Position: Language model developers should report train-test overlap

Andy K Zhang [1] **Kevin Klyman** [1] **Yifan Mai** [1] **Yoav Levine** [1] **Yian Zhang** [1] **Rishi Bommasani** [1] **Percy Liang** [1]

Abstract

Language models are extensively evaluated, but correctly interpreting evaluation results requires knowledge of *train-test overlap*, which refers to the extent to which the language model is trained on the very data it is being tested on. The public currently lacks adequate information about train-test overlap: most models have no public train-test overlap statistics, and third parties cannot directly measure train-test overlap since they do not have access to the training data. To make this clear, we document the practices of 30 models, finding that just 9 models report train-test overlap: 4 models release training data under open-source licenses, enabling the community to directly measure train-test overlap, and 5 models publish their train-test overlap methodology and statistics. By engaging with language model developers, we provide novel information about train-test overlap for three additional models. Overall, this position paper argues that language model developers should publish train-test overlap statistics and/or training data whenever they report evaluation results on public test sets. We hope our work increases transparency into train-test overlap to increase the community-wide trust in model evaluations.

1. Introduction

The artificial intelligence (AI) community has built hundreds of evaluations to better understand language models (Srivastava et al., 2023; Hendrycks et al., 2021a; Gao et al., 2024; Myrzakhan et al., 2024; Chiang et al., 2024; Rein et al., 2023; Liang et al., 2023). These evaluations cannot be correctly interpreted without knowledge of *train-test overlap*, which we define as the extent to which the evaluation test data appears in the training data.

Prior to the rise of language models trained on web-scale data, the AI community used standard train/test set splits, where a model would be trained on the training set and tested on the test set to ensure validity of results (Jurafsky & Martin, 2009; Russell & Norvig, 2009). In that regime, the designer of an evaluation generally would specify both the training and test sets. In contrast, today foundation model developers decide on their own training sets, which they often do not release, and evaluation designers decide on test sets, which they often release. Overall, the shift to web-scale training data with poor documentation of data provenance, along with the two-party specification of training and test data, contributes to poor understanding of train-test overlap (Longpre et al., 2023).

Train-test overlap can arise for several reasons. First, since evaluation datasets are often made public on the Internet (e.g. via public repositories like GitHub and Hugging Face), these datasets may be scraped and then trained upon. Second, since evaluation datasets often use already-public material (e.g. the held-out examples in SQuAD still depend on public Wikipedia data (Rajpurkar et al., 2016)), the underlying data may be easily trained upon. Third, since evaluation datasets are often input into models to conduct evaluations (e.g. to evaluate GPT-4 via the OpenAI API), these datasets may be stored and used to train future models. Better understanding how train-test overlap arises may facilitate solutions for appropriately navigating the challenges it presents (Oren et al., 2023).

A growing literature demonstrates high train-test overlap for language models, which contributes to significant degradation in performance between seen and unseen test examples (Lewis et al., 2020; Elangovan et al., 2021; Vu et al., 2023). For example, OpenAI initially reported that GPT-4 had achieved state-of-the-art performance on a test set of Codeforces coding questions, claiming that there was no contamination (OpenAI, 2023). Yet it was later demonstrated[1] that while GPT-4 achieves 100% accuracy for 10 pre-2021 problems, the model achieves 0% accuracy on more recent problems. More recently, Anthropic claimed a "zero to one" moment in solving Capture The Flag tasks (Anthropic, 2025). Yet Transluce found that Claude 3.5

---

[1]Stanford University, Stanford, CA, USA. Correspondence to: Andy, Zhang <andyzh@stanford.edu>.

[1]See `https://twitter.com/cHHillee/status/1635790330854526981?lang=en`

solves broken tasks by memorizing answers and hallucinating them (Meng, 2025). More generally, Kapoor & Narayanan (2023) document that test data often leaks into training data across many domains. Therefore, improving the community's understanding of train-test overlap will increase the validity of, and trust in, evaluations.

Given the value of understanding train-test overlap, we study the practices of 30 language models. We find that 9 models have published sufficient data for the AI community to contextualize train-test overlap: 4 models (OLMo—AI2, GPT-NeoX—EleutherAI, RedPajama INCITE—Together, StarCoder 2—BigCode/HuggingFace/ServiceNow) have released open-source datasets that the community can inspect for train-test overlap, and 5 models (GPT-4—OpenAI, Llama 3.1—Meta, Qwen2—Alibaba, Palmyra—Writer, Apple Intelligence—Apple) have published their methodology and statistics for train-test overlap. The remaining 23 models do report evaluation results on public test sets, but do not (adequately) report train-test overlap results. In parallel to models reporting train-test overlap, the community is building black-box methods to estimate train-test overlap without access to training data (Golchin & Surdeanu, 2023; Shi et al., 2023; Oren et al., 2023), but these approaches are quite limited at present. **We take the position that language model developers should report train-test overlap**.

> Position: Language model developers should report train-test overlap.
>
> Language model developers routinely publish evaluations of their models on public test sets. However, these evaluations are often not accompanied with train-test overlap statistics, making it difficult to assess their validity. Similar to how statisticians are expected to release confidence intervals to ensure validity of their results, we argue that language model developers that publish results on public test sets should release their models' training data and/or publish accompanying train-test overlap statistics so that the AI community can correctly interpret the evaluation results.

2. Alternative Views

The prevalence of train-test overlap as an issue in the AI community[2] has led to the development of various strategies to estimate and address train-test overlap, including

---

[2]Potential evidence of train-test overlap is often flagged by members of the AI community on social media. See, e.g., `https://twitter.com/dhuynh95/status/1775568278557192411`

black-box methods, private test sets, novel test sets, and canary strings. We cover each of these in turn then discuss our approach.

Black-box methods involve researchers working to estimate train-test overlap through model API access and the test set rather than directly through access to the training set. Notably, there have been efforts to estimate train-test overlap via prompting, word probabilities, and test example orderings (Golchin & Surdeanu, 2023; Shi et al., 2023; Oren et al., 2023). Golchin & Surdeanu (2023) prompt the model with the dataset name, partition type, and an initial segment of the reference string, and mark train-test overlap if the model responds with an exact or similar copy in the output. Shi et al. (2023) estimate train-test overlap via the probability outputs of outlier words, with the hypothesis that unseen examples is likely to contain few outliers with low probabilities. Oren et al. (2023) estimate train-test overlap by considering the ordering of test instances, noting that language models with train-test overlap are likely to memorize such ordering. These methods can be helpful for estimation and as a sanity check to white-box approaches, but currently have limitations as they are not robust to adversarial settings such as if a developer fine-tuned its model to avoid revealing training data and even in the benign setting, require certain assumptions such as requiring a certain threshold of frequencies for detection or certain methods of training (Casper et al., 2024; Golchin & Surdeanu, 2023; Shi et al., 2023; Oren et al., 2023). Estimating and interpreting train-test overlap is difficult even in the white-box setting with direct access to the training data as current approaches have significant limitations; with further constraints in the black-box setting, the challenges only increase.

