Task Contamination: Language Models May Not Be Few-Shot Anymore

Changmao Li, Jeffrey Flanigan

University of California, Santa Cruz
changmao.li@ucsc.edu, jmflanig@ucsc.edu

Abstract

Large language models (LLMs) offer impressive performance in various zero-shot and few-shot tasks. However, their success in zero-shot and few-shot settings may be affected by task contamination, a potential limitation that has not been thoroughly examined. This paper investigates how zero-shot and few-shot performance of LLMs has changed chronologically over time. Utilizing GPT-3 series models and several other recent open-sourced LLMs, and controlling for dataset difficulty, we find that on datasets released before the LLM training data creation date, LLMs perform surprisingly better than on datasets released after. This strongly indicates that, for many LLMs, there exists task contamination on zero-shot and few-shot evaluation for datasets released prior to the LLMs' training data creation date. Additionally, we utilize training data inspection, task example extraction, and a membership inference attack, which reveal further evidence of task contamination. Importantly, we find that for classification tasks with no possibility of task contamination, LLMs rarely demonstrate statistically significant improvements over simple majority baselines, in both zero and few-shot settings.

1 Introduction

Recently there has been much interest in few-shot methods, in particular in-context learning (ICL, Brown et al. 2020) with large language models. In-context learning has the benefit of yielding excellent performance while requiring very little data, sometimes relying on only a few examples for the task. These promising results have led to an explosion of work on in-context learning methods across a wide variety of tasks (Schick and Schütze 2021a,b; Poesia et al. 2022; Hu et al. 2022b), including prompt tuning methods (Qin and Eisner 2021; Lester, Al-Rfou, and Constant 2021), chain-of-thought methods (Wei et al. 2022; Wang, Deng, and Sun 2022; Wang et al. 2023; Aiyappa et al. 2023), tool-based methods (Schick et al. 2023; Yang et al. 2023).

However, along with this explosion of work in ICL, many have raised concerns about data contamination (Brown et al. 2020; Jacovi et al. 2023), that is, prior knowledge of data or a task which is thought to be unseen by the model. Data contamination can happen in multiple ways. One common contaminant is **test data contamination**, the inclusion of test data examples and labels in the pre-training data. Another contaminant for zero or few-shot methods, which we call **task contamination**, is the inclusion of task training examples in the pre-training data, effectively making the evaluation no longer zero or few-shot.[1]

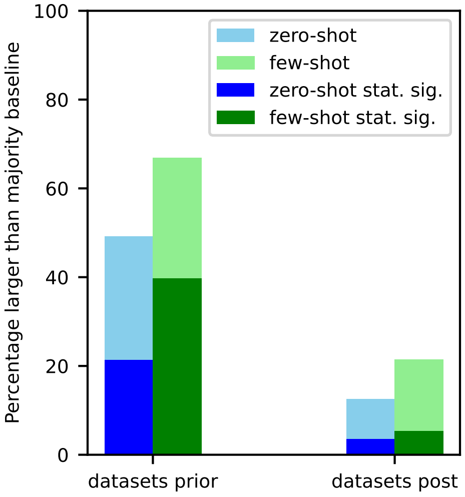


Figure 1: Percentage of datasets with accuracy higher than the majority baseline for datasets released prior and post LLM training data collection date, for both zero-shot (blue, left) and few-shot (green, right). Results are across all models and all datasets. On datasets released post training data collection date for the LLM, the LLM is much less likely to improve upon the simple majority baseline. *Stat. sig.* (darker) is the percent of datasets for which the performance above majority baseline is significant at the 99% confidence level.

Simply evaluating the scope of this contamination is difficult to do (Magar and Schwartz 2022; Jacovi et al. 2023). Closed models do not release their pre-training data. While

[1] Zero-shot evaluation is evaluation where a model has seen zero examples for the task. Few-shot, or N-shot, where N is a small number, is where the model has seen N examples for the task. Prior work has sometimes defined zero-shot for multi-class classification as predicting *classes* that have never been seen during training, but most recent work does not use this definition.

open models give the sources, crawling the sites to obtain that data is non-trivial, especially if the data has changed from when it was crawled. For models that are pre-trained on freely available pre-training corpora, simply grepping for examples in the pre-training corpora may not be reliable due to differences in data formatting (such as XML vs CVS, etc) or differences in text normalization and tokenization.

In this paper we empirically measure the scope of task contamination for few-shot methods across various models and tasks. To the best of our knowledge, we are the first to systematically analyze this problem. We evaluate 12 different models, ranging from closed GPT-3 series models (OpenAI 2023b) to open models including Fairseq MoE (Artetxe et al. 2022), GPT-J (Wang and Komatsuzaki 2021), Bloom (Scao et al. 2022), OPT (Zhang et al. 2022) , LLaMA (Touvron et al. 2023), Alpaca (Taori et al. 2023), and Vicuna (Chiang et al. 2023) on 16 classification tasks and 1 semantic parsing task.

We analyze each model on datasets created before its training data was crawled on the internet versus datasets created afterward. We find that datasets created before the LLM training data was collected have a significantly higher chance of having performance higher than the majority baseline (Fig. 1).

We perform training data inspection and task example extraction to look for possible task contamination. Importantly, we find that for classification tasks with no possibility of task contamination, models rarely demonstrate statistically significant improvements over simple majority baselines across a range of tasks, in both zero and few-shot settings (Fig. 2).

As a case study, we also attempt to conduct a membership inference attack for a semantic parsing task (Spider, Yu et al. 2019) for all models in our analysis. We find a strong correlation (R=.88) between the number of extracted examples and the accuracy of the model on the final task (Fig. 6). This is strong evidence that the performance increase in zero-shot performance on this task is due to task contamination.

Additionally, we look closely at the GPT-3 series models. We find that training examples can be extracted from the GPT-3 models, and that the number of extractable training examples increased from each version from `davinci` to `GPT-3.5-turbo`, and closely tracks the increase in zero-shot performance of the GPT-3 models on that task (Fig. 2). This is strong evidence that the increase in performance on these tasks across GPT-3 models from `davinci` to `GPT-3.5-turbo` is due to task contamination.

2 Overview

We employ four methods of measuring task contamination.

1. **Training data inspection**: Search through the training data to find task training examples.
2. **Task example extraction**: Extract task examples from an existing model. Extraction is only possible with instruction-tuned models. This analysis can also be done for training data or testing data extraction (Sainz et al. 2023b). Note: For the purposes of detecting task contamination, the extracted task examples need not exactly match existing training data examples. Any examples demonstrating the task indicate possible contamination for zero and few-shot learning.
3. **Membership inference**: This method only applies to generation tasks. Check if the model generated content for an input instance is exactly the same as the original dataset (Hu et al. 2022a). If there is an exact match, we can infer it is a member of the LLM's training data. This differs from task example extraction because generated output is checked for an exact match. Exact matches for an open-ended generation task strongly indicate the model has seen those examples during training. The model is not just good, it is psychic: it has knowledge of the exact phrasing used in the data. Note: this can only be used for generation tasks.[2]
4. **Chronological analysis**: For a set of models whose training data has been collected at a range of known times, measure performance on a dataset with a known release date, and check for evidence of contamination using chronological evidence.

The first three methods have high precision, but suffer from low recall. If data is found in the training data for the task, then it is certain that it has seen examples. But because of data formatting variations, variations in keywords used to define the task, and the size of the dataset, the absence of evidence for contamination using the first three methods is not evidence of absence.

The fourth method, chronological analysis, is high recall, but low precision. If the performance is high due to task contamination, then a chronological analysis will have a high chance of catching it. But other factors could also contribute to increased performance over time, so the precision is low.

Due to their inherent trade-offs, we employ all four methods for detecting task contamination. With all four methods, we find strong evidence of task contamination for some combinations of models and datasets. We begin with a chronological analysis for all models and datasets we tested, since it has the highest potential for catching possible contamination (§4). We then look for further evidence of task contamination using training data inspection (§5) and task example extraction (§6). Next we look at the performance of LLMs on tasks without contamination (§7), and conclude with additional analysis using a membership inference attack (§8).

3 Models and Datasets

Models We experimented with 12 models. Table 1 lists these models, along with the collection dates of the training data and release dates for each model.[3] The 12 models we use can be further categorized into two broad groups: (1) five proprietary GPT-3 series models ("closed") and (2) seven open models with free access to their weights ("open"). Comparing models from these two groups yields valuable insights into the difference between proprietary, high-performance

[2] Exact matches for the input do not indicate task contamination because the input text could have been seen, but it needs to be paired with the output label for task contamination.

[3] GPT-3 series training data collection dates are obtained from https://platform.openai.com/docs/models/overview

Model	Training data	Release
davinci	Up to Oct 2019	May 2020
davinci-001	Up to Oct 2019	Jun 2020
davinci-002	Up to Jun 2021	Jan 2022
davinci-003	Up to Jun 2021	Nov 2022
GPT-3.5-T	Up to Sep 2021	Mar 2023

(a) GPT-3 Series LLMs

Model	Training data	Release
Fairseq MoE	Up to Feb 2019	Dec 2021
GPT-J	Up to 2020	Jun 2021
OPT	Up to Oct 2021	May 2022
BLOOM	Prior Aug 2022	Nov 2022
LLaMA	Up to Aug 2022	Feb 2023
Alpaca	From davinci-003	Mar 2023
Vicuna	From ChatGPT	Mar 2023

(b) Open LLMs

Table 1: Dates for the training data creation and model release. davinci-XXX refers to `text-davinci-XXX`. GPT-3.5-T refers to `GPT-3.5-turbo-0301`.

models like those from the GPT-3 series and more accessible, community-driven open models. More information about hyperparameters for these models is given in the Appendix A.

Datasets Zero-shot and few-shot evaluations involve models making predictions on tasks that they have never seen or seen only a few times during training. The key premise is that the models have no prior exposure to the particular task at hand, ensuring a fair evaluation of their learning capacity. Contaminated models, however, give a false impression of its zero- or few-shot competency, as they have already been trained on task examples during pretraining. Detecting such inconsistencies would be relatively easier in a chronologically ordered dataset, where any overlap or anomaly would stand out. Based on this narrative, we split the datasets into two categories: datasets released before or after January 1st, 2021, identified as **pre-2021** datasets and **post-2021** datasets. We use this division to analyze the zero-shot or few-shot performance difference between older datasets and newer ones, with the same division applied for all LLMs. We also use the per-LLM division **pre-collection** and **post-collection** datasets, which distinguishes datasets that the model was possibly trained on (pre-collection datasets) from the datasets it could not have been trained on (post-collection datasets). Table 1 presents the creation time of the training data for each model. Information about the datasets can be found in the Appendix B, while release dates for each dataset are listed in Table 2.

4 Chronological Analysis

We start with a chronological analysis. This allows us to detect patterns of possible task contamination across the LLMs and datasets we examine.

Pre-2021		Post-2021	
Dataset	Year	Dataset	Year
RTE	2009	StrategyQA	2021
WNLI	2011	NewsMTSC-MT	2021
COPA	2011	NewsMTSC-RW	2021
SST-2	2013	NLI4Wills	2022
MRPC	2015	CREPE	2023
QNLI	2018	FOMC	2023
CB	2019	NewsMet	2023
WiC	2019		
BoolQ	2019		

Table 2: Dataset release year for each dataset, split into pre-2021 datasets and post-2021 datasets.

