithms, and basic operations. Tasks in this subset may involve checking graph acyclicity, computing node centrality, or finding maximum cardinality matching.	Gemini1.0	xx	xx	Accuracy	27.7

Table 11: Curated list of examples from LLMEVALDB (Target Models: GPT-4o (Hurst et al., 2024), Claude3 Opus (Anthropic, 2024), and Gemini1.0 Pro (Team et al., 2023)).

Skill Category	**Dataset**
Reasoning	MMLU, CommonsenseQA, StrategyQA, HotPotQA, ARC Challenge, BIG-Bench Hard, Spider, DROP, Winogrande, ScienceQA
Knowledge	MMLU, MedQA, MedMCQA, PubmedQA, NQ, TruthfulQA, United States USMLE, TriviaQA, ScienceQA, Financial Phrase Bank
Multilinguality	C-Eval, WMT22, FLORES-200, XQuAD, Flores-101, MKQA, Flores, MMMLU, SIB-200, TOEFL11
Multimodality	ScienceQA, EgoSchema, MMMU, DAIC-WOZ, MathVerse, MathVista, MMBench, ChartQA, Text-VQA, DocVQA
Math	GSM8K, MATH, SVAMP, AQuA, AddSub, Multi-Arith, ASDIV, Multilingual Grade School Math, Single Eq, MathVista
Coding	Spider, BIRD, Human Eval, MBPP, APPS, CodeReviewer, Exception Handling Code Generation, BigCloneBench, WikiSQL, SecurityEval
Instruction	Mind2Web, SCAN, PromptCBLUE, InstructExcel, Len, RuLES, SFRES, SemEval, InstructDial++, Alpaca Eval
Safety	AdvBench, ETHICS, Bias Benchmark for Question Answering (BBQ), HHH Alignment, Simple-SafetyTests, i2b2/UTHealth de-identification challenge dataset, Fifty Shades of Bias, LLM-generated emails, Tensor Trust, MULTITuDE benchmark
Tool Use	API-Bank, NERetrieve, PubMed Retrieval and Synthesis (PubMedRS-200), student help requests, PowerPoint Task Completion (PPTC), ToolTalk, TaskBench, AgentBench-WB, Compliance Checking, CardBench

Table 12: Top 10 Most Frequent Datasets per Skill Category. Frequency is measured by the number of unique papers evaluating a dataset, without counting multiple experiments within the same paper.

Component	**Prompt**
Identifying Leaderboard Tables	Determine if the given Table LaTeX represents a leaderboard table. Leaderboard tables showcase the main results of the paper on a specific benchmark, often comparing these results with those from other studies. Tables are NOT considered leaderboard tables if they focus on ablation studies, hyperparameter tuning, dataset statistics, or other supplementary experiments. Respond with ’true’ if the table is a leaderboard table, and ’false’ otherwise, with no additional explanation. **Input:** Table LaTeX: **Output:** Classification Output: true/false
Schema-Driven Extraction	Your task is to extract all numeric cells representing the experimental results of a specified target model from a table LaTeX source, following the provided template. When extracting results for the target model, exclude results from its variant models. For example: - For GPT-4, exclude GPT-4o, GPT-4-v, or Deplot + GPT-4. - For Claude3 Opus, exclude Claude3 Sonnet, Claude3 Haiku, Claude2, Claude 3.5. - For Gemini 1.0 Pro, exclude Gemini 1.5, Gemini 1.5 Pro, Gemini Ultra, and Gemini Flash. - For GPT-4o, exclude GPT-4, GPT-4-o1, GPT4-Turbo, and GPT4-V. However, include results from different versions of the target model across time periods (e.g., GPT-4, GPT-4-0828, GPT-4-0623, GPT-4-0314). If the table contains results where the target model is used for generation or evaluation, exclude those results. The goal is to extract results about the target model itself, not those where it is used as a tool. If no numeric cells related to the target model are found, output "<FAILED>". During output, return only the extracted results in the following template. Do not provide explanations. For any unanswerable attributes, leave their value as "xx". Template: {"value": "xx", "dataset": "xx", "dataset_citation_tag": "xx", "subset": "xx", "model_name": "xx", "metric": "xx", "prompting_method": "xx", "number_of_shots": "xx"} Field Descriptions: - value: Extracted numeric cell value. - dataset: Name of the dataset or benchmark (must be a proper noun, e.g., "synthetic dataset" is not acceptable). - dataset_citation_tag: Citation tag for the dataset. - subset: Dataset subset used (e.g., subtask, domain, split). - model_name: Name of the model used in the experiment. - metric: Evaluation metric used. - prompting_method: Prompting method used (do not include shot count here). - number_of_shots: Integer value representing the number of shots used. **Input:** Target Model: Table LaTeX Source: **Output** Extracted Results:

Table 13: Prompts for identifying leaderboard tables and schema-driven data extraction, where the model needs to identify if the table contains the experimental results of target models.

Component	**Prompt**
Context Augmentation	Augment the extracted records from the table’s LaTeX source by incorporating additional context from the text source to enrich and complete the records. Extracted Record Template: {”value”: ”xx”, ”dataset”: ”xx”, ”dataset_citation_tag”: ”xx”, ”subset”: ”xx”, ”model_name”: ”xx”, ”metric”: ”xx”, ”prompting_method”: ”xx”, ”number_of_shots”: ”xx”} To accurately augment and enrich the extracted records, follow these steps systematically: 1. value: Referencing the table source, if a numeric value is only partially extracted from a table cell, ensure that the entire content of the cell is used to update the value. 2. dataset: Referencing the table source, table caption, and source text, locate the full name of the dataset and update the name accordingly. 3. dataset_citation_tag: Referencing the table source, table caption, and source text, identify the dataset citation tag to the extracted record. Avoid using LaTeX syntax (e.g., cite and curly brackets); return only the tag name contained within. 4. subset: Referencing the table source, table caption, and source text, identify specific subsets of the dataset, such as subtasks, domains, splits, or language pairs, and provide detailed descriptions of each subset. Prioritize the use of column information from the table source to identify the subset. If the subset is not explicitly mentioned in the table source, refer to the table caption or source text to identify the subset. 5. model_name: Referencing the table source, if a model name is partially extracted, revisit the corresponding table cell to ensure the entire content is included. 6. metric: Referencing the table source, table caption, and source text, extract the metrics used in the experiment along with detailed descriptions and any additional information about the evaluation protocol. 7. prompting_method: Referencing the table source, table caption, and source text, search for and identify the prompting technique (e.g., direct, CoT, etc.) applied in the experiment and provide a detailed explanation of it. Do not include any information related to the number of shots (e.g., few-shot, zero-shot, three-shot) in this field. 8. number_of_shots: Referencing the table source, table caption, and source text, specify the number of shots used in the experiment. This must be an integer value. During output, output only the template following extracted results. Do not output any explanations or use LaTeX grammar. For any unanswerable attributes in the templates, leave their value as ”xx”. **Input** Extracted Records: Table LaTeX Source: Text Source: **Output** Augmented Extracted Records:

Table 14: Prompt for context augmentation.