Private test sets such as SQuAD (Rajpurkar et al., 2018) and SEAL (Scale, 2024) allow researchers to keep a portion or all of the test set hidden, meaning that the test set is not publicly accessible on the internet and developers are therefore much less likely to train models on it. While private test sets can be valuable, they raise potential concerns regarding data transparency. For instance, unless the private test set is shared with a trustworthy third party, the community must rely upon a single organization's assessment of the test set's validity. In any event, public test sets are the industry standard and will continue to exist, though private and public test sets can coexist in a healthy testing ecosystem.

Novel test sets that include data that was produced after the knowledge cutoff date of a model also help mitigate train-test overlap. Including recent data is a best practice for new test sets, though this may be difficult if, for example, a new test set is derived from existing data (e.g. based

on old Wikipedia data or AI-generated data). Even when this approach is implemented successfully, new models are released regularly that are trained on more recent data, necessitating some analysis of train-test overlap with the previously novel test set. One modification of this approach is to add novel data to the test set at regular intervals, as with Livebench (White et al., 2024) or Image2Struct(Roberts et al., 2024). In addition to the financial cost of continually adding novel data, which may not be feasible for every domain or project, one challenge of this approach is that it is difficult to interpret longitudinal progress.

Canary strings as introduced by BIG-bench (Luo et al., 2024), are another strategy to cope with train-test overlap. Here, tasks in a test set are marked by a unique string, called a canary string, allowing developers to filter out data that contain canary strings during training. If a model outputs a given canary string, it signals that there is likely train-test overlap with the associated test set. But canary strings are not implemented uniformly or consistently—tasks exist without canary strings, whether within the test set or in other instances across the internet, and canary strings can be easily filtered out of test sets. More often test sets are derived from other raw sources that do not contain the canary string. It is also possible that canary strings may be referenced independently of the tasks in test sets, producing potential false positives.

Our position: To complement these above approaches, language model developers should report train-test overlap statistics or openly release their training data. A developer chooses the specific test sets it uses to evaluate its language model, and it can choose to report train-test overlap for those test sets (e.g. through a transparency report or a model card) using its preferred method for computing train-test overlap (Bommasani et al., 2024b). This is similar to norms in the field of statistics, where published results must be accompanied by confidence intervals, rather than arbitrary reporting criteria imposed by a third party. This approach would complement existing strategies: for instance, black-box methods and canary strings are powerful tools to sanity check train-test overlap statistics that a developer reports. Similarly, private or novel test sets can further sanity check results on existing public test sets, such as drawing attention to cases of significant divergence.

3. Consequences of Not Reporting Train-Test Overlap

Failing to disclose train-test overlap can degrade trust within both the AI community and the broader public, ultimately diminishing the value of evaluations that are pivotal for research progress and real-world applications. The current landscape of AI evaluation is increasingly charac-

terized by accusations of "cheating" or undisclosed advantages, fostering a climate of low trust within the community . This lack of transparency not only undermines the credibility of individual model evaluations but also has broader negative consequences for the advancement and trustworthiness of AI as a whole.

One significant consequence of not reporting train-test overlap is that it leads to unsubstantiated claims by model developers, and consequently an erosion of trust. For example, OpenAI initially reported that GPT-4 had achieved state-of-the-art performance on a test set of Codeforces coding questions, claiming that there was no contamination (OpenAI, 2023). Yet it was later demonstrated[3] that while GPT-4 achieves 100% accuracy for 10 pre-2021 problems, the model achieves 0% accuracy on more recent problems. More recently, Anthropic claimed a "zero to one" moment in solving Capture The Flag tasks (Anthropic, 2025). Yet Transluce found that Claude 3.5 solves broken tasks by memorizing answers and hallucinating them (Meng, 2025). Instead, if the model developers had provided train-test overlap statistics associated with these results, it would have been clear that train-test overlap was a significant contributor of these breakthrough results. Accordingly, it increases accountability of model developer, which then increases the trust in their claims.

Perhaps even more significant than these instances with clear evidence are accusations of cheating that are hard to verify, which end up eroding trust further. For instance, there are "cheating" accusations against o3 on the ARC-AGI benchmark (Lee, 2024), which OpenAI has denied. While we can attempt to derive evidence via black-box methods (Golchin & Surdeanu, 2023; Shi et al., 2023; Oren et al., 2023), they have limitations and cannot fully substitute for transparent reporting from model developers. The inherent uncertainty associated with black-box methods, coupled with the potential for adversarial manipulation, means that accusations and counter-accusations can persist, leading to a fragmented and less productive research environment.

Accordingly, the failure to report train-test overlap has significant negative consequences for the AI community. It fosters a climate of distrust, hinders scientific progress by obscuring the true nature of model capabilities, and makes it difficult to build upon existing research. By embracing transparency and routinely reporting train-test overlap statistics, the AI community can move towards a more trustworthy and collaborative environment that ultimately accelerates the development of safer and more trustworthy AI.

---

[3]See `https://twitter.com/cHHillee/status/1635790330854526981?lang=en`

4. Language Models

To establish a broad understanding of the landscape, we comprehensively consider the train-test overlap practices of the flagship language model of 30 developers (01.ai, Adept, AI2, AI21 Labs, Aleph Alpha, Alibaba, Amazon, Anthropic, Apple, BigCode, Cohere, Databricks, DeepSeek, EleutherAI, Technology Innovation Institute, Google, IBM, Imbue, Inflection, Meta, Microsoft, Mistral, NVIDIA, OpenAI, Reka AI, Snowflake, Stability AI, Together AI, Writer, and xAI). We assembled this list by considering all models on the HELM MMLU leaderboard[4] and additional models that we selected for impact and relevance based on Ecosystem Graphs (Bommasani et al., 2023b).

Next, we selected the latest flagship model for which the developer had published benchmarks results. This is because we emphasize that a developer should disclose information about train-test overlap on the subset of benchmarks that the developer publishes rather than a pre-defined list of benchmarks decided by another party. For some developers, this meant selecting an older flagship model as there are as of yet no published results on the newer model. We chose to exclude developers that have not published results on public language benchmarks such as Baidu. We consider only models with results released before September 1, 2024 (the date we provided as a deadline to model developers to share additional train-test overlap information) and accordingly reported on Qwen2 rather than Qwen2.5, GPT-4 rather than GPT-4o [5], and OLMo rather than OLMoE.

5. Results

5.1. Documenting Current Practices

We followed a standardized procedure in order to document current practices regarding reporting of train-test overlap statistics. For each developer-model pair, we followed the following process to collate the developer's current practices with respect to reporting train-test overlap:

1. We identified papers, technical reports, and company websites that were potential sources of information on train-test overlap.

2. We queried and identified any data the developer has published regarding the model's results on public benchmarks. We documented each public benchmark on which the developer reports results for the model.