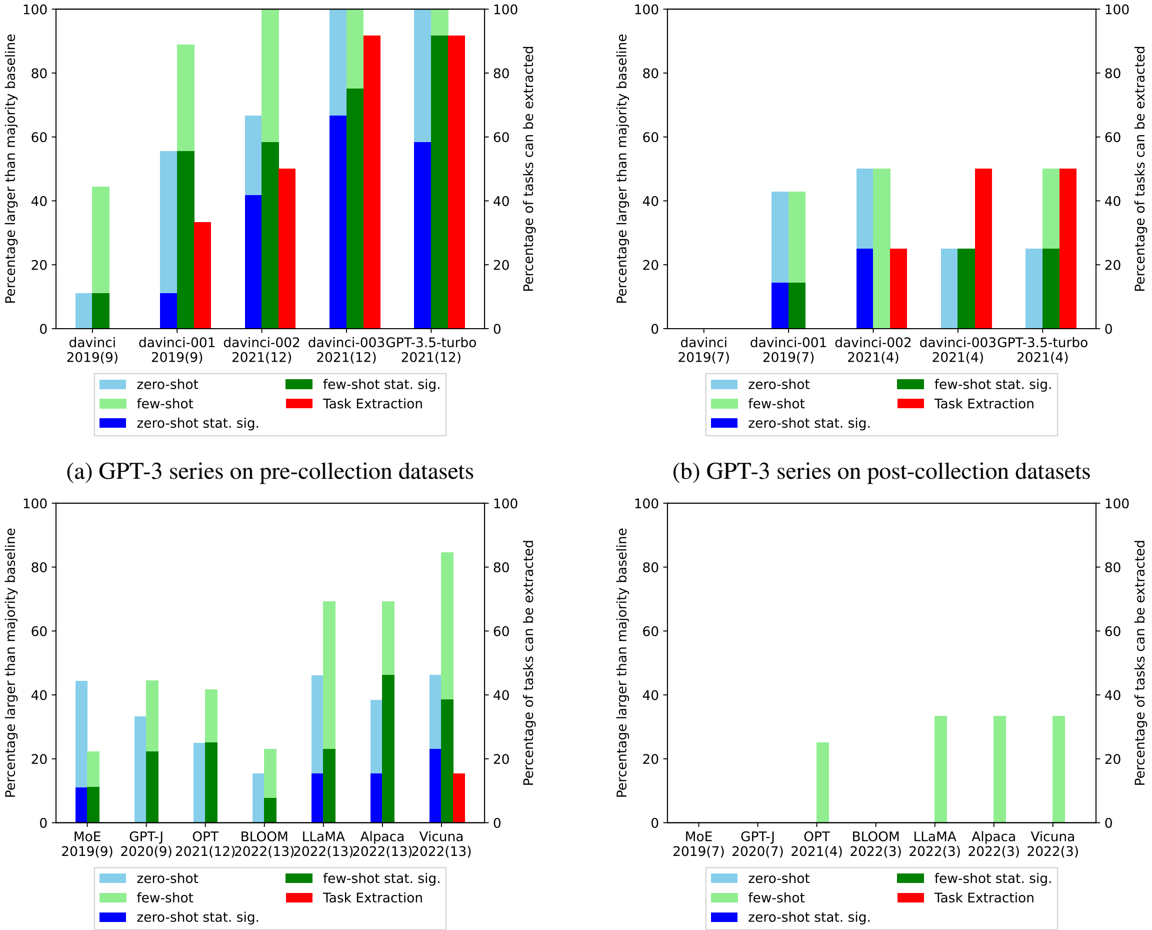
Analysis of Pre- and Post-collection Datasets

We perform a global chronological analysis across all datasets and LLMs. We look at the difference between performance on datasets released before the training data collection date for the LLM (**pre-collection**) versus after the training data collection date (**post-collection**). Specifically, we focus on whether the model is above the majority baseline.[4] In this section we use this measure, instead of averaging the performance across datasets, to avoid datasets with large performance differences dominating the analysis.

With 12 models and 16 datasets, we have 192 model/dataset combinations. Of these combinations, 136 the datasets were released before the LLM training data collection date (pre-collection) and 56 the dataset were release after (post-collection). For both sets, we compute the percentage of model/dataset combinations for which the model beats the majority baseline, both zero-shot and few-shot. The results are shown in Fig. 1. We find that for datasets released prior to the creation of the LLM, it is more likely the LLM beats the majority baseline for both zero and few-shot settings. Using the Mann-Whitney U test (Mann and Whitney 1947), we find the difference in those above the majority baseline between pre- and post-collection populations to be statistically significant at the 99% confidence level for both zero and few shot settings.

For some model/dataset combinations, the performance difference above the majority baseline is small, so we also we compute the percentage of model/dataset combinations and for which the model beats the majority baseline and the difference above the majority baseline is statistically significant at the 99% level, calculated using the student t-test (Student 1908) (Fig. 1, darker). Again, we find that for datasets released prior to the creation of the LLM, it is far more likely the LLM beats the majority baseline with statistical significance for both zero and few-shot settings. Similarly, the Mann-Whitney U test indicates these differences between

[4] The majority baseline for a classification task is the performance of a model that labels every example with the label that occurs most frequently in the dataset.



(a) GPT-3 series on pre-collection datasets

(b) GPT-3 series on post-collection datasets

(c) Open LLMs on pre-collection datasets

(d) Open LLMs on post-collection datasets

Figure 2: Percentage of datasets larger than majority baselines for each LLM (light color), as well as the percentage of tasks for which training data can be extracted with an instruction prompt (Red, see also Table 4). Dark color is the percentage of datasets significantly larger ($p = .99$) than the majority baseline using a t-test. Below each LLM, we list the training data collection year, and the total number of datasets in pre- or post-collection in parenthesis (e.g. MoE has 7 datasets post training collection date.) For tasks without demonstrated possibility of task contamination (post-collection datasets (b) and (d), with no extracted task examples in red), models rarely show statistically significant improvements over majority baselines (see §7 for details).

pre and post are statistically significant at the 99% confidence level for both zero and few shot settings.

These results indicate the possibility of task contamination for open LLMs and GPT-3 series LLMs.

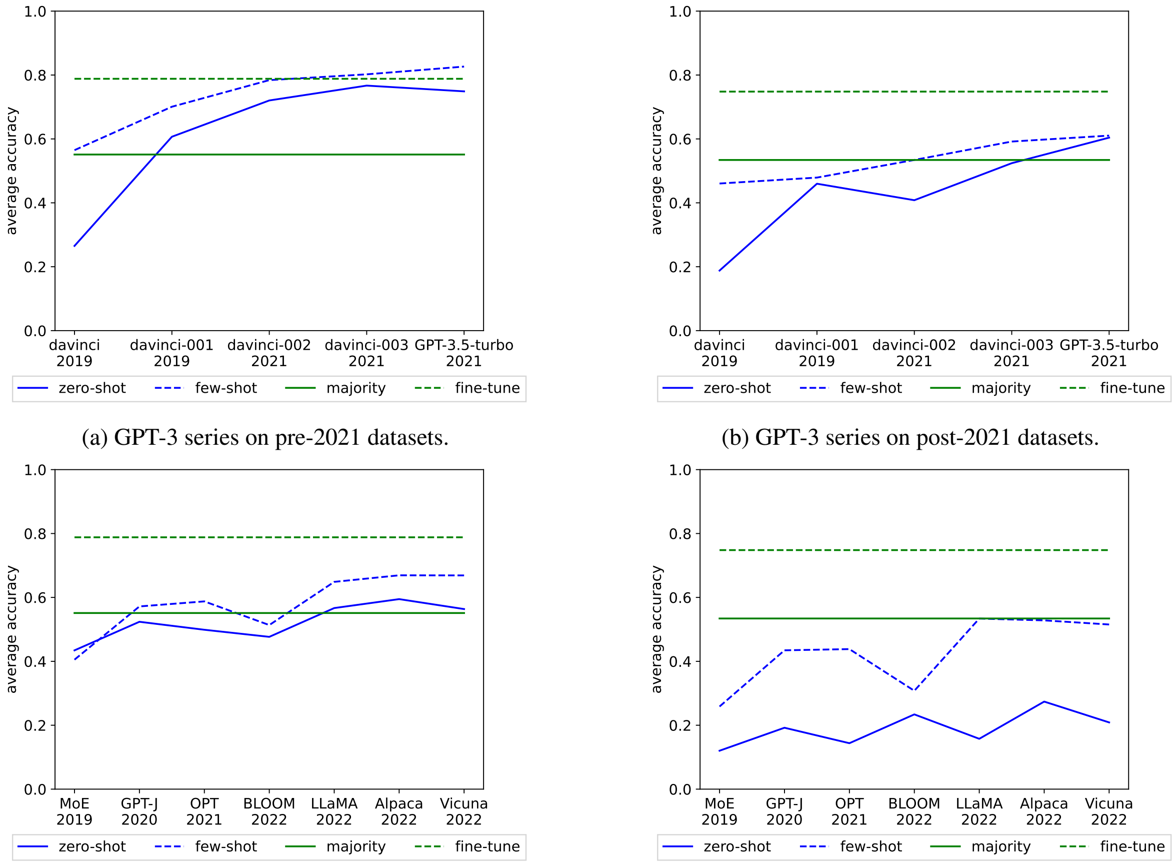
Caveats There are two considerations we need to make in the global chronological analysis.

First, datasets may have become more difficult over time, meaning LLMs are less likely to outperform the majority baseline despite the lack of task contamination. To account for this, we carefully review the tasks and remove tasks known to be difficult for LLMs, such as GSM8K (Cobbe et al. 2021) and TrackingShuffledObjects (Srivastava et al. 2023). The remaining datasets all have acceptable performance using fine-tuned pretrained language models (PLMs), and, importantly, there is no correlation between release date and the performance of fine-tuned PLMs ($R^2 = 0.001$) on our datasets, as shown in Fig. 4.

Secondly, post-collection datasets, despite being released after data collection, may still suffer from contamination. For example, the FOMC dataset (Shah, Paturi, and Chava 2023) was officially released post-collection for the GPT-3 series, but the performance of subsequent versions of GPT-3 is notably high. This may be the result of the authors' preliminary experimentation with the GPT-3 series (as stated in their paper), as OpenAI may have then utilized their experimental data for model updates.

Analysis of Pre- and Post-collection for Individual LLMs

In this section, we consider the performance on pre- and post-collection datasets for each LLM individually (see Fig. 2). We find the difference in performance between the two cat-



(a) GPT-3 series on pre-2021 datasets.

(b) GPT-3 series on post-2021 datasets.

(c) Open LLMs on pre-2021 datasets.

(d) Open LLMs on post-2021 datasets.

Figure 3: Average performance on datasets pre/post-2021. In the *x* axis, LLMs are ordered chronologically by training data collection date (collection year is listed below the LLM).

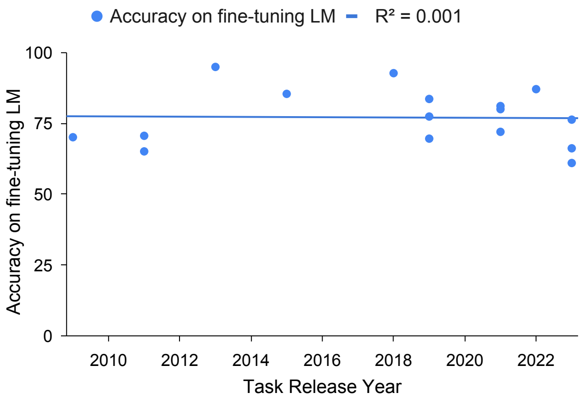


Figure 4: Task accuracy of a fine-tuned LLM baseline vs. task release year. $R^2 = .001$, which indicates that the task difficulty for our datasets does not increase over time.

egories to be statistically significant at 95% confidence according to the paired sign test (Dixon and Mood 1946).

We plot the percentage of datasets larger than the majority baseline as in the last section, but for each LLM individually. The results are shown in Fig. 2. We observe that the global trend from the previous section has remained true across models with the full range of dates, further indicating that the absolute date of the dataset is not the main factor, but rather the date of the dataset relative to the training data collection date for the LLM is the more important factor. (Note: because of the recency of BLOOM, LLaMA, Alpaca, and Vicuna, we have fewer datasets in our experiments post their training data collection date). The results indicate the possibility of task contamination for both open LLMs and GPT-3 series LLMs, with a stronger indication of contamination in the GPT-3 series with `davinci-001` and after.

