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 or 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 datasets released over time, and over LLMs released over time. Utilizing GPT-3 series models and several other recent open-sourced LLMs, and controlling for dataset difficulty, we find that datasets released prior to the LLM training data creation date perform surprisingly better than datasets released post the LLM training data creation date. This strongly indicates that, for many LLMs, there exists task contamination on zero-shot and few-shot evaluation for datasets prior to the LLMs’ training data creation date. Additionally, we utilize training data inspection, training data extraction, and a membership inference attack, which reveal further evidence of task contamination. Importantly, we find that for 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

Simply evaluating the scope of this contamination is difficult to do (Magar and Schwartz 2022; Jacovi et al. 2023).}

1Zero-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.
Closed models do not release their pre-training data. While 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 CSV, 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 2023) 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 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 data extraction to look for possible task contamination. Importantly, we find that for 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) for all models in our analysis, and find a strong correlation (R=.88) between 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 of 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 this task 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 data extraction**: Extract task data 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 data need not exactly match existing training data examples. Any training 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 data 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.

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), task data extraction (§6) and membership inference attack (§7).

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. 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 models like those from the GPT-3 series and more accessible, community-driven open models. More information

---

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
about these models is given in the Appendix of the arXiv version of the paper.

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 their 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, while release dates for each dataset are listed in Table 2.

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.

<table>
<thead>
<tr>
<th>Model</th>
<th>Training data</th>
<th>Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>davinci</td>
<td>Up to Oct 2019</td>
<td>May 2020</td>
</tr>
<tr>
<td>davinci-001</td>
<td>Up to Oct 2019</td>
<td>Jun 2020</td>
</tr>
<tr>
<td>davinci-002</td>
<td>Up to Jun 2021</td>
<td>Jan 2022</td>
</tr>
<tr>
<td>davinci-003</td>
<td>Up to Jun 2021</td>
<td>Nov 2022</td>
</tr>
<tr>
<td>GPT-3.5-T</td>
<td>Up to Sep 2021</td>
<td>Mar 2023</td>
</tr>
</tbody>
</table>

(a) GPT-3 Series LLMs

<table>
<thead>
<tr>
<th>Model</th>
<th>Training data</th>
<th>Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fairseq MoE</td>
<td>Up to Feb 2019</td>
<td>Dec 2021</td>
</tr>
<tr>
<td>GPT-J</td>
<td>Up to 2020</td>
<td>Jun 2021</td>
</tr>
<tr>
<td>OPT</td>
<td>Up to Oct 2021</td>
<td>May 2022</td>
</tr>
<tr>
<td>BLOOM</td>
<td>Prior Aug 2022</td>
<td>Nov 2022</td>
</tr>
<tr>
<td>LLaMA</td>
<td>Up to Aug 2022</td>
<td>Feb 2023</td>
</tr>
<tr>
<td>Alpaca</td>
<td>From davinci-003</td>
<td>Mar 2023</td>
</tr>
<tr>
<td>Vicuna</td>
<td>From ChatGPT</td>
<td>Mar 2023</td>
</tr>
</tbody>
</table>

(b) Open LLMs

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

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Year</th>
<th>Pre-2021</th>
<th>Post-2021</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTE</td>
<td>2009</td>
<td></td>
<td>StrategyQA</td>
</tr>
<tr>
<td>WNL1</td>
<td>2011</td>
<td></td>
<td>NewsMTSC-MT</td>
</tr>
<tr>
<td>COPA</td>
<td>2011</td>
<td></td>
<td>NewsMTSC-RW</td>
</tr>
<tr>
<td>SST-2</td>
<td>2013</td>
<td></td>
<td>NLI4Wills</td>
</tr>
<tr>
<td>MRPC</td>
<td>2015</td>
<td></td>
<td>CREPE</td>
</tr>
<tr>
<td>QNLI</td>
<td>2018</td>
<td></td>
<td>FOMC</td>
</tr>
<tr>
<td>CB</td>
<td>2019</td>
<td></td>
<td>NewsMet</td>
</tr>
<tr>
<td>WiC</td>
<td>2019</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BoolQ</td>
<td>2019</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

4 Chronological Analysis

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

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 datasets) versus after the training data collection date (post-collection datasets). Specifically, we focus on whether the model is above the majority baseline. In this section we use this measure, instead of averaging the performance across datasets, to avoid datasets with large performance differences dominating the analysis.