3. We queried each document that includes results on public benchmarks for information on train-test overlap. In addition to reading the document, we queried for terms including "contamination", "overlap", and "gram", then manually inspected the occurrence to determine whether any train-test overlap data was released.

5.2. Scoring Criteria

We assign each developer a binary score of 1 or 0 to indicate whether the developer has provided sufficient information to contextualize train-test overlap for its flagship model. In this work we do not evaluate the specific methodology that each developer choses to employ to estimate train-test overlap, as these methodologies are inconsistent, opaque, and often not comprehensive. Instead, we identify whether a developer meets some minimum threshold with respect to publicly reporting some meaningful information about train-test overlap.

In assigning scores to developers, we consider the following criteria:

1. Is the training data publicly available?

2. Is train-test overlap reported on public benchmarks for which the model's results are reported? That is, for each test set, we want a number that measures overlap. Note that this can be an implicit 0 for those who prefilter their training data.

   (a) Is train-test overlap reported with sufficient specificity to be meaningful?

   (b) Is there a clear description of the method the developer used to compute train-test overlap?

If none of these criteria are met, then the developer scores 0. If the training data is publicly available, the developer scores 1 as third parties can directly compute train-test overlap statistics for any public test set of interest. If the training data is not publicly available, but train-test overlap is reported with sufficient specificity and a clear description of the method, the developer scores 1.

For each developer that scored 0, we reached out to the developer to provide them an opportunity to engage with or rebut the score. Each of these developers was given the opportunity to point to any relevant information that our analysis was missing, or provide additional information that would be publicized.

5.3. Scores

Here we document the train-test overlap practices of 30 models with published results on public test sets. Of these,

---

Model	Developer	Score	Explanation
Pythia	EleutherAI	1	Open training data (Biderman et al., 2023)
OLMo	AI2	1	Open training data (Groeneveld et al., 2024)
RedPajama-INCITE 7B	Together AI	1	Open training data (Computer, 2024)
StarCoder 2	BigCode	1	Open training data (Lozhkov et al., 2024)
Palmyra X V3	Writer	1	Published analysis and code (Writer, 2024)
GPT-4	OpenAI	1	Published analysis (OpenAI et al., 2024)
Llama 3.1	Meta	1	Published analysis (Dubey et al., 2024)
Qwen2	Alibaba	1	Published analysis (Yang et al., 2024)
Apple Intelligence	Apple	1	Published prefiltering (Gunter et al., 2024)
Gemini 1.5 Pro	Google	0	Insufficient methodological details (Team et al., 2024a)
Arctic	Snowflake	0	No analysis (Snowflake, 2024)
Claude 3.5 Sonnet	Anthropic	0	No analysis (Anthropic, 2024)
Command R	Cohere	0	No analysis (Cohere, 2024)
Core	Reka AI	0	No analysis (Team et al., 2024b)
DBRX	Databricks	0	No analysis (Databricks, 2024)
DeepSeek	DeepSeek	0	No analysis (DeepSeek-AI et al., 2024)
Falcon	TII	0	No analysis (Almazrouei et al., 2023)
Fuyu-Heavy	Adept	0	No analysis (Adept, 2024)
Granite	IBM	0	No analysis (Mishra et al., 2024)
Grok-2	xAI	0	No analysis (x.ai, 2024)
Imbue 70B	Imbue	0	No analysis (Imbue, 2024)
Inflection-2.5	Inflection	0	No analysis (AI, 2024a)
Jamba-1.5	AI21 Labs	0	No analysis (AI21, 2024)
Luminous Supreme	Aleph Alpha	0	No analysis (Alpha, 2024)
Mixtral Large 2	Mistral	0	No analysis (AI, 2024b)
Nemotron-4-340B-Instruct	NVIDIA	0	No analysis (NVIDIA, 2024)
Phi 3	Microsoft	0	No analysis (Abdin et al., 2024)
Stable LM 2	Stability AI	0	No analysis (AI, 2024c)
Titan Text Express	Amazon	0	No analysis (Amazon, 2024)
Yi-34B	01.ai	0	No analysis (AI et al., 2024)

Table 1: Models and scores. This table displays the score of 30 developers on our metric for train-test overlap transparency—a developer scores 1 if it releases sufficient information to contextualize for its flagship language model, and 0 otherwise. From left to right, the table includes: a list of flagship language models, a list of major model developers, the model developers' scores, an abbreviated explanation for why the developer received that score.

9 models have published sufficient data for the community to contextualize train-test overlap: 4 models (OLMo—AI2, GPT-NeoX—EleutherAI, RedPajama INCITE—Together, StarCoder 2—BigCode/HuggingFace/ServiceNow) models have released training data under open-source licenses, which researchers can inspect for train-test overlap and 5 models (GPT-4—OpenAI, Llama 3.1—Meta, Qwen2—Alibaba, Palmyra—Writer, Apple Intelligence—Apple) have published their methodology and statistics for train-test overlap. For model developers that do not openly release their training data, see Appendix A for additional explanation below as to why their transparency regarding train-test overlap is meaningful.

6. Discussion

Overall, while train-test overlap is a fundamental to interpreting evaluation results, there is still significant limitations in the measurement methodology, beyond data access challenges and developer responsibility. As described above, direct string comparison is the most common way to quantify train-test overlap. This method has slight variations, but typically involves detecting substring matches between training and test data. N-gram matching is commonly employed (Yang et al., 2024; Dubey et al., 2024; Brown et al., 2020b; Chowdhery et al., 2022a), where documents are tokenized and then compared, though OpenAI compares characters rather than tokens in its analysis for GPT-4 (OpenAI, 2023). There are important design decisions developers make in employing n-gram strategies,

such as what to set as N, whether to allow fuzzy matching or skipgrams (Dubey et al., 2024), and whether to filter based only a threshold of matches. This lack of uniformity in measurement approaches can make it challenging to directly compare train-test overlap analyses.

There are a number of limitations to this class of approaches. One limitation is that n-grams are coarse and do not capture the differences in types of overlap between different test sets. For instance, CivilComments is derived from news sites, so overlap between training data and CivilComments is likely due to news articles that appear in both the training and test data (Duchene et al., 2023). In contrast, MMLU (Hendrycks et al., 2021a) and MATH (Hendrycks et al., 2021b) are in question-answer format, so overlap could stem from leakage of questions or answers or repetition of common phrases in questions and answers. We categorize the overlap types for different scenarios for The Pile (Gao et al., 2020) in Table 2. Overlap types can be broadly categorized into question leakage; quotes from news, laws, books, and songs; common phrases; and multi-token identifiers. Question leakage is the canonical concern for training data: if a model trains on the questions in the test set, it can achieve high results that fail to generalize well to new questions. However, these other overlap types can also be informative; for instance, simply matching the LSAT question stem "Which one of the following could be a complete and accurate list of the" suggests that the training data likely contains LSAT questions or similar questions. Indeed, train-test overlap is a construct that captures the relation between train and test data, and the different type of overlap add complexity that make it difficult to capture in a single statistic. Future work could explore these various overlap types in more detail and devise more granular metrics of measurement.