Performance over Time

Next we perform a chronological analysis that examines the change in average performance over time for both GPT-3 series and open LLMs (Fig. 3). In the *x* axis, LLMs are ordered chronologically by training data collection date. To also be sensitive to time of the datasets, we split our datasets into two sets: datasets released before or after January 1st, 2021, identified as **pre-2021** datasets and **post-2021** datasets, respectively.

Pre-2021 Datasets For open LLMs, on pre-2021 datasets, we see a slight increase over time for open LLMs (Fig. 3c). We find that the performance hovers around the majority baseline for both zero and few-shot settings, and does not increase very much from LLM data collection dates ranging from 2019 to 2022.

For the GPT-3 series, on the other hand, the trend on pre-2021 datasets is particularly suspect (Fig. 3a). We see that for prior GPT-3 datasets, the performance has increased dramatically over time, with later `davinci` models much higher than the majority baseline for both zero and few-shot settings. The comparison to open LLMs indicates that zero and few-shot evaluations may have task contamination issues due to data collected from user inputs.

Post-2021 Datasets For post-2021 datasets, GPT-3 average performance has also increased over time (Fig. 3b), particularly in the zero-shot setting. This makes sense, as many of the post-2021 datasets are released prior the training data collection date for the later `davinci` models. (To see which datasets are pre- or post- training data collection time, see the line separating pre- and post- collection datasets in Table 4.) Open LLMs average performance also increased over time, but they remain lower than the majority baseline and the GPT-3 series.

One could hypothesize that the high performance of the GPT-3 series is due to instruction tuning (Ouyang et al. 2022), however we do not believe this is the case. While we observe an increase in performance from `davinci-001` to `davinci-002` on pre-2021 datasets, there is a corresponding decrease in performance on post-2021 datasets, which we measure with the sign test to be statistically significant at the 95%. This demonstrates that the GPT-3 series instruction tuning is specific to certain earlier datasets, and suggests dataset contamination for zero and few-shot evaluation of GPT-3 series.

5 Training Data Inspection

To search for direct evidence of task contamination, we conduct training data inspection on two instruction fine-tuned open LLMs (Alpaca and Vicuna) for all experimented classification tasks. We search for task-related instruction patterns in the training data, and manually inspect them to see if they contain task training examples. Because we must check manually, we can perform this analysis only for the small fine-tuning datasets of Alpaca and Vicuna. We then compare the performance to see if more task-specific training examples has boosted performance.

Table 3 shows the number of task examples on Alpaca and Vicuna, as well as the change in performance over LLaMA averaged over zero and few-shot settings and all tasks. We find that performance has improved for Alpaca and Vicuna over the original LLaMA model for tasks with more than one task example. Because Alpaca and Vicuna are fine-tuned LLaMA models, this indicates that the performance can be improved with small sets of task examples in the training data, which can compromise zero-shot or few-shot evaluation.

Dataset	Alpaca	Vicuna
RTE	0, +3.1%	**33, +10.6%**
WNLI	0, -1.4%	**33, +7.7%**
COPA	?, 0%	?, +10%
SST-2	**8, +14.6%**	0, -1.0%
MRPC	0, -0.7%	0, -8.0%
QNLI	0, -0.4%	**28, +10.0%**
CB	0, +9.8%	0, -23.2%
WiC	0, -4.9%	0, -2.5%
BoolQ	?, +1.9%	?, +4.0%
StrategyQA	0, -3.3%	0, +10.3%
MTSC-RW	?, +9.6%	?, +11.3%
MTSC-MT	?, +6.9%	?, +8.0%
NLI4Wills	0, -13.5%	0, -11.6%
CREPE	0, +24.2%	0, -0.4%
FOMC	0, -5.7%	**1, -5.4%**
NewsMet	**4, +7.2%**	0, -11.4%

Table 3: Training data inspection results: # of datapoints in the Alpaca and Vicuna datasets that are examples of the task, and Δ%, the performance difference compared to LLaMA averaged across zero and few-shot settings. Task examples are found by matching a regular expression for the task followed by a manual inspection. Bold indicates task examples are found. "?" indicates there is no specific pattern to match, so we cannot count the number of examples. Regular expressions for each task are listed in the Appendix D.

6 Task Example Extraction

We test for task data contamination by attempting to extract task examples from the LLM. Prior work (Sainz et al. 2023b) has tested if there exists testing data contamination by prompting an LLM to generate examples for a task. If the LLM can generate examples that exactly match examples in the test data, it is evidence that the test set of the task has been seen during training by the LLM. Inspired by their method, we adopt a similar approach to test for task contamination. Instead of attempting to generate test data, we prompt the model to generate training examples, since for zero- or few-shot evaluation, the model should not be trained on any task examples. If an LLM can generate training examples based on the prompt, this is evidence of task contamination. Note we do not require an exact match of the generated examples with the training data for the task, since any examples for the task seen during training indicate possible task contamination. Our prompts for task example extraction are given in Appendix H.

Table 4 shows the task example extraction results on all tasks across all models. For all **pre-collection datasets**, GPT-3 series models starting from `davinci-001` can generate task specific training examples. There are some **post-collection datasets** that have evidence of contamination for the GPT-3 series. These datasets may have been contaminated if the authors of these datasets experimented with the GPT-3 series before releasing the dataset. For example, the FOMC paper (Shah, Paturi, and Chava 2023) states they tested with the GPT-3 series, which could have caused contamination. For open LLMs, almost no models can generate training examples of specific tasks except for Vicuna, which

Task	Davinci	davinci-001	davinci-002	davinci-003	GPT-3.5-T	MoE	GPT-J	OPT	Bloom	LLaMA	Alpaca	Vicuna
RTE	▨	X	X	X	X	▨	▨	▨	▨	▨	🟩	X
WNLI	▨	X	X	X	X	▨	▨	▨	▨	▨	🟩	X
COPA	▨	🟩	🟩	X	X	▨	▨	▨	▨	▨	🟩	🟩
SST-2	▨	🟩	X	X	X	▨	▨	▨	▨	▨	🟩	🟩
MRPC	▨	🟩	🟩	X	X	▨	▨	▨	▨	▨	🟩	🟩
QNLI	▨	🟩	X	X	X	▨	▨	▨	▨	▨	🟩	🟩
CB	▨	X	X	X	X	▨	▨	▨	▨	▨	🟩	🟩
WiC	▨	🟩	X	X	X	▨	▨	▨	▨	▨	🟩	🟩
BoolQ	▨	🟩	🟩	X	X	▨	▨	▨	▨	▨	🟩	🟩
StrategyQA	▨	🟩	🟩	🟩	🟩	▨	▨	▨	▨	▨	🟩	🟩
NewsMTSC-MT	▨	🟩	🟩	X	X	▨	▨	▨	▨	▨	🟩	X
NewsMTSC-RW	▨	🟩	🟩	X	X	▨	▨	▨	▨	▨	🟩	X
NLI4Wills	▨	🟩	🟩	🟩	🟩	▨	▨	▨	▨	▨	🟩	🟩
CREPE	▨	🟩	🟩	🟩	🟩	▨	▨	▨	▨	▨	🟩	🟩
FOMC	▨	🟩	X	X	X	▨	▨	▨	▨	▨	🟩	🟩
NewsMet	▨	🟩	🟩	X	X	▨	▨	▨	▨	▨	🟩	🟩

Table 4: Task example extraction results on all tasks (tasks ordered top to bottom by release date). A line separates those datasets released before the LLM's training data collection date (pre-collection, top) and those after (post-collection, bottom) for each LLM. X indicates the model can generate training examples for the task. We indicate models with instruction tuning and those without using 🟩 and ▨, respectively. 🟩 indicates a model with instruction tuning cannot generate task examples, while ▨ indicates a model without instruction tuning cannot generate task examples. Models without instruction tuning cannot follow the instructions directing them to generate task examples.

is fine-tuned on the ChatGPT data. Note models without instruction tuning cannot follow the instructions directing them to generate task examples, so this analysis is not conclusive for these models.

Comparison to Training Data Inspection

Comparing Tables 3 and 4, we find that training data inspection (TDI) and task example extraction (TEE) both suffer from low recall. TDI has demonstrated task contamination in Alpaca for SST-2 and NewsMet datasets, but TEE failed to catch this contamination. Similarly, TEE has demonstrated task contamination for Vicuna for NewsMTSC, but TDI has failed to catch it. Both suffer from low recall, and highlight the difficulties of employing these methods for detecting task contamination.

7 LLM Performance on Tasks With No Contamination

We find that for tasks without demonstrated possibility of task contamination, LLMs rarely show statistically significant improvements over majority baselines. In Table 4, for the 51 model/dataset combinations that are post-collection and have no extracted task examples, only 1 out of 51, or 2%, demonstrate a statistically significant improvements over the majority baseline for either zero or few-shot settings. This combination is `davinci-001` on MTSC-RW, which shows a statistically significant improvement over the majority baseline (Tables 8 and 9 in the Appendix) but does not generate task examples with our prompt. This dataset is found by cross-referencing Table 4 and Tables 8 and 9 in the Appendix, and looking for datasets which are post-collection and not marked X in Table 4, and are bold in either Table 8 or 9.

8 Membership Inference

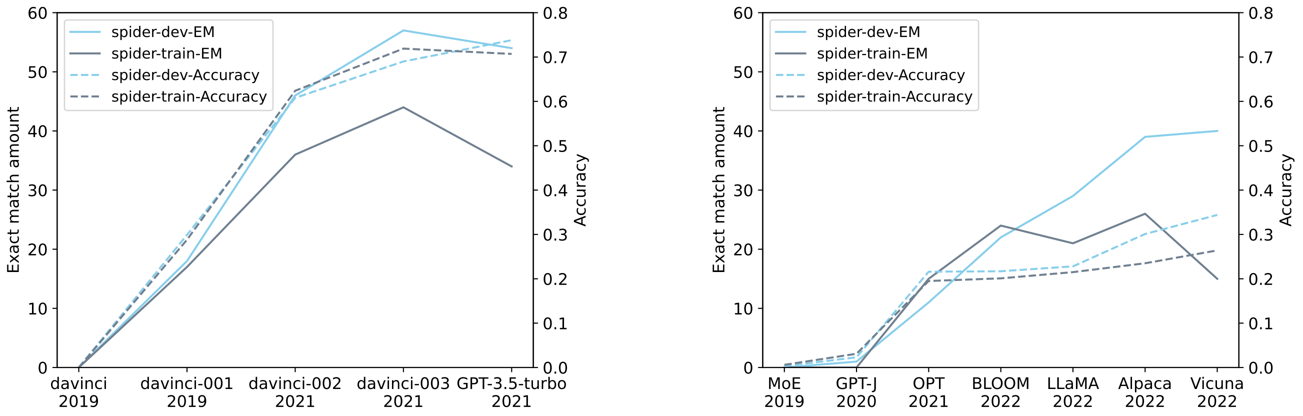
To further examine the effect of training data contamination, we apply a membership inference attack (Hu et al. 2022a), which checks if model generated content exactly matches the examples in the dataset. While this test is possible for generation tasks, it is not possible for classification tasks, since inputs may be in the training data of LLMs (and likely are, for many datasets), but we do not know for certain if the inputs are also paired with the labels without looking at the training data. We use Spider, a semantic parsing and text-to-SQL generation task, (Yu et al. 2018) as our target for analysis.

Fig. 5a and Fig. 5b show how many generated examples from the sampled training set and full development set are exactly the same over versions of the GPT-3 series and recent open sourced LLMs, respectively. The database schemas are not in the zero-shot prompts, so if the model can generate exactly the same table name or field name as found in the training or development data, there must be contamination. As shown in Fig. 5, the number of exact matched generated examples increases over time, which indicates the extent of the task contamination on Spider is increasing.