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 datasets and models, the performance difference above the majority baseline is small, so we also perform the same comparison counting datasets for which the results above the majority baseline are 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 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 both open LLMs and GPT-3 series LLMs, with a stronger indication of contamination in the GPT-3 series with davinci-001 and after.
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. 2022). The remaining datasets all have decent 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 cause contamination. For example, the FOMC dataset (Shah, Paturi, and Chava 2023) was officially released post-collection for the GPT-3 series, but its performance on subsequent versions 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 categories 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 indi-
Importantly, we find that for tasks with no possibility of task contamination, models do not demonstrate statistically significant improvements over majority baselines, in both zero and few-shot settings. The exception is davinci-001 on post-collection datasets, which shows a statistically significant improvement over one post-collection dataset (MTSC-RW, a sentiment classification dataset), but does not generate task examples with our prompt (Table 4).

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). 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. We then compare the performance to see if more task-specific training examples has boosted performance. Because we must check manually, we can perform this analysis only for the small fine-tuning datasets of Alpaca and Vicuna.

Table 3 shows the number of task examples on Alpaca and Vicuna, as well as the change in performance averaged over zero and few-shot settings. We find that performance has improved for Alpaca and Vicuna over the original LLaMA model for tasks with task examples. 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.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Alpaca</th>
<th>Vicuna</th>
</tr>
</thead>
<tbody>
<tr>
<td>SST-2</td>
<td>8. +14.6%</td>
<td>0. -1.0%</td>
</tr>
<tr>
<td>MRPC</td>
<td>0. -0.7%</td>
<td>0. -8.0%</td>
</tr>
<tr>
<td>RTE</td>
<td>0. +3.1%</td>
<td>33. +10.6%</td>
</tr>
<tr>
<td>QNLI</td>
<td>0. -0.4%</td>
<td>28. +10.0%</td>
</tr>
<tr>
<td>WNLI</td>
<td>0. -1.4%</td>
<td>33. +7.7%</td>
</tr>
<tr>
<td>CB</td>
<td>0. +9.8%</td>
<td>0. -23.2%</td>
</tr>
<tr>
<td>COPA</td>
<td>?. 0%</td>
<td>?. +10%</td>
</tr>
<tr>
<td>WiC</td>
<td>0. -4.9%</td>
<td>0. -2.5%</td>
</tr>
<tr>
<td>BoolQ</td>
<td>?. +19%</td>
<td>?. +4.0%</td>
</tr>
</tbody>
</table>

| StrategyQA | 0. -3.3%  | 0. +10.3% |
| NL4Wills   | 0. -13.5% | 0. -11.6% |
| MTSC-RW    | ?. +9.6%  | ?. +11.3% |
| MTSC-MT    | ?. +6.9%  | ?. +8.0%  |
| CREPE      | 0. +24.2% | 0. -0.4%  |
| FOMC       | 0. -5.7%  | 1. -5.4%  |
| NewsMet    | 4. +7.2%  | 0. -11.4% |

Table 3: Contamination analysis for tasks: # of datapoints in the Alpaca and Vicuna datasets that match a regular expression for the task, and Δ%, the change in performance averaged across zero and few-shot settings. "?" means 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 of the arXiv paper. Δ% is the average performance difference over zero-shot and few-shot compared to the original Llama model.

5 Task Data Extraction

We test for task data contamination by attempting to extract task data 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.

Table 4 shows the training data 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 prior, which could have caused contamination. For the open LLMs, almost no models can generate training examples of specific tasks except for Vicuna, which 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.

18476
Figure 5: The number of generated examples which exactly match the original set and the performance (Accuracy).

Figure 6: Exact Match Amount vs. Accuracy for Spider on development set. $R^2 = 0.88$

Table 4: Task example extraction results on tasks, ordered by release year. A line separates pre-training data collection datasets (top) and post-training data collection datasets (bottom) for each LLM. ■ 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.
7 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.

8 Take-Aways

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

- 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 tasks with no possibility of task contamination, models rarely demonstrate 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 diagnosing of contamination issues.

9 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. Maier 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 circumventing testing data contamination. Additional work has further looked into text data contamination (Sainz et al. 2023b; Zhou et al. 2023; Golchin and Surendran 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.

10 Conclusion and Future Work

We investigate task contamination for LLMs, and conduct a chronological analysis, training data inspection, training data 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 tasks without the possibility of task contamination, models 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 the 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