Another limitation is that n-gram analysis fails to catch many classes of train-test overlap that may be relevant, such as translations, summaries, or paraphrases of the text (Lee et al., 2024). Yang et al. (2023) demonstrate that there can be significant train-test overlap even with OpenAI's prefiltering methods. Prior work has made progress on addressing this limitation, including by making use of embeddings or an LM-evaluator for a more semantic-based match (Dong et al., 2024; Jiang et al., 2024). These gaps demonstrate the need for further work on developing improved methods for estimating train-test overlap.

Indeed, in light of these limitations, we note that we are fundamentally interested in measuring the amount of generalization, rather than direct string matches or any specific approaches. This could extend to train-test overlap at the task or domain level. Additionally, it highlights that unlike the common perception, train-test overlap is not necessarily a negative (in part why we choose this term as opposed

to "contamination"). Instead, it is helpful to guide understanding and help contextualize results.

Additionally, there are complexities in determining what qualifies as the training set, as there are often multiple stages of training and datasets, including pretraining, fine-tuning, and safety alignment among others (Yang et al., 2024; Dubey et al., 2024; Brown et al., 2020b; Chowdhery et al., 2022a; OpenAI, 2023). This is often not captured in developers' public reports (Bommasani et al., 2024a), and it may not be well captured internally either. Precision about the training set is important, though it is beyond the scope of this paper.

This paper does not assert a position on which method is best, and acknowledges that there is substantial research remaining to investigate better methods of computing train-test overlap. Nevertheless, the limitations of black-box approaches are far greater than those of white-box approaches. Just as the Foundation Model Transparency Index has helped improve the transparency of foundation model developers (Bommasani et al., 2023a; 2024a), our hope is that an increase in the number of model developers that report train-test overlap will produce better methods of measurement and help standardize reporting such that developers' transparency on train-test overlap improves.

7. Future Work: Standardization of Train-Test Overlap

The first critical step toward addressing train-test overlap is establishing a baseline expectation for model developers to publish overlap statistics for the benchmarks they themselves report, which we have advocated for here. We have seen promising adoption of this by nine model developers, including three new model developers from outreach. As this expectation becomes increasingly a reality, we have substantial additional work to further increase train-test overlap transparency.

In particular, the model developers employ different methodology for computing train-test overlap statistics, which make it challenging to compare statistics between models. In fact, train-test overlap statistics can differ even between models for a given model developer, without clear reasoning. As such, we believe that standardization of the protocol to compute train-test overlap will be an essential next step to improving transparency and trust in the space.

Here, we document an initial effort where we advocated for a standardized protocol requiring developers to compute overlap using shared benchmarks, tools, and scripts to ensure comparability (see Appendix B-D for more details). While this approach garnered interest among a few model developers including Writer, Meta, and Cohere, widespread adoption proved challenging due to a lack of

Benchmark	Overlap Type	Example
APPS	Phrase (Question)	Input—— The first line of the input contains a single integer n (1 ≤ n ≤ 100 000)
	Phrase (Sequence)	['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z']
CivilComments	Quote (News)	Crews removed Baltimore's Confederate statues early Wednesday, days after the deadly unrest in Charlottesville
EntityDataImputation	Multi-Token Identifier	AF-S DX Nikkor 18-55mm f/3.5-5.6G VR Lens
HumanEval	Phrase (Math)	prime number, or a prime, is a natural number greater than 1 that
	Phrase (Sequence)	'zero': 0, 'one': 1, 'two': 2, 'three': 3, 'four': 4, 'five': 5, 'six': 6, 'seven': 7, 'eight': 8, 'nine': 9
ICE	Phrase (Legal)	with such intention, and under such circumstances that if by that act, you had caused the death of
LSAT	Phrase (Question)	Which one of the following could be a complete and accurate list of the
MATH	Question Leakage	2-4+6-8+10-12+143-6+9-12+15-18+21
	Phrase (Question)	where B is the area of the base and h is the height
MMLU	Question Leakage	If K is a normal subgroup of H and H is a normal subgroup of G, then K is a normal subgroup of G.
NarrativeQA	Quote (Book)	M. Emanuel was away three years. Reader, they were the three happiest years of my life
RAFT	Quote (News)	the French language is very difficult but we try hard. If we come every day, maybe
RealToxicityPrompts	Quote (News)	When it comes to dashing the hopes of thousands of college-bound African
ThePile	Phrase (Speech)	If you have any questions, please feel free to give me a call.
	Multi-Token Identifier	the U.S. District Court for the Northern District of New York
	Phrase (Sequence)	Mime-Version: 1.0 Content-type: text/plain; charset=us-ascii Content-Disposition: inline
TwitterAAE	Quote (Song)	to be this way. Where did we go wrong we both made mistakes we gotta carry on
Summarization	Quote (News)	There are still many questions that the families of the 96 have and we believe that these people may be able to provide answers to some of those questions
Wikifact	Quote (Wikipedia)	swimming at the 1896 Summer Olympics – men's sailors 100 metre freestyle

Table 2: **Overlap Types and Examples on EleutherAI's The Pile.** For various test sets with overlap on The Pile, we chose contiguous overlapping n-gram sequences to illustrate the different types of overlap. Quotes refer to n-grams that exist in news, law, books, songs, or Wikipedia; Phrases refer to sequence of tokens that seem commonly grouped together; Multi-Token Identifier refers to a logical entity that is split into multiple tokens; and Question Leakage refers to an n-gram that may indicate that the core component of a question is overlapping. This was computed via `https://github.com/stanford-crfm/data-overlap`.

incentives to promote transparency.

Our hope is that the community can learn from our efforts in standardization to better improve transparency in this space. Our view is that as community awareness of and discussion of train-test overlap becomes increasingly prevalent, model developers will have stronger incentives to release train-test overlap statistics and work toward standardization. With our efforts, we have taken initial steps in this direction, but there is still a significant amount of work to be done.

8. Conclusion

In this work, we highlight the need to improve transparency of train-test overlap. Our position is that any language model developer that publishes results on public test sets should release its training data and/or publish accompanying train-test overlap so that the community can interpret the results. We discuss various strategies to address train-test overlap, and how our position complements these efforts. We document the train-test overlap practices of 30 models with published results on public test sets. Of these, 9 models have published sufficient data for the community to contextualize train-test overlap. Finally, we discuss limitations with current approaches to quantifying train-test overlap, while emphasizing that current methods still have value. Instead, we suggest that as the AI community increasingly becomes aware of train-test overlap we can continue to improve upon and align on methodology for measuring and reducing train-test overlap.

Impact Statement

Transparency of train-test overlap is essential for community-wide trust in model evaluation. In this work, we seek to increase transparency by urging language model developers to publish train-test overlap statistics and/or training data whenever they report evaluation results on public test sets.