We also compute the execution accuracy after adding the schema in the prompts, and plot it against the number of exact matched generations (Fig. 6). We find a strong positive correlation between the number of exact matched generated examples and execution accuracy ($R = 0.88$), strongly indicating increased contamination is related to increased performance. However, we still cannot determine the extent of the contamination's effect on performance improvement. We leave this for future work.

9 Take-Aways

We now share some takeaways which our experiments have brought to light:



(a) Over GPT-3 series.

(b) Over recent LLMs.

Figure 5: The number of generated examples which exactly match the original set and the performance (accuracy).

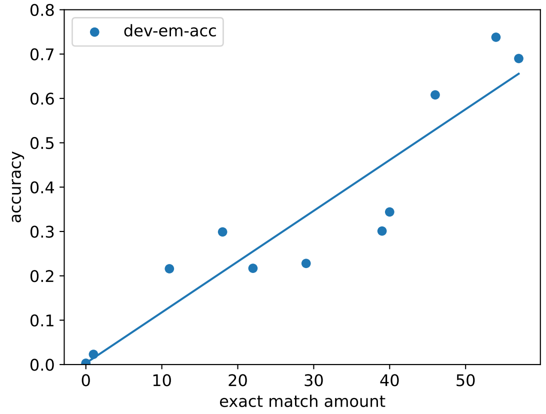


Figure 6: Membership inference: Exact match count vs. accuracy for Spider on development set. $R^2 = 0.88$

- Due to task contamination, closed-sourced models may demonstrate inflated performance in zero-shot or few-shot evaluation, and are therefore not trustworthy baselines in these settings, especially those including instruction fine-tuning or reinforcement learning with human feedback (RLHF). The extent of this contamination is still unknown, and we therefore recommend caution.
- In our experiments, for classification tasks without demonstrated possibility of task contamination, LLMs rarely show statistically significant improvements over majority baselines, in both zero and few-shot settings.
- The observed increase over time of GPT-3 series models for zero-shot or few-shot performance for many downstream tasks is likely due to task contamination.
- Inspection for task contamination of training data even for open-sourced LLMs can be difficult for several reasons. First, determining membership is difficult unless the processed dataset used for training the LLM is released (e.g., OPT and LLaMA did not release the data they used to train the model, but Alpaca and Vicuna did, so we can obtain more definite information). Second, we cannot always rely on the model to reproduce evidence of contamination even if it exists. And third, formatting differences (such as CSV and JSON) of a dataset complicate analysis.
- We encourage publicly releasing training datasets to allow for easier diagnosis of contamination issues.

10 Related Work

The investigation into potential data contamination in large language models (LLMs) has recently been gaining attention in the research community. Brown et al. (2020), in their work with GPT-3, presented an in-depth analysis of data contamination. Although they acknowledged the presence of a bug that led to data contamination in multiple datasets, their position was that it did not affect the overall performance of the model. Intriguingly, they noted that contaminated datasets outperformed the uncontaminated ones which, in a way, contradicted their original assertion. Magar and Schwartz (2022) extracted training data from GPT-2 and indicated potential leaks of private data in the pre-trained language model. Chang et al. (2023) discovered that OpenAI models were memorizing substantial amounts of copyrighted materials, which increased concern over data contamination. Aiyappa et al. (2023) highlighted the severity and scope of data contamination problems for ChatGPT evaluations. Highlighting the need for strategic interventions to address these issues, Jacovi et al. (2023) proposed several strategies for mitigating testing data contamination. Additional work has further looked into test data contamination (Sainz et al. 2023b; Zhou et al. 2023; Golchin and Surdeanu 2023; Sainz et al. 2023a; Deng et al. 2023; Oren et al. 2023; Li 2023).

The previous work listed above has investigated test data contamination, but has not considered task contamination for zero-shot or few-shot settings. Prior work has noticed our proposed task contamination problem for zero-shot or few-shot learning (Blevins, Gonen, and Zettlemoyer 2023; Briakou, Cherry, and Foster 2023), but did not systematically analyze it. Our work seeks to add to the existing knowledge by providing an exhaustive evaluation of task contamination for few-shot or zero-shot learning scenarios.

11 Conclusion and Future Work

We investigate task contamination for LLMs, and conduct a chronological analysis, training data inspection, task example extraction, and a membership inference attack to analyze it. We find evidence that some LLMs have seen task examples during pre-training for a range of tasks, and are therefore no longer zero or few-shot for these tasks. Additionally, we find that for classification tasks with no possibility of task contamination, LLMs rarely demonstrate statistically significant improvements over simple majority baselines, in both zero and few-shot settings. We recommend additional research be conducted on task contamination for zero and few-shot settings to reveal the extent and impact of task contamination for large language models in these settings.

Acknowledgements

We are grateful for valuable feedback from Nilay Patel on an earlier version of this draft. We are thankful for the computing resources provided by the Pacific Research Platform's Nautilus cluster, supported in part by National Science Foundation (NSF) awards CNS-1730158, ACI-1540112, ACI-1541349, OAC-1826967, OAC-2112167, CNS-2100237, CNS-2120019, the University of California Office of the President, and the University of California San Diego's California Institute for Telecommunications and Information Technology/Qualcomm Institute. Thanks to CENIC for the 100Gbps networks.

References

Aiyappa, R.; An, J.; Kwak, H.; and Ahn, Y.-Y. 2023. Can we trust the evaluation on ChatGPT? arXiv:2303.12767.

Artetxe, M.; Bhosale, S.; Goyal, N.; Mihaylov, T.; Ott, M.; Shleifer, S.; Lin, X. V.; Du, J.; Iyer, S.; Pasunuru, R.; Anantharaman, G.; Li, X.; Chen, S.; Akin, H.; Baines, M.; Martin, L.; Zhou, X.; Koura, P. S.; O'Horo, B.; Wang, J.; Zettlemoyer, L.; Diab, M.; Kozareva, Z.; and Stoyanov, V. 2022. Efficient Large Scale Language Modeling with Mixtures of Experts. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 11699–11732. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Bang, Y.; Cahyawijaya, S.; Lee, N.; Dai, W.; Su, D.; Wilie, B.; Lovenia, H.; Ji, Z.; Yu, T.; Chung, W.; Do, Q. V.; Xu, Y.; and Fung, P. 2023. A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity.

Blevins, T.; Gonen, H.; and Zettlemoyer, L. 2023. Prompting Language Models for Linguistic Structure. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, 6649–6663. Toronto, Canada: Association for Computational Linguistics.

Briakou, E.; Cherry, C.; and Foster, G. 2023. Searching for Needles in a Haystack: On the Role of Incidental Bilingualism in PaLM's Translation Capability. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, 9432–9452. Toronto, Canada: Association for Computational Linguistics.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners.

Chang, K. K.; Cramer, M.; Soni, S.; and Bamman, D. 2023. Speak, Memory: An Archaeology of Books Known to ChatGPT/GPT-4. arXiv:2305.00118.

Chiang, W.-L.; Li, Z.; Lin, Z.; Sheng, Y.; Wu, Z.; Zhang, H.; Zheng, L.; Zhuang, S.; Zhuang, Y.; Gonzalez, J. E.; Stoica, I.; and Xing, E. P. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality.

Clark, C.; Lee, K.; Chang, M.-W.; Kwiatkowski, T.; Collins, M.; and Toutanova, K. 2019. BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2924–2936. Minneapolis, Minnesota: Association for Computational Linguistics.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Hilton, J.; Nakano, R.; Hesse, C.; and Schulman, J. 2021. Training Verifiers to Solve Math Word Problems. *CoRR*, abs/2110.14168.

de Marneffe, M.-C.; Simons, M.; and Tonhauser, J. 2019. The CommitmentBank: Investigating projection in naturally occurring discourse.

Demszky, D.; Guu, K.; and Liang, P. 2018. Transforming Question Answering Datasets Into Natural Language Inference Datasets. *ArXiv*, abs/1809.02922.

Deng, C.; Zhao, Y.; Tang, X.; Gerstein, M.; and Cohan, A. 2023. Investigating Data Contamination in Modern Benchmarks for Large Language Models. arXiv:2311.09783.

Dixon, W. J.; and Mood, A. M. 1946. The Statistical Sign Test. *Journal of the American Statistical Association*, 41(236): 557–566.

Dolan, W. B.; and Brockett, C. 2005. Automatically Constructing a Corpus of Sentential Paraphrases. In *Proceedings of the 3rd International Workshop on Paraphrasing (IWP2005)*.

Geva, M.; Khashabi, D.; Segal, E.; Khot, T.; Roth, D.; and Berant, J. 2021. Did Aristotle Use a Laptop? A Question Answering Benchmark with Implicit Reasoning Strategies. *Transactions of the Association for Computational Linguistics*, 9: 346–361.

Giampiccolo, D.; Dang, H. T.; Magnini, B.; Dagan, I.; Cabrio, E.; and Dolan, W. B. 2008. The Fourth PASCAL Recognizing Textual Entailment Challenge. In *Text Analysis Conference*.

Golchin, S.; and Surdeanu, M. 2023. Time Travel in LLMs: Tracing Data Contamination in Large Language Models. arXiv:2308.08493.

Hamborg, F.; and Donnay, K. 2021. NewsMTSC: A Dataset for (Multi-)Target-dependent Sentiment Classification in Political News Articles. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 1663–1675. Online: Association for Computational Linguistics.

Hu, H.; Salcic, Z.; Sun, L.; Dobbie, G.; Yu, P. S.; and Zhang, X. 2022a. Membership inference attacks on machine learning: A survey. *ACM Computing Surveys (CSUR)*, 54(11s): 1–37.

Hu, Y.; Lee, C.-H.; Xie, T.; Yu, T.; Smith, N. A.; and Ostendorf, M. 2022b. In-Context Learning for Few-Shot Dialogue State Tracking. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, 2627–2643. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Jacovi, A.; Caciularu, A.; Goldman, O.; and Goldberg, Y. 2023. Stop Uploading Test Data in Plain Text: Practical Strategies for Mitigating Data Contamination by Evaluation Benchmarks. arXiv:2305.10160.

Joseph, R.; Liu, T.; Ng, A. B.; See, S.; and Rai, S. 2023. NewsMet : A 'do it all' Dataset of Contemporary Metaphors in News Headlines. In *Findings of the Association for Computational Linguistics: ACL 2023*, 10090–10104. Toronto, Canada: Association for Computational Linguistics.

Kwak, A.; Israelsen, J.; Morrison, C.; Bambauer, D.; and Surdeanu, M. 2022. Validity Assessment of Legal Will Statements as Natural Language Inference. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, 6047–6056. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 3045–3059. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Levesque, H. J.; Davis, E.; and Morgenstern, L. 2012. The Winograd schema challenge. *KR*, 2012: 13th.

Li, Y. 2023. Estimating Contamination via Perplexity: Quantifying Memorisation in Language Model Evaluation. arXiv:2309.10677.

Magar, I.; and Schwartz, R. 2022. Data Contamination: From Memorization to Exploitation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 157–165. Dublin, Ireland: Association for Computational Linguistics.

Mann, H. B.; and Whitney, D. R. 1947. On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other. *The Annals of Mathematical Statistics*, 18(1): 50–60.

OpenAI. 2023a. OpenAI Examples.

OpenAI. 2023b. OpenAI Models.

Oren, Y.; Meister, N.; Chatterji, N.; Ladhak, F.; and Hashimoto, T. B. 2023. Proving Test Set Contamination in Black Box Language Models. arXiv:2310.17623.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Gray, A.; Schulman, J.; Hilton, J.; Kelton, F.; Miller, L.; Simens, M.; Askell, A.; Welinder, P.; Christiano, P.; Leike, J.; and Lowe, R. 2022. Training language models to follow instructions with human feedback. In Oh, A. H.; Agarwal, A.; Belgrave, D.; and Cho, K., eds., *Advances in Neural Information Processing Systems*.