References

Abdin, M., Aneja, J., Awadalla, H., Awadallah, A., Awan, A. A., Bach, N., Bahree, A., Bakhtiari, A., Bao, J., Behl, H., Benhaim, A., Bilenko, M., Bjorck, J., Bubeck, S., Cai, M., Cai, Q., Chaudhary, V., Chen, D., Chen, D., Chen, W., Chen, Y.-C., Chen, Y.-L., Cheng, H., Chopra, P., Dai, X., Dixon, M., Eldan, R., Fragoso, V., Gao, J., Gao, M., Gao, M., Garg, A., Giorno, A. D., Goswami, A., Gunasekar, S., Haider, E., Hao, J., Hewett, R. J., Hu, W., Huynh, J., Iter, D., Jacobs, S. A., Javaheripi, M., Jin, X., Karampatziakis, N., Kauffmann, P., Khademi, M., Kim, D., Kim, Y. J., Kurilenko, L., Lee, J. R., Lee, Y. T., Li, Y., Li, Y., Liang, C., Liden, L., Lin, X., Lin, Z., Liu, C., Liu, L., Liu, M., Liu, W., Liu, X., Luo, C., Madan, P., Mahmoudzadeh, A., Majercak, D., Mazzola, M., Mendes, C. C. T., Mitra, A., Modi, H., Nguyen, A., Norick, B., Patra, B., Perez-Becker, D., Portet, T., Pryzant, R., Qin, H., Radmilac, M., Ren, L., de Rosa, G., Rosset, C., Roy, S., Ruwase, O., Saarikivi, O., Saied, A., Salim, A., Santacroce, M., Shah, S., Shang, N., Sharma, H., Shen, Y., Shukla, S., Song, X., Tanaka, M., Tupini, A., Vaddamanu, P., Wang, C., Wang, G., Wang, L., Wang, S., Wang, X., Wang, Y., Ward, R., Wen, W., Witte, P., Wu, H., Wu, X., Wyatt, M., Xiao, B., Xu, C., Xu, J., Xu, W., Xue, J., Yadav, S., Yang, F., Yang, J., Yang, Y., Yang, Z., Yu, D., Yuan, L., Zhang, C., Zhang, C., Zhang, J., Zhang, L. L., Zhang, Y., Zhang, Y., Zhang, Y., and Zhou, X. Phi-3 technical report: A highly capable language model locally on your phone, 2024. URL https://arxiv.org/abs/2404.14219.

Adept. Adept fuyu heavy, 2024. URL https://www.adept.ai/blog/adept-fuyu-heavy.

AI, ., :, Young, A., Chen, B., Li, C., Huang, C., Zhang, G., Zhang, G., Li, H., Zhu, J., Chen, J., Chang, J., Yu, K., Liu, P., Liu, Q., Yue, S., Yang, S., Yang, S., Yu, T., Xie, W., Huang, W., Hu, X., Ren, X., Niu, X., Nie, P., Xu, Y., Liu, Y., Wang, Y., Cai, Y., Gu, Z., Liu, Z., and Dai, Z. Yi: Open foundation models by 01.ai, 2024. URL https://arxiv.org/abs/2403.04652.

AI, I. Inflection 2.5 announcement. https://inflection.ai/blog/inflection-2-5, 2024a.

AI, M. Mistral large model announcement. https://mistral.ai/news/mistral-large-2407/, 2024b.

AI, S. Introducing stable lm 2. https://stability.ai/news/introducing-stable-lm-2, 2024c.

AI21. Jamba-1.5 models, 2024. URL https://docs.ai21.com/docs/jamba-15-models.

Almazrouei, E., Alobeidli, H., Alshamsi, A., Cappelli, A., Cojocaru, R., Debbah, M., Étienne Goffinet, Hesslow, D., Launay, J., Malartic, Q., Mazzotta, D., Noune, B., Pannier, B., and Penedo, G. The falcon series of open language models, 2023. URL https://arxiv.org/abs/2311.16867.

Alpha, A. Luminous performance benchmarks, 2024. URL https://aleph-alpha.com/luminous-performance-benchmarks/.

Amazon. Aws ai service cards – amazon titan text lite and titan text express, 2024. URL https:

//aws.amazon.com/machine-learning/
responsible-machine-learning/
titan-text/.

Anthropic. Claude 3.5 sonnet, 2024. URL https://www.anthropic.com/news/claude-3-5-sonnet.

Anthropic. Progress from our frontier red team, March 2025. URL https://www.anthropic.com/news/strategic-warning-for-ai-risk-progress-and-insights-from-our-frontier-red-team. https://www.anthropic.com/news/strategic-warning-for-ai-risk-progress-and-insights-from-our-frontier-red-team.

Biderman, S., Schoelkopf, H., Anthony, Q., Bradley, H., O'Brien, K., Hallahan, E., Khan, M. A., Purohit, S., Prashanth, U. S., Raff, E., Skowron, A., Sutawika, L., and van der Wal, O. Pythia: A suite for analyzing large language models across training and scaling, 2023. URL https://arxiv.org/abs/2304.01373.

Bommasani, R., Klyman, K., Longpre, S., Kapoor, S., Maslej, N., Xiong, B., Zhang, D., and Liang, P. The foundation model transparency index, 2023a. URL https://arxiv.org/abs/2310.12941.

Bommasani, R., Soylu, D., Liao, T., Creel, K. A., and Liang, P. Ecosystem graphs: The social footprint of foundation models. *arXiv*, 2023b.

Bommasani, R., Klyman, K., Kapoor, S., Longpre, S., Xiong, B., Maslej, N., and Liang, P. The foundation model transparency index v1.1: May 2024, 2024a. URL https://arxiv.org/abs/2407.12929.

Bommasani, R., Klyman, K., Longpre, S., Xiong, B., Kapoor, S., Maslej, N., Narayanan, A., and Liang, P. Foundation model transparency reports, 2024b. URL https://arxiv.org/abs/2402.16268.

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. Language models are few-shot learners, 2020a.

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.

Language models are few-shot learners, 2020b. URL https://arxiv.org/abs/2005.14165.

Casper, S., Ezell, C., Siegmann, C., Kolt, N., Curtis, T. L., Bucknall, B., Haupt, A., Wei, K., Scheurer, J., Hobbhahn, M., et al. Black-box access is insufficient for rigorous ai audits. In *The 2024 ACM Conference on Fairness, Accountability, and Transparency*, pp. 2254–2272, 2024.

Chiang, W.-L., Zheng, L., Sheng, Y., Angelopoulos, A. N., Li, T., Li, D., Zhang, H., Zhu, B., Jordan, M., Gonzalez, J. E., and Stoica, I. Chatbot arena: An open platform for evaluating llms by human preference, 2024. URL https://arxiv.org/abs/2403.04132.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., Schuh, P., Shi, K., Tsvyashchenko, S., Maynez, J., Rao, A., Barnes, P., Tay, Y., Shazeer, N., Prabhakaran, V., Reif, E., Du, N., Hutchinson, B., Pope, R., Bradbury, J., Austin, J., Isard, M., Gur-Ari, G., Yin, P., Duke, T., Levskaya, A., Ghemawat, S., Dev, S., Michalewski, H., Garcia, X., Misra, V., Robinson, K., Fedus, L., Zhou, D., Ippolito, D., Luan, D., Lim, H., Zoph, B., Spiridonov, A., Sepassi, R., Dohan, D., Agrawal, S., Omernick, M., Dai, A. M., Pillai, T. S., Pellat, M., Lewkowycz, A., Moreira, E., Child, R., Polozov, O., Lee, K., Zhou, Z., Wang, X., Saeta, B., Diaz, M., Firat, O., Catasta, M., Wei, J., Meier-Hellstern, K., Eck, D., Dean, J., Petrov, S., and Fiedel, N. Palm: Scaling language modeling with pathways, 2022a. URL https://arxiv.org/abs/2204.02311.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., Schuh, P., Shi, K., Tsvyashchenko, S., Maynez, J., Rao, A., Barnes, P., Tay, Y., Shazeer, N. M., Prabhakaran, V., Reif, E., Du, N., Hutchinson, B., Pope, R., Bradbury, J., Austin, J., Isard, M., Gur-Ari, G., Yin, P., Duke, T., Levskaya, A., Ghemawat, S., Dev, S., Michalewski, H., García, X., Misra, V., Robinson, K., Fedus, L., Zhou, D., Ippolito, D., Luan, D., Lim, H., Zoph, B., Spiridonov, A., Sepassi, R., Dohan, D., Agrawal, S., Omernick, M., Dai, A. M., Pillai, T. S., Pellat, M., Lewkowycz, A., Moreira, E., Child, R., Polozov, O., Lee, K., Zhou, Z., Wang, X., Saeta, B., Diaz, M., Firat, O., Catasta, M., Wei, J., Meier-Hellstern, K., Eck, D., Dean, J., Petrov, S., and Fiedel, N. PaLM: Scaling language modeling with pathways. *arXiv*, 2022b.

Cohere. Introducing command-r. https://cohere.com/blog/command-r, 2024.

Computer. Redpajama dataset. https://github.com/togethercomputer/RedPajama-Data, 2024.

Databricks. Introducing dbrx: New state-of-the-art open llm. https://www.databricks.com/blog/introducing-dbrx-new-state-art-open-llm, 2024.

DeepSeek-AI, :, Bi, X., Chen, D., Chen, G., Chen, S., Dai, D., Deng, C., Ding, H., Dong, K., Du, Q., Fu, Z., Gao, H., Gao, K., Gao, W., Ge, R., Guan, K., Guo, D., Guo, J., Hao, G., Hao, Z., He, Y., Hu, W., Huang, P., Li, E., Li, G., Li, J., Li, Y., Li, Y. K., Liang, W., Lin, F., Liu, A. X., Liu, B., Liu, W., Liu, X., Liu, X., Liu, Y., Lu, H., Lu, S., Luo, F., Ma, S., Nie, X., Pei, T., Piao, Y., Qiu, J., Qu, H., Ren, T., Ren, Z., Ruan, C., Sha, Z., Shao, Z., Song, J., Su, X., Sun, J., Sun, Y., Tang, M., Wang, B., Wang, P., Wang, S., Wang, Y., Wang, Y., Wu, T., Wu, Y., Xie, X., Xie, Z., Xie, Z., Xiong, Y., Xu, H., Xu, R. X., Xu, Y., Yang, D., You, Y., Yu, S., Yu, X., Zhang, B., Zhang, H., Zhang, L., Zhang, L., Zhang, M., Zhang, M., Zhang, W., Zhang, Y., Zhao, C., Zhao, Y., Zhou, S., Zhou, S., Zhu, Q., and Zou, Y. Deepseek llm: Scaling open-source language models with longtermism, 2024. URL https://arxiv.org/abs/2401.02954.

Dong, Y., Jiang, X., Liu, H., Jin, Z., Gu, B., Yang, M., and Li, G. Generalization or memorization: Data contamination and trustworthy evaluation for large language models, 2024. URL https://arxiv.org/abs/2402.15938.