Pilehvar, M. T.; and Camacho-Collados, J. 2019. WiC: the Word-in-Context Dataset for Evaluating Context-Sensitive Meaning Representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1267–1273. Minneapolis, Minnesota: Association for Computational Linguistics.

Poesia, G.; Polozov, A.; Le, V.; Tiwari, A.; Soares, G.; Meek, C.; and Gulwani, S. 2022. Synchromesh: Reliable Code Generation from Pre-trained Language Models. In *International Conference on Learning Representations*.

Qin, C.; Zhang, A.; Zhang, Z.; Chen, J.; Yasunaga, M.; and Yang, D. 2023. Is ChatGPT a General-Purpose Natural Language Processing Task Solver?

Qin, G.; and Eisner, J. 2021. Learning How to Ask: Querying LMs with Mixtures of Soft Prompts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 5203–5212. Online: Association for Computational Linguistics.

Roemmele, M.; Bejan, C. A.; and Gordon, A. S. 2011. Choice of Plausible Alternatives: An Evaluation of Commonsense Causal Reasoning. In *AAAI spring symposium: logical formalizations of commonsense reasoning*, 90–95.

Sainz, O.; Campos, J.; García-Ferrero, I.; Etxaniz, J.; de Lacalle, O. L.; and Agirre, E. 2023a. NLP Evaluation in trouble: On the Need to Measure LLM Data Contamination for each Benchmark. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Findings of the Association for Computational Linguistics: EMNLP 2023*.

Sainz, O.; Campos, J. A.; García-Ferrero, I.; Etxaniz, J.; and Agirr, E. 2023b. Did ChatGPT cheat on your test? https://hitz-zentroa.github.io/lm-contamination/blog/.

Scao, T. L.; Fan, A.; Akiki, C.; Pavlick, E.; Ilić, S.; Hesslow, D.; Castagné, R.; Luccioni, A. S.; Yvon, F.; Gallé, M.; et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Schick, T.; Dwivedi-Yu, J.; Dessì, R.; Raileanu, R.; Lomeli, M.; Zettlemoyer, L.; Cancedda, N.; and Scialom, T. 2023. Toolformer: Language Models Can Teach Themselves to Use Tools.

Schick, T.; and Schütze, H. 2021a. Exploiting Cloze-Questions for Few-Shot Text Classification and Natural Language Inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 255–269. Online: Association for Computational Linguistics.

Schick, T.; and Schütze, H. 2021b. Few-Shot Text Generation with Natural Language Instructions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 390–402. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Shah, A.; Paturi, S.; and Chava, S. 2023. Trillion Dollar Words: A New Financial Dataset, Task & Market Analysis. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, 6664–6679. Toronto, Canada: Association for Computational Linguistics.

Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A.; and Potts, C. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 1631–1642. Seattle, Washington, USA: Association for Computational Linguistics.

Srivastava, A.; Rastogi, A.; Rao, A.; Shoeb, A. A. M.; Abid, A.; Fisch, A.; Brown, A. R.; Santoro, A.; Gupta, A.; Garriga-Alonso, A.; Kluska, A.; Lewkowycz, A.; Agarwal, A.; Power, A.; Ray, A.; Warstadt, A.; Kocurek, A. W.; Safaya, A.; Tazarv, A.; Xiang, A.; Parrish, A.; Nie, A.; Hussain, A.; Askell, A.; Dsouza, A.; Slone, A.; Rahane, A.; Iyer, A. S.; Andreassen, A. J.; Madotto, A.; Santilli, A.; Stuhlmüller, A.; Dai, A. M.; La, A.; Lampinen, A.; Zou, A.; Jiang, A.; Chen, A.; Vuong, A.; Gupta, A.; Gottardi, A.; Norelli, A.; Venkatesh, A.; Gholamidavoodi, A.; Tabassum, A.; Menezes, A.; Kirubarajan, A.; Mullokandov, A.; Sabharwal, A.; Herrick, A.; Efrat, A.; Erdem, A.; Karakaş, A.; Roberts, B. R.; Loe, B. S.; Zoph, B.; Bojanowski, B.; Özyurt, B.; Hedayatnia, B.; Neyshabur, B.; Inden, B.; Stein, B.; Ekmekci, B.; Lin, B. Y.; Howald, B.; Orinion, B.; Diao, C.; Dour, C.; Stinson, C.; Argueta, C.; Ferri, C.; Singh, C.; Rathkopf, C.; Meng, C.; Baral, C.; Wu, C.; Callison-Burch, C.; Waites, C.; Voigt, C.; Manning, C. D.; Potts, C.; Ramirez, C.; Rivera, C. E.; Siro, C.; Raffel, C.; Ashcraft, C.; Garbacea, C.; Sileo, D.; Garrette, D.; Hendrycks, D.; Kilman, D.; Roth, D.; Freeman, C. D.; Khashabi, D.; Levy, D.; González, D. M.; Perszyk, D.; Hernandez, D.; Chen, D.; Ippolito, D.; Gilboa, D.; Dohan, D.; Drakard, D.; Jurgens, D.; Datta, D.; Ganguli, D.; Emelin, D.; Kleyko, D.; Yuret, D.; Chen, D.; Tam, D.; Hupkes, D.; Misra, D.; Buzan, D.; Mollo, D. C.; Yang, D.; Lee, D.-H.; Schrader, D.; Shutova, E.; Cubuk, E. D.; Segal, E.; Hagerman, E.; Barnes, E.; Donoway, E.; Pavlick, E.; Rodolà, E.; Lam, E.; Chu, E.; Tang, E.; Erdem, E.; Chang, E.; Chi, E. A.; Dyer, E.; Jerzak, E.; Kim, E.; Manyasi, E. E.; Zheltonozhskii, E.; Xia, F.; Siar, F.; Martínez-Plumed, F.; Happé, F.; Chollet, F.; Rong, F.; Mishra, G.; Winata, G. I.; de Melo, G.; Kruszewski, G.; Parascandolo, G.; Mariani, G.; Wang, G. X.; Jaimovitch-Lopez, G.; Betz, G.; Gur-Ari, G.; Galijasevic, H.; Kim, H.; Rashkin, H.; Hajishirzi, H.; Mehta, H.; Bogar, H.; Shevlin, H. F. A.; Schuetze, H.; Yakura, H.; Zhang, H.; Wong, H. M.; Ng, I.; Noble, I.; Jumelet, J.; Geissinger, J.; Kernion, J.; Hilton, J.; Lee, J.; Fisac, J. F.; Simon, J. B.; Koppel, J.; Zheng, J.; Zou, J.; Kocon, J.; Thompson, J.; Wingfield, J.; Kaplan, J.; Radom, J.; Sohl-Dickstein, J.; Phang, J.; Wei, J.; Yosinski, J.; Novikova, J.; Bosscher, J.; Marsh, J.; Kim, J.; Taal, J.; Engel, J.; Alabi, J.; Xu, J.; Song, J.; Tang, J.; Waweru, J.; Burden, J.; Miller, J.; Balis, J. U.; Batchelder, J.; Berant, J.; Frohberg, J.; Rozen, J.; Hernandez-Orallo, J.; Boudeman, J.; Guerr, J.; Jones, J.; Tenenbaum, J. B.; Rule, J. S.; Chua, J.; Kanclerz, K.; Livescu, K.; Krauth, K.; Gopalakrishnan, K.; Ignatyeva, K.; Markert, K.; Dhole, K.; Gimpel, K.; Omondi, K.; Mathewson, K. W.; Chiafullo, K.; Shkaruta, K.; Shridhar, K.; McDonell, K.; Richardson, K.; Reynolds, L.; Gao, L.; Zhang, L.; Dugan, L.; Qin, L.; Contreras-Ochando, L.; Morency, L.-P.; Moschella, L.; Lam, L.; Noble, L.; Schmidt, L.; He, L.; Oliveros-Colón, L.; Metz, L.; Senel, L. K.; Bosma, M.; Sap, M.; Hoeve, M. T.; Farooqi, M.; Faruqui, M.; Mazeika, M.; Baturan, M.; Marelli, M.; Maru, M.; Ramirez-Quintana, M. J.; Tolkiehn, M.; Giulianelli, M.; Lewis, M.; Potthast, M.; Leavitt, M. L.; Hagen, M.; Schubert, M.; Baitemirova, M. O.; Arnaud, M.; McElrath, M.; Yee, M. A.; Cohen, M.; Gu, M.; Ivanitskiy, M.; Starritt, M.; Strube, M.; Swędrowski, M.; Bevilacqua, M.; Yasunaga, M.; Kale, M.; Cain, M.; Xu, M.; Suzgun, M.; Walker, M.; Tiwari, M.; Bansal, M.; Aminnaseri, M.; Geva, M.; Gheini, M.; T, M. V.; Peng, N.; Chi, N. A.; Lee, N.; Krakover, N. G.-A.; Cameron, N.; Roberts, N.; Doiron, N.; Martinez, N.; Nangia, N.; Deckers, N.; Muennighoff, N.; Keskar, N. S.; Iyer, N. S.; Constant, N.; Fiedel, N.; Wen, N.; Zhang, O.; Agha, O.; Elbaghdadi, O.; Levy, O.; Evans, O.; Casares, P. A. M.; Doshi, P.; Fung, P.; Liang, P. P.; Vicol, P.; Alipoormolabashi, P.; Liao, P.; Liang, P.; Chang, P. W.; Eckersley, P.; Htut, P. M.; Hwang, P.; Miłkowski, P.; Patil, P.; Pezeshkpour, P.; Oli, P.; Mei, Q.; Lyu, Q.; Chen, Q.; Banjade, R.; Rudolph, R. E.; Gabriel, R.; Habacker, R.; Risco, R.; Millière, R.; Garg, R.; Barnes, R.; Saurous, R. A.; Arakawa, R.; Raymaekers, R.; Frank, R.; Sikand, R.; Novak, R.; Sitelew, R.; Bras, R. L.; Liu, R.; Jacobs, R.; Zhang, R.; Salakhutdinov, R.; Chi, R. A.; Lee, S. R.; Stovall, R.; Teehan, R.; Yang, R.; Singh, S.; Mohammad, S. M.; Anand, S.; Dillavou, S.; Shleifer, S.; Wiseman, S.; Gruetter, S.; Bowman, S. R.; Schoenholz, S. S.; Han, S.; Kwatra, S.; Rous, S. A.; Ghazarian, S.; Ghosh, S.; Casey, S.; Bischoff, S.; Gehrmann, S.; Schuster, S.; Sadeghi, S.; Hamdan, S.; Zhou, S.; Srivastava, S.; Shi, S.; Singh, S.; Asaadi, S.; Gu, S. S.; Pachchigar, S.; Toshniwal, S.; Upadhyay, S.; Debnath, S. S.; Shakeri, S.; Thormeyer, S.; Melzi, S.; Reddy, S.; Makini, S. P.; Lee, S.-H.; Torene, S.; Hatwar, S.; Dehaene, S.; Divic, S.; Ermon, S.; Biderman, S.; Lin, S.; Prasad, S.; Piantadosi, S.; Shieber, S.; Misherghi, S.; Kiritchenko, S.; Mishra, S.; Linzen, T.; Schuster, T.; Li, T.; Yu, T.; Ali, T.; Hashimoto, T.; Wu, T.-L.; Desbordes, T.; Rothschild, T.; Phan, T.; Wang, T.; Nkinyili, T.; Schick, T.; Kornev, T.; Tunduny, T.; Gerstenberg, T.; Chang, T.; Neeraj, T.; Khot, T.; Shultz, T.; Shaham, U.; Misra, V.; Demberg, V.; Nyamai, V.; Raunak, V.; Ramasesh, V. V.; vinay uday prabhu; Padmakumar, V.; Srikumar, V.; Fedus, W.; Saunders, W.; Zhang, W.; Vossen, W.; Ren, X.; Tong, X.; Zhao, X.; Wu, X.; Shen, X.; Yaghoobzadeh, Y.; Lakretz, Y.; Song, Y.; Bahri, Y.; Choi, Y.; Yang, Y.; Hao, Y.; Chen, Y.; Belinkov, Y.; Hou, Y.; Hou, Y.; Bai, Y.; Seid, Z.; Zhao, Z.; Wang, Z.; Wang, Z. J.; Wang, Z.; and Wu, Z. 2023. Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

Student. 1908. The probable error of a mean. *Biometrika*, 1–25.

Taori, R.; Gulrajani, I.; Zhang, T.; Dubois, Y.; Li, X.; Guestrin, C.; Liang, P.; and Hashimoto, T. B. 2023. Stanford Alpaca: An Instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; Rodriguez, A.; Joulin, A.; Grave, E.; and Lample, G. 2023. LLaMA: Open and Efficient Foundation Language Models. arXiv:2302.13971.

Wang, A.; Pruksachatkun, Y.; Nangia, N.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; and Bowman, S. R. 2019. *SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems*. Red Hook, NY, USA: Curran Associates Inc.

Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; and Bowman, S. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 353–355. Brussels, Belgium: Association for Computational Linguistics.

Wang, B.; Deng, X.; and Sun, H. 2022. Iteratively Prompt Pre-trained Language Models for Chain of Thought. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2714–2730. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Wang, B.; and Komatsuzaki, A. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.

Wang, X.; Wei, J.; Schuurmans, D.; Le, Q. V.; Chi, E. H.; Narang, S.; Chowdhery, A.; and Zhou, D. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *The Eleventh International Conference on Learning Representations*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Chi, E.; Le, Q.; and Zhou, D. 2022. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Yang, R.; Song, L.; Li, Y.; Zhao, S.; Ge, Y.; Li, X.; and Shan, Y. 2023. GPT4Tools: Teaching Large Language Model to Use Tools via Self-instruction. arXiv:2305.18752.

Yu, T.; Zhang, R.; Yang, K.; Yasunaga, M.; Wang, D.; Li, Z.; Ma, J.; Li, I.; Yao, Q.; Roman, S.; Zhang, Z.; and Radev, D. 2018. Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 3911–3921. Brussels, Belgium: Association for Computational Linguistics.

Yu, X.; Min, S.; Zettlemoyer, L.; and Hajishirzi, H. 2023. CREPE: Open-Domain Question Answering with False Presuppositions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, 10457–10480. Toronto, Canada: Association for Computational Linguistics.

Zhang, S.; Roller, S.; Goyal, N.; Artetxe, M.; Chen, M.; Chen, S.; Dewan, C.; Diab, M.; Li, X.; Lin, X. V.; Mihaylov, T.; Ott, M.; Shleifer, S.; Shuster, K.; Simig, D.; Koura, P. S.; Sridhar, A.; Wang, T.; and Zettlemoyer, L. 2022. OPT: Open Pre-trained Transformer Language Models. arXiv:2205.01068.

Zhou, K.; Zhu, Y.; Chen, Z.; Chen, W.; Zhao, W. X.; Chen, X.; Lin, Y.; Wen, J.-R.; and Han, J. 2023. Don't Make Your LLM an Evaluation Benchmark Cheater. arXiv:2311.01964.

A Hyperparameters

We use greedy decoding to ensure a fair comparison for all approaches. For GPT-3 series models, we set the temperature as 0 to ensure deterministic results. For few-shot learning, we use the same few-shot examples across models for each instance in a task. We run open sourced models on an NVIDIA A100 GPU.

B Datasets

The pre-2021 datasets are common GLUE (Wang et al. 2018) and Super GLUE (Wang et al. 2019) tasks: MRPC (Dolan and Brockett 2005), boolq (Clark et al. 2019), SST-2 (Socher et al. 2013), QNLI (Demszky, Guu, and Liang 2018), WNLI (Levesque, Davis, and Morgenstern 2012), RTE (Giampiccolo et al. 2008), CB (de Marneffe, Simons, and Tonhauser 2019), COPA (Roemmele, Bejan, and Gordon 2011), WiC (Pilehvar and Camacho-Collados 2019). The post-2021 datasets are StrategyQA (Geva et al. 2021), NLI4Wills (Kwak et al. 2022), NewsMTSC (Hamborg and Donnay 2021), CREPE (Yu et al. 2023), FOMC (Shah, Paturi, and Chava 2023) and NewsMet (Joseph et al. 2023).

Dataset	Year	Test set size
RTE	2009	277
WNLI	2011	71
COPA	2011	100
SST-2	2013	872
MRPC	2015	408
QNLI	2018	5463
CB	2019	56
WiC	2019	638
BoolQ	2019	3270
StrategyQA	2021	229
NewsMTSC-mt	2021	1476
NewsMTSC-rw	2021	1146
NLI4Wills	2022	255
CREPE	2023	2000
FOMC	2023	496
NewsMet	2023	554

Table 5: Dataset release year and test set size for each task.

C Prompt Sources

The prompts for these tasks are taken from previous research (Bang et al. 2023; Qin et al. 2023) that use them as evaluation benchmarks and OpenAI (2023a) Examples or designed based on the related tasks from these sources. Table 6 shows prompt source for each dataset. Appendix G lists example prompts for each task.

D Training Data Inspection Details

We manually inspect training examples found using regular expressions for each task. Our regular expression or string search pattern for each task are listed in Table 7. Some tasks such as COPA and BoolQ do not have a specific pattern that can be matched. We count an example if it is directly related to the task and contains the input and output for the task. We do not count examples that talk about the task without giving input and output examples.

Dataset	Prompt source
RTE	Bang et al. (2023)*
WNLI	Bang et al. (2023)*
COPA	Bang et al. (2023)*
SST-2	OpenAI (2023a)
MRPC	OpenAI (2023a)*
QNLI	Bang et al. (2023)*
CB	Bang et al. (2023)*
WiC	OpenAI (2023a)*
BoolQ	Qin et al. (2023)*
StrategyQA	Qin et al. (2023)
Newsmtsc-mt	OpenAI (2023a)*
Newsmtsc-rw	OpenAI (2023a)*
NLI4Wills	Bang et al. (2023)*
CREPE	OpenAI (2023a)*
FOMC	Shah, Paturi, and Chava (2023)
NewsMet	Bang et al. (2023)*

Table 6: Prompt source for each task. * indicates we designed our prompt based on the referenced source.

Dataset	RE pattern
RTE	[Ee]ntailment
WNLI	[Ee]ntailment
COPA	–
SST-2	[cC]lassify the sentiment
MRPC	[Pp]paraphrase
QNLI	[Ee]ntailment
CB	[Ee]ntailment
WiC	[Ww]ord sense
BoolQ	–
StrategyQA	([tT]he answer is)*([Yy]es|[Nn]o)
NLI4Wills	[sS]upport|[Rr]efute
MTSC-RW	–
MTSC-MT	–
CREPE	presupposition
FOMC	"hawkish" or "dovish"
NewsMet	"metaphorical"

Table 7: RE patterns used for each task. – indicates there is no specific pattern to match for this task.

E Detailed Results Tables

In this section, we report the performance numbers for all models and datasets in our experiments with confidence intervals.

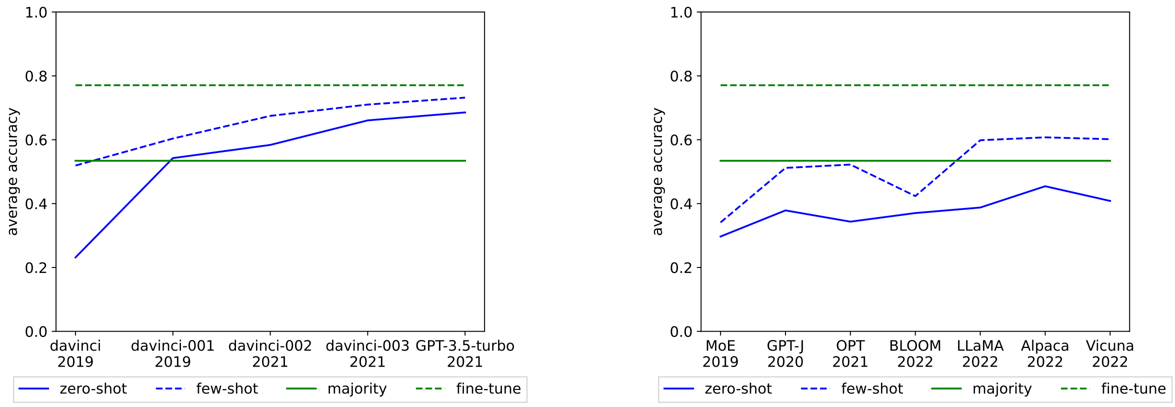
Dataset	Majority	davinci	davinci-001	davinci-002	davinci-003	GPT-3.5-T	MoE-7B	GPT-J-6B	OPT-6.7B	BLOOM-7B	LLama-7B	Alpaca-7B	Vicuna-7B
RTE	52.7	29.6±2.9	57.4±3.5	**75.1±2.6**	**83.8±1.9**	**72.6±2.8**	61.7±3.3	53.1±3.5	53.1±3.5	52.7±3.5	63.2±3.3	54.9±3.5	60.7±3.4
WNLI	56.3	33.8±6.4	43.7±7.0	66.2±6.4	60.6±6.8	66.2±6.4	45.1±7.1	43.7±7.0	43.7±7.0	43.7±7.0	46.5±7.1	43.7±7.0	43.7±7.0
COPA	55.0	66.0±5.4	70.0±5.0	**89.0±2.3**	**93.0±1.6**	**82.0±3.5**	56.0±5.9	50.0±6.0	53.0±5.9	53.0±5.9	55.0±5.9	58.0±5.8	72.0±4.8
SST-2	50.9	0.3±0.0	58.0±1.9	**85.1±1.0**	**73.4±1.5**	**81.8±1.2**	5.4±0.4	49.1±2.0	34.7±1.8	53.4±2.0	57.8±1.9	**87.3±0.9**	**62.0±1.9**
MRPC	68.4	9.3±1.0	68.4±2.5	68.4±2.5	72.5±2.3	69.9±2.4	34.8±2.6	69.9±2.4	55.6±2.9	31.6±2.5	68.9±2.5	68.4±2.5	68.4±2.5
QNLI	50.5	28.0±0.6	49.5±0.8	**57.2±0.8**	**84.6±0.4**	**85.1±0.4**	**55.0±0.8**	49.7±0.8	53.0±0.8	49.5±0.8	51.5±0.8	49.6±0.8	**59.0±0.8**
CB	50.0	35.7±7.5	75.0±6.1	75.0±6.1	76.8±5.8	75.0±6.1	26.8±6.4	44.6±8.1	41.1±7.9	50.0±8.1	41.1±7.9	48.2±8.1	12.5±3.6
WiC	50.0	16.3±1.2	45.5±2.2	48.9±2.2	**60.5±2.1**	54.4±2.2	50.3±2.2	51.3±2.2	55.3±2.2	50.5±2.2	**59.6±2.2**	50.3±2.2	52.7±2.2
BoolQ	62.2	19.6±0.6	**78.7±0.6**	**83.5±0.5**	**85.0±0.5**	**87.1±0.4**	55.8±0.9	60.1±0.9	59.5±0.9	44.6±0.9	**66.5±0.8**	**74.9±0.7**	**76.3±0.7**
StrategyQA	53.3	31.9±3.4	55.9±3.8	53.7±3.9	62.0±3.7	65.1±3.5	46.7±3.9	23.6±2.8	12.2±1.7	24.0±2.8	36.2±3.6	21.8±2.7	53.3±3.9
MTSC-MT	50.7	3.3±0.2	48.8±1.5	34.8±1.4	**63.8±1.4**	**67.1±1.3**	0.0±0.0	4.2±0.2	2.6±0.2	3.3±0.2	2.2±0.1	5.1±0.3	12.3±0.7
MTSC-RW	39.7	4.5±0.3	**50.4±1.7**	34.8±1.6	**60.9±1.6**	**69.2±1.5**	0.0±0.0	4.3±0.3	3.1±0.2	3.3±0.2	2.3±0.2	7.8±0.5	10.7±0.7
NLI4Wills	55.7	17.6±2.1	23.1±2.6	15.7±1.9	33.7±3.3	41.6±3.6	14.5±1.8	14.5±1.8	2.0±0.3	3.5±0.5	7.1±1.0	19.2±2.3	21.6±2.5
CREPE	72.8	20.5±0.9	40.1±1.3	28.1±1.1	42.1±1.3	69.3±1.1	4.1±0.2	16.5±0.7	44.3±1.3	68.5±1.1	20.4±0.8	67.2±1.1	18.1±0.8
FOMC	49.4	33.3±2.3	52.6±2.6	**61.5±2.5**	54.0±2.6	59.5±2.5	11.1±1.0	24.2±1.9	11.5±1.1	25.0±2.0	39.1±2.5	25.0±2.0	28.4±2.1
NewsMet	52.3	20.4±1.6	50.9±2.5	57.0±2.4	50.2±2.5	51.1±2.5	7.8±0.7	47.5±2.5	34.8±2.3	36.1±2.3	31.0±2.1	46.9±2.5	8.7±0.8

Table 8: Zero-shot performances on experimented LLMs and datasets. Datasets above the single line are pre- LMM training data collection datasets. Confidence intervals are computed using a t-distribution. Bold text indicates significantly larger than the majority baseline using a t-test with $p = .99$. A graphical representation of this data is in Figs. 8 and 9.

Dataset	Majority	davinci	davinci-001	davinci-002	davinci-003	GPT-3.5-T	MoE-7B	GPT-J-6B	OPT-6.7B	BLOOM-7B	LLama-7B	Alpaca-7B	Vicuna-7B
RTE	52.7	50.5±3.5	65.0±3.2	**83.4±2.0**	**85.6±1.7**	**84.8±1.8**	46.6±3.5	46.6±3.5	62.8±3.3	51.6±3.5	48.0±3.5	62.5±3.3	**71.8±2.9**
WNLI	56.3	57.7±7.0	46.5±7.1	60.6±6.8	71.8±5.8	**85.9±3.5**	56.3±7.0	46.5±7.1	43.7±7.0	52.1±7.1	46.5±7.1	46.5±7.1	64.8±6.5
COPA	55.0	47.0±5.9	**83.0±3.4**	**96.0±0.9**	**96.0±0.9**	**97.0±0.7**	**90.0±2.1**	45.0±5.9	54.0±5.9	45.0±5.9	69.0±5.1	66.0±5.4	72.0±4.8
SST-2	50.9	**91.7±0.6**	**92.7±0.5**	**92.2±0.6**	**78.2±1.3**	**90.1±0.7**	1.7±0.1	**79.5±1.3**	**87.4±0.9**	**84.7±1.0**	**93.6±0.5**	**93.2±0.5**	**87.3±0.9**
MRPC	68.4	52.7±2.9	69.1±2.5	71.6±2.4	77.0±2.1	72.8±2.3	31.6±2.5	**85.3±1.5**	67.2±2.6	31.6±2.5	69.4±2.5	68.4±2.5	53.9±2.9
QNLI	50.5	51.7±0.8	**59.0±0.8**	**79.0±0.5**	**79.9±0.5**	**84.4±0.4**	50.6±0.8	49.5±0.8	**55.6±0.8**	52.1±0.8	**57.7±0.8**	**58.8±0.8**	**70.3±0.7**
CB	50.0	50.0±8.1	**80.4±5.1**	78.6±5.5	78.6±5.5	**80.4±5.1**	0.0±0.0	44.6±8.1	41.1±7.9	41.1±7.9	71.4±6.6	**83.9±4.4**	53.6±8.1
WiC	50.0	51.1±2.2	55.6±2.2	57.2±2.2	**66.5±2.0**	**63.2±2.1**	50.0±2.2	54.9±2.2	50.2±2.2	51.3±2.2	50.5±2.2	49.8±2.2	52.4±2.2
BoolQ	62.2	55.8±0.9	**79.5±0.6**	**87.1±0.4**	**88.4±0.4**	**85.1±0.5**	37.9±0.9	62.9±0.9	**66.9±0.8**	52.6±1.0	**77.8±0.7**	**73.2±0.7**	**76.0±0.7**
StrategyQA	53.3	52.4±3.9	58.5±3.8	62.4±3.6	**70.3±3.2**	**69.0±3.3**	48.5±3.9	45.0±3.8	52.8±3.9	49.8±3.9	53.3±3.9	61.1±3.7	56.8±3.8
MTSC-MT	50.7	40.0±1.5	43.2±1.5	**61.0±1.4**	**68.4±1.3**	**70.7±1.3**	0.1±0.0	36.7±1.4	24.1±1.1	2.9±0.2	48.3±1.5	**59.2±1.5**	54.3±1.5
MTSC-RW	39.7	33.2±1.5	**52.9±1.7**	**66.8±1.5**	**64.6±1.6**	**69.4±1.5**	0.1±0.0	31.0±1.5	30.8±1.5	3.1±0.2	41.4±1.7	**55.2±1.7**	**55.7±1.7**
NLI4Wills	55.7	47.1±3.7	30.2±3.1	5.1±0.7	28.2±3.0	36.5±3.4	0.4±0.1	21.6±2.5	24.3±2.7	54.9±3.6	56.9±3.6	17.6±2.1	19.2±2.3
CREPE	72.8	60.9±1.2	44.9±1.3	73.8±1.0	70.9±1.1	62.2±1.2	67.7±1.1	72.8±1.0	72.8±1.0	14.8±0.7	71.2±1.1	72.8±1.0	72.8±1.0
FOMC	49.4	40.7±2.5	54.4±2.6	55.2±2.6	**61.7±2.5**	**63.5±2.4**	25.0±2.0	49.4±2.6	49.4±2.6	42.3±2.6	50.2±2.6	52.8±2.6	50.0±2.6
NewsMet	52.3	48.0±2.5	51.3±2.5	49.5±2.5	50.2±2.5	56.0±2.4	39.4±2.4	47.7±2.5	52.5±2.5	47.7±2.5	52.3±2.5	50.9±2.5	52.0±2.5

Table 9: Few-shot performances on GPT-series models. Datasets above the single line are pre- LMM training data collection datasets. Confidence intervals are computed using a t-distribution. Bold text indicates significantly larger than the majority baseline using a t-test with $p = .99$. A graphical representation of this data is in Figs. 8 and 9.

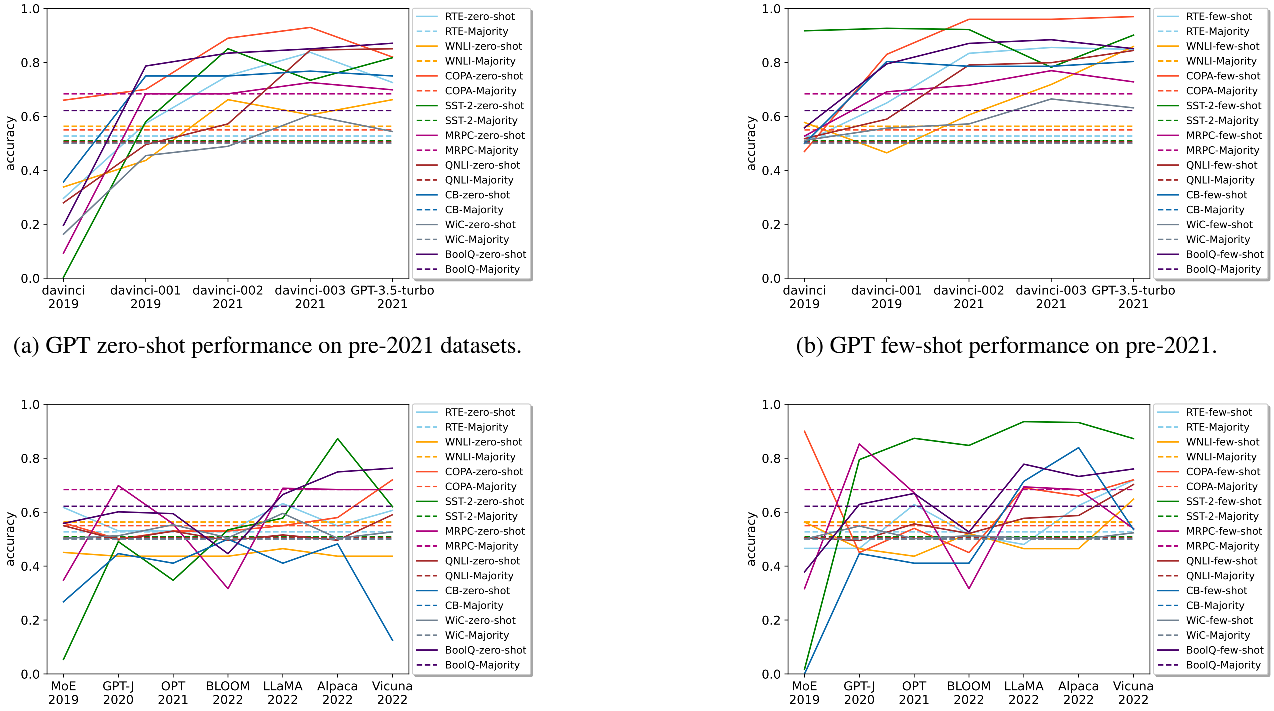
F Additional Figures



(a) GPT-3 series

(b) Open LLMs

Figure 7: Average performance across all datasets for GPT-3 series and open LLMs. In the x axis, LLMs are ordered chronologically by training data collection date, and the collection year is listed below the LLM.



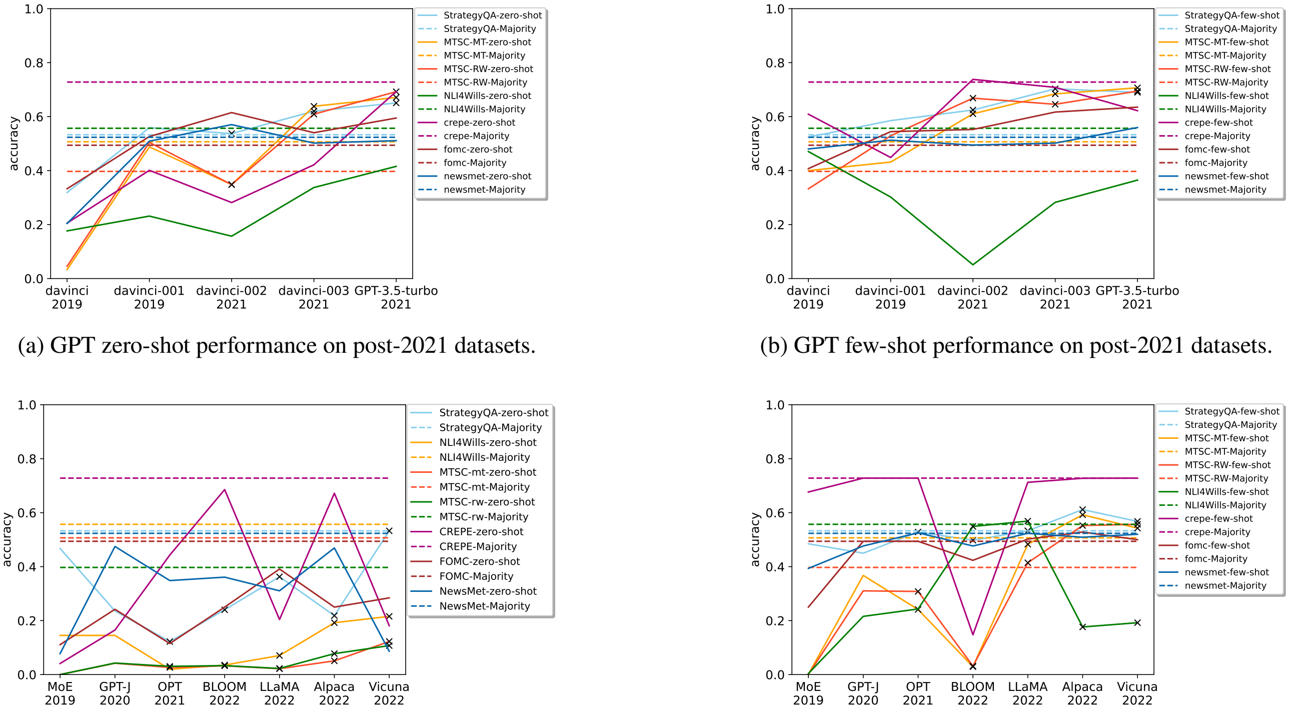
(a) GPT zero-shot performance on pre-2021 datasets.

(b) GPT few-shot performance on pre-2021.

(c) Open LLM zero-shot performance on pre-2021.

(d) Open LLM few-shot performance on pre-2021.

Figure 8: Performance on pre-2021 datasets. In the *x* axis, LLMs are ordered chronologically. Dotted lines are majority baselines.



(a) GPT zero-shot performance on post-2021 datasets.

(b) GPT few-shot performance on post-2021 datasets.

(c) Open LLM zero-shot performance on post-2021.

(d) Open LLM few-shot performance on post-2021.

Figure 9: Performance on post-2021 datasets. In the *x* axis, LLMs are ordered chronologically. Dotted lines are majority baselines. "x" indicates the model may have seen the dataset based on the date: the model training data collection date is after the dataset release date.

G Prompt Examples for Each Task

In this section we give examples of zero-shot prompts for each task.

Task: MRPC
Prompting Inputs:
He said the foodservice pie business doesn 't fit the company 's long-term growth strategy . " The foodservice pie business does not fit our long-term growth strategy . Are the previous two sentences are paraphrased, respond as yes or no?
Expected Outputs:
Yes

Task: BOOLQ
Prompting Inputs:
Ethanol fuel – All biomass goes through at least some of these steps: it needs to be grown, collected, dried, fermented, distilled, and burned. All of these steps require resources and an infrastructure. The total amount of energy input into the process compared to the energy released by burning the resulting ethanol fuel is known as the energy balance (or "energy returned on energy invested"). Figures compiled in a 2007 report by National Geographic Magazine point to modest results for corn ethanol produced in the US: one unit of fossil-fuel energy is required to create 1.3 energy units from the resulting ethanol. The energy balance for sugarcane ethanol produced in Brazil is more favorable, with one unit of fossil-fuel energy required to create 8 from the ethanol. Energy balance estimates are not easily produced, thus numerous such reports have been generated that are contradictory. For instance, a separate survey reports that production of ethanol from sugarcane, which requires a tropical climate to grow productively, returns from 8 to 9 units of energy for each unit expended, as compared to corn, which only returns about 1.34 units of fuel energy for each unit of energy expended. A 2006 University of California Berkeley study, after analyzing six separate studies, concluded that producing ethanol from corn uses much less petroleum than producing gasoline. Does ethanol take more energy make that produces, respond as yes or no?
Expected Outputs:
No

Task: SST
Prompting Inputs:
Classify the sentiment:it 's a charming and often affecting journey .
Expected Outputs:
Positive

Task: QQP
Prompting Inputs:
Why are African-Americans so beautiful? Why are hispanics so beautiful? Are the previous two sentences are paraphrased, respond as yes or no?
Expected Outputs:
No

Task: QNLI

Prompting Inputs:

Entailment: if the context contains the answer to the question, then it is entailment.
Question: What came into force after the new constitution was herald?
Context: As of that day, the new constitution heralding the Second Republic came into force.
Is the context entailment, Yes or No?

Expected Outputs:

Yes

Task: WNLI

Prompting Inputs:

Entailment: if the premise is true, then the hypothesis must be true. Premise: The drain is clogged with hair. It has to be cleaned. Hypothesis: The hair has to be cleaned. Is the hypothesis entailment?

Expected Outputs:

No

Task: RTE

Prompting Inputs:

Entailment: if the premise is true, then the hypothesis must be true. Premise: Dana Reeve, the widow of the actor Christopher Reeve, has died of lung cancer at age 44, according
to the Christopher Reeve Foundation. Hypothesis: Christopher Reeve had an accident.
Is the hypothesis entailment?

Expected Outputs:

No

Task: CB

Prompting Inputs:

Please identify whether the premise entails the hypothesis.
The answer should be exact ’yes’, ’no’ or ’neutral’.
premise: Valence the void-brain, Valence the virtuous valet.
Why couldn’t the figger choose his own portion of titanic anatomy to shaft?
Did he think he was helping?
hypothesis: Valence was helping
answer:

Expected Outputs:

No

Task: COPA

Prompting Inputs:

The man turned on the faucet. What happened as a result? 1. The toilet filled with water. 2. Water flowed from the spout. Which one, 1 or 2?

Expected Outputs:

2

Task: WIC

Prompting Inputs:

An emerging professional class. Apologizing for losing your temper,
even though you were badly provoked, showed real class.
Does the word class have the same word sense, Yes or No?

Expected Outputs:

No

Task: STRATEGYQA

Prompting Inputs:

Q: Will the Albany in Georgia reach a hundred thousand occupants before the one in New York?
A: The answer (Yes or No) is

Expected Outputs:

No

Task: NLI4WILLS

Prompting Inputs:

Law: 32-3-111. Specifically devised or bequeathed property. (a) A specific legatee or devisee has a right to the specifically gifted or devised property in the testator's estate at death or if the property has been disposed of and a contrary intention is not manifest during the testator's lifetime: (1) Any balance of the purchase price, together with any security interest, owing from a purchaser to the testator at death by reason of sale of the property; (2) Any amount of a condemnation award for the taking of the property unpaid at death; (3) Any proceeds unpaid at death on fire or casualty insurance on, or other recovery for injury to, the property; and (4) Property owned by the testator at death and acquired as a result of foreclosure, or obtained in lieu of foreclosure, of the security interest for a specifically devised obligation.
Condition: The testator and his wife didn't divorce until the testator's death, and the testator's wife survived the testator.
Statement: I give, devise and bequeath all my property, real, personal and mixed, of whatever kind and nature and wheresoever situated, to my wife, [Person-2], if she survives me.
Given the law and condition, check the statement for validity (output Support, Refute, or Unrelated).
Answer:

Expected Outputs:

Refute

Task: NEWSMTSC-RW

Prompting Inputs:

Classify the sentiment of the sentence concerning target Mr. Trump as positive, neutral, or negative:
A group of congressional Democrats said Wednesday that they will ask Congress to take the rare step of officially censuring Mr. Trump.

Expected Outputs:

negative

Task: NEWSMTSC-MT

Prompting Inputs:

Classify the sentiment of the sentence concerning target Hillary Clinton's as positive, neutral, or negative:
While White House officials said in the days after Comey's dismissal that it was largely the result of a memo written by Deputy Attorney General Rod J. Rosenstein criticizing the FBI director's handling of the investigation into Hillary Clinton's use of a private email server when she was secretary of state, Trump suggested in the NBC interview that the Russian investigation played a role in his decision.

Expected Outputs:

negative

Task: Spider without schema

Prompting Inputs:

Create a SQL request to how many singers do we have?
SELECT

Expected Outputs:

SELECT count(*) FROM singer

Task: Spider with schema
Prompting Inputs:

### Postgres SQL tables, with their properties: 	
#
# stadium(Stadium_ID, Location, Name, Capacity, Highest, Lowest, Average) 	
# singer(Singer_ID, Name, Country, Song_Name, Song_release_year, Age, Is_male) 	
# concert(concert_ID, concert_Name, Theme, Stadium_ID, Year) 	
# singer_in_concert(concert_ID, Singer_ID) 	
#
### A query to how many singers do we have? 	
SELECT

Expected Outputs:
SELECT count(*) FROM singer

Task: FOMC
Prompting Inputs:
Classify the following sentence from FOMC into ’HAWKISH’, ’DOVISH’, or ’NEUTRAL’ class. Label ’HAWKISH’ if it is corresponding to tightening of the monetary policy, ’DOVISH’ if it is corresponding to easing of the monetary policy, , or ’NEUTRAL’ if the stance is neutral. The sentence: During the past several years, workers across the wage distribution–not just at the upper end–have seen noticeable increases in the inflation-adjusted value of their wages. Label:
Expected Outputs:
Hawkish

Task: CREPE
Prompting Inputs:
Question: Why does a cold cause your voice to get deeper? Comment: Swelling of the vocal folds makes them heavier and that causes them to vibrate at lower (deeper) frequencies. If you look at a guitar or any string instrument you will notice the thicker strings are the lower notes. Does comment have false presuppositions to the question, Yes or No?
Expected Outputs:
No

Task: NewsMet
Prompting Inputs:
Classify the following sentence into ’literal’, or ’metaphorical’ class. Label ’literal’ if it is not metaphorical. Label ’metaphorical’ if it is metaphorical. The sentence: President Donald Trump kicks CNN reporter out of Oval Office Label:
Expected Outputs:
metaphorical

H Prompts for Task Example Extraction

Task	Prompt used
RTE	Generate several training examples for Recognizing Textual Entailment dataset including premise and hypothesis with entailment and not_entailment as labels.
WNLI	Generate several training examples for Winograd Schema Natural Language Inference dataset including premise and hypothesis with entailment and not_entailment as labels.
COPA	Generate several training examples for Choice of Plausible Alternatives (COPA) dataset including premise and choices as input with 0 or 1 as labels.
SST-2	Generate several training examples for sentiment analysis task with positve and negative as labels
MRPC	Generate several training examples for Microsoft Research Paraphrase Corpus task.
QNLI	Generate several training examples for Question answering Natural Language Inference dataset using question answer pairs with entailment and not_entailment as labels.
CB	Generate several training examples for CommitmentBank Natural Language Inference dataset including premise and hypothesis as input with entailment, neutral, as contradiction labels.
WiC	Generate several training examples for The Word-in-Context (WiC) Dataset task including 2 sentences and a word in both sentences as input with true or false as labels.
BoolQ	Generate several training examples for BoolQ dataset which is a question answering dataset for yes/no questions including passage and question as input with yes or no as labels.
StrategyQA	Generate several training examples for StrategyQA task which is a question-answering task focusing on open-domain questions where the required reasoning steps are implicit in the question and should be inferred using a strategy. Generate with a question and reasoning steps as input and Yes or No as Labels.
NewsMTSC	Generate several training examples for Multi-Target-dependent Sentiment Classification in Political News Articles including a sentence and a target in the sentence as input with positive and negative as labels.
NLI4Wills	Generate several training examples for the validity evaluation of the legal will statements including statement, conditions and law as input with support, refute, or unrelated as labels.
CREPE	Generate several training examples for a QA task containing a natural distribution of presupposition failures for questions with whether there is any false presuppositions including question and comment as input with true or false as labels
FOMC	Generate several training examples for Federal Open Market Committee (FOMC) dataset for a measure of monetary policy stance task including sentence from FOMC document as input with Dovish, Hawkish or Neutral as labels.
NewsMet	Generate several training examples from NewsMet, a large high-quality contemporary dataset of news headlines hand-annotated with metaphorical verbs with a task to detect if the headline is metaphorical including a headline sentence as input with 0 or 1 as labels to represent metaphorical or not metaphorical.

Table 10: Prompts used for each task for task example extraction.