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A., Yang, A., Mitra, A., Sravankumar, A., Korenev, A., Hinsvark, A., Rao, A., Zhang, A., Rodriguez, A., Gregerson, A., Spataru, A., Roziere, B., Biron, B., Tang, B., Chern, B., Caucheteux, C., Nayak, C., Bi, C., Marra, C., McConnell, C., Keller, C., Touret, C., Wu, C., Wong, C., Ferrer, C. C., Niko-laidis, C., Allonsius, D., Song, D., Pintz, D., Livshits, D., Esiobu, D., Choudhary, D., Mahajan, D., Garcia-Olano, D., Perino, D., Hupkes, D., Lakomkin, E., AlBadawy, E., Lobanova, E., Dinan, E., Smith, E. M., Radenovic, F., Zhang, F., Synnaeve, G., Lee, G., Anderson, G. L., Nail, G., Mialon, G., Pang, G., Cucurell, G., Nguyen, H., Korevaar, H., Xu, H., Touvron, H., Zarov, I., Ibarra, I. A., Kloumann, I., Misra, I., Evtimov, I., Copet, J., Lee, J., Geffert, J., Vranes, J., Park, J., Mahadeokar, J., Shah, J., van der Linde, J., Billock, J., Hong, J., Lee, J., Fu, J., Chi, J., Huang, J., Liu, J., Wang, J., Yu, J., Bitton, J., Spisak, J., Park, J., Rocca, J., Johnstun, J., Saxe, J., Jia, J., Alwala, K. V., Upasani, K., Plawiak, K., Li, K., Heafield, K., Stone, K., El-Arini, K., Iyer, K., Malik, K., Chiu, K., Bhalla, K., Rantala-Yeary, L., van der Maaten, L., Chen, L., Tan, L., Jenkins, L., Martin, L., Madaan, L., Malo, L., Blecher, L., Landzaat, L., de Oliveira, L., Muzzi, M., Pasupuleti, M., Singh, M., Paluri, M., Kar-das, M., Oldham, M., Rita, M., Pavlova, M., Kambadur, M., Lewis, M., Si, M., Singh, M. K., Hassan, M., Goyal, N., Torabi, N., Bashlykov, N., Bogoychev, N., Chatterji, N., Duchenne, O., Çelebi, O., Alrassy, P., Zhang, P., Li, P., Vasic, P., Weng, P., Bhargava, P., Dubal, P., Krishnan, P., Koura, P. S., Xu, P., He, Q., Dong, Q., Srinivasan, R., Ganapathy, R., Calderer, R., Cabral, R. S., Stojnic, R., Raileanu, R., Girdhar, R., Patel, R., Sauvestre, R., Polidoro, R., Sumbaly, R., Taylor, R., Silva, R., Hou, R., Wang, R., Hosseini, S., Chennabasappa, S., Singh, S., Bell, S., Kim, S. S., Edunov, S., Nie, S., Narang, S., Raparthy, S., Shen, S., Wan, S., Bhosale, S., Zhang, S., Vandenhende, S., Batra, S., Whitman, S., Sootla, S., Collot, S., Gururangan, S., Borodinsky, S., Herman, T., Fowler, T., Sheasha, T., Georgiou, T., Scialom, T., Speckbacher, T., Mihaylov, T., Xiao, T., Karn, U., Goswami, V., Gupta, V., Ramanathan, V., Kerkez, V., Gonguet, V., Do, V., Vogeti, V., Petrovic, V., Chu, W., Xiong, W., Fu, W., Meers, W., Martinet, X., Wang, X., Tan, X. E., Xie, X., Jia, X., Wang, X., Goldschlag, Y., Gaur, Y., Babaei, Y., Wen, Y., Song, Y., Zhang, Y., Li, Y., Mao, Y., Coudert, Z. D., Yan, Z., Chen, Z., Pa-pakipos, Z., Singh, A., Grattafiori, A., Jain, A., Kelsey, A., Shajnfeld, A., Gangidi, A., Victoria, A., Goldstand, A., Menon, A., Sharma, A., Boesenberg, A., Vaughan, A., Baevski, A., Feinstein, A., Kallet, A., Sangani, A., Yunus, A., Lupu, A., Alvarado, A., Caples, A., Gu, A., Ho, A., Poulton, A., Ryan, A., Ramchandani, A., Franco, A., Saraf, A., Chowdhury, A., Gabriel, A., Bharambe, A., Eisenman, A., Yazdan, A., James, B., Maurer, B., Leonhardi, B., Huang, B., Loyd, B., Paola, B. D., Paran-jape, B., Liu, B., Wu, B., Ni, B., Hancock, B., Wasti, B., Spence, B., Stojkovic, B., Gamido, B., Montalvo, B., Parker, C., Burton, C., Mejia, C., Wang, C., Kim, C., Zhou, C., Hu, C., Chu, C.-H., Cai, C., Tindal, C., Feicht-enhofer, C., Civin, D., Beaty, D., Kreymer, D., Li, D., Wyatt, D., Adkins, D., Xu, D., Testuggine, D., David, D., Parikh, D., Liskovich, D., Foss, D., Wang, D., Le, D., Holland, D., Dowling, E., Jamil, E., Montgomery, E., Presani, E., Hahn, E., Wood, E., Brinkman, E., Arcaute, E., Dunbar, E., Smothers, E., Sun, F., Kreuk, F., Tian, F., Ozgenel, F., Caggioni, F., Guzmán, F., Kanayet, F., Seide, F., Florez, G. M., Schwarz, G., Badeer, G., Swee, G., Halpern, G., Thattai, G., Herman, G., Sizov, G., Guangyi, Zhang, Lakshminarayanan, G., Shojanazeri, H., Zou, H., Wang, H., Zha, H., Habeeb, H., Rudolph, H., Suk, H., Aspegren, H., Goldman, H., Damlaj, I., Molybog, I., Tufanov, I., Veliche, I.-E., Gat, I., Weiss-man, J., Geboski, J., Kohli, J., Asher, J., Gaya, J.-B., Marcus, J., Tang, J., Chan, J., Zhen, J., Reizenstein, J., Teboul, J., Zhong, J., Jin, J., Yang, J., Cummings, J., Carvill, J., Shepard, J., McPhie, J., Torres, J., Ginsburg, J., Wang, J., Wu, K., U, K. H., Saxena, K., Prasad, K., Khandelwal, K., Zand, K., Matosich, K., Veeraragha-

van, K., Michelena, K., Li, K., Huang, K., Chawla, K., Lakhotia, K., Huang, K., Chen, L., Garg, L., A, L., Silva, L., Bell, L., Zhang, L., Guo, L., Yu, L., Moshkovich, L., Wehrstedt, L., Khabsa, M., Avalani, M., Bhatt, M., Tsimpoukelli, M., Mankus, M., Hasson, M., Lennie, M., Reso, M., Groshev, M., Naumov, M., Lathi, M., Keneally, M., Seltzer, M. L., Valko, M., Restrepo, M., Patel, M., Vyatskov, M., Samvelyan, M., Clark, M., Macey, M., Wang, M., Hermoso, M. J., Metanat, M., Rastegari, M., Bansal, M., Santhanam, N., Parks, N., White, N., Bawa, N., Singhal, N., Egebo, N., Usunier, N., Laptev, N. P., Dong, N., Zhang, N., Cheng, N., Chernoguz, O., Hart, O., Salpekar, O., Kalinli, O., Kent, P., Parekh, P., Saab, P., Balaji, P., Rittner, P., Bontrager, P., Roux, P., Dollar, P., Zvyagina, P., Ratanchandani, P., Yuvraj, P., Liang, Q., Alao, R., Rodriguez, R., Ayub, R., Murthy, R., Nayani, R., Mitra, R., Li, R., Hogan, R., Battey, R., Wang, R., Maheswari, R., Howes, R., Rinott, R., Bondu, S. J., Datta, S., Chugh, S., Hunt, S., Dhillon, S., Sidorov, S., Pan, S., Verma, S., Yamamoto, S., Ramaswamy, S., Lindsay, S., Lindsay, S., Feng, S., Lin, S., Zha, S. C., Shankar, S., Zhang, S., Zhang, S., Wang, S., Agarwal, S., Sajuyigbe, S., Chintala, S., Max, S., Chen, S., Kehoe, S., Satterfield, S., Govindaprasad, S., Gupta, S., Cho, S., Virk, S., Subramanian, S., Choudhury, S., Goldman, S., Remez, T., Glaser, T., Best, T., Kohler, T., Robinson, T., Li, T., Zhang, T., Matthews, T., Chou, T., Shaked, T., Vontimitta, V., Ajayi, V., Montanez, V., Mohan, V., Kumar, V. S., Mangla, V., Albiero, V., Ionescu, V., Poenaru, V., Mihailescu, V. T., Ivanov, V., Li, W., Wang, W., Jiang, W., Bouaziz, W., Constable, W., Tang, X., Wang, X., Wu, X., Wang, X., Xia, X., Wu, X., Gao, X., Chen, Y., Hu, Y., Jia, Y., Qi, Y., Li, Y., Zhang, Y., Zhang, Y., Adi, Y., Nam, Y., Yu, Wang, Hao, Y., Qian, Y., He, Y., Rait, Z., DeVito, Z., Rosnbrick, Z., Wen, Z., Yang, Z., and Zhao, Z. The llama 3 herd of models, 2024. URL `https://arxiv.org/abs/2407.21783`.

Duchene, C., Jamet, H., Guillaume, P., and Dehak, R. A benchmark for toxic comment classification on civil comments dataset, 2023. URL `https://arxiv.org/abs/2301.11125`.

Elangovan, A., He, J., and Verspoor, K. Memorization vs. generalization: quantifying data leakage in nlp performance evaluation. *arXiv preprint arXiv:2102.01818*, 2021.

Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., Phang, J., He, H., Thite, A., Nabeshima, N., Presser, S., and Leahy, C. The pile: An 800gb dataset of diverse text for language modeling, 2020. URL `https://arxiv.org/abs/2101.00027`.

Gao, L., Tow, J., Abbasi, B., Biderman, S., Black, S., DiPofi, A., Foster, C., Golding, L., Hsu, J., Le Noac'h, A., Li, H., McDonell, K., Muennighoff, N., Ociepa, C., Phang, J., Reynolds, L., Schoelkopf, H., Skowron, A., Sutawika, L., Tang, E., Thite, A., Wang, B., Wang, K., and Zou, A. A framework for few-shot language model evaluation, 07 2024. URL `https://zenodo.org/records/12608602`.

Golchin, S. and Surdeanu, M. Time travel in llms: Tracing data contamination in large language models, 2023.

Groneveld, D., Beltagy, I., Walsh, P., Bhagia, A., Kinney, R., Tafjord, O., Jha, A. H., Ivison, H., Magnusson, I., Wang, Y., Arora, S., Atkinson, D., Authur, R., Chandu, K. R., Cohan, A., Dumas, J., Elazar, Y., Gu, Y., Hessel, J., Khot, T., Merrill, W., Morrison, J., Muennighoff, N., Naik, A., Nam, C., Peters, M. E., Pyatkin, V., Ravichander, A., Schwenk, D., Shah, S., Smith, W., Strubell, E., Subramani, N., Wortsman, M., Dasigi, P., Lambert, N., Richardson, K., Zettlemoyer, L., Dodge, J., Lo, K., Soldaini, L., Smith, N. A., and Hajishirzi, H. Olmo: Accelerating the science of language models, 2024. URL `https://arxiv.org/abs/2402.00838`.

Gunter, T., Wang, Z., Wang, C., Pang, R., Narayanan, A., Zhang, A., Zhang, B., Chen, C., Chiu, C.-C., Qiu, D., Gopinath, D., Yap, D. A., Yin, D., Nan, F., Weers, F., Yin, G., Huang, H., Wang, J., Lu, J., Peebles, J., Ye, K., Lee, M., Du, N., Chen, Q., Keunebroek, Q., Wiseman, S., Evans, S., Lei, T., Rathod, V., Kong, X., Du, X., Li, Y., Wang, Y., Gao, Y., Ahmed, Z., Xu, Z., Lu, Z., Rashid, A., Jose, A. M., Doane, A., Bencomo, A., Vanderby, A., Hansen, A., Jain, A., Anupama, A. M., Kamal, A., Wu, B., Brum, C., Maalouf, C., Erdenebileg, C., Dulhanty, C., Moritz, D., Kang, D., Jimenez, E., Ladd, E., Shi, F., Bai, F., Chu, F., Hohman, F., Kotek, H., Coleman, H. G., Li, J., Bigham, J., Cao, J., Lai, J., Cheung, J., Shan, J., Zhou, J., Li, J., Qin, J., Singh, K., Vega, K., Zou, K., Heckman, L., Gardiner, L., Bowler, M., Cordell, M., Cao, M., Hay, N., Shahdadpuri, N., Godwin, O., Dighe, P., Rachapudi, P., Tantawi, R., Frigg, R., Davarnia, S., Shah, S., Guha, S., Sirovica, S., Ma, S., Ma, S., Wang, S., Kim, S., Jayaram, S., Shankar, V., Paidi, V., Kumar, V., Wang, X., Zheng, X., Cheng, W., Shrager, Y., Ye, Y., Tanaka, Y., Guo, Y., Meng, Y., Luo, Z. T., Ouyang, Z., Aygar, A., Wan, A., Walkingshaw, A., Narayanan, A., Lin, A., Farooq, A., Ramerth, B., Reed, C., Bartels, C., Chaney, C., Riazati, D., Yang, E. L., Feldman, E., Hochstrasser, G., Seguin, G., Belousova, I., Pelemans, J., Yang, K., Vahid, K. A., Cao, L., Najibi, M., Zuliani, M., Horton, M., Cho, M., Bhendawade, N., Dong, P., Maj, P., Agrawal, P., Shan, Q., Fu, Q., Poston, R., Xu, S., Liu, S., Rao, S., Heeramun, T., Merth, T., Rayala, U., Cui, V., Sridhar, V. R., Zhang, W., Zhang, W., Wu, W., Zhou, X., Liu, X., Zhao, Y., Xia, Y., Ren, Z., and Ren, Z. Ap-

ple intelligence foundation language models, 2024. URL `https://arxiv.org/abs/2407.21075`.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding, 2021a. URL `https://arxiv.org/abs/2009.03300`.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the math dataset, 2021b. URL `https://arxiv.org/abs/2103.03874`.

Imbue. Introducing the 70b model. `https://imbue.com/research/70b-intro/`, 2024.

Iyer, S., Lin, X. V., Pasunuru, R., Mihaylov, T., Simig, D., Yu, P., Shuster, K., Wang, T., Liu, Q., Koura, P. S., Li, X., O'Horo, B., Pereyra, G., Wang, J., Dewan, C., Celikyilmaz, A., Zettlemoyer, L., and Stoyanov, V. Optiml: Scaling language model instruction meta learning through the lens of generalization, 2023.

Jiang, M., Liu, K. Z., Zhong, M., Schaeffer, R., Ouyang, S., Han, J., and Koyejo, S. Investigating data contamination for pre-training language models, 2024. URL `https://arxiv.org/abs/2401.06059`.

Jurafsky, D. and Martin, J. H. *Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition.* Pearson Prentice Hall, Upper Saddle River, N.J., 2009. ISBN 9780131873216 0131873210. URL `http://www.amazon.com/Speech-Language-Processing-2nd-Edition/dp/0131873210/ref=pd_bxgy_b_img_y`.

Kapoor, S. and Narayanan, A. Leakage and the reproducibility crisis in machine-learning-based science. *Patterns*, 4(9), 2023.

Lee, K. Arc-agi is a genuine agi test but o3 cheated :(, May 2024. URL [https://www.lesswrong.com/posts/KHCyituifsHFbZoAC/arc-agi-is-a-genuine-agi-test-but-o3-cheated] (https://www.lesswrong.com/posts/KHCyituifsHFbZoAC/arc-agi-is-a-genuine-agi-test-but-o3-cheated) [https://www.lesswrong.com/posts/KHCyituifsHFbZoAC/arc-agi-is-a-genuine-agi-test-but-o3-cheated] (https://www.lesswrong.com/posts/KHCyituifsHFbZoAC/arc-agi-is-a-genuine-agi-test-but-o3-cheated).

Lee, K., Cooper, A. F., and Grimmelmann, J. Talkin' 'bout ai generation: Copyright and the generative-ai supply chain, 2024. URL `https://arxiv.org/abs/2309.08133`.

Lewis, P., Stenetorp, P., and Riedel, S. Question and answer test-train overlap in open-domain question answering datasets. *arXiv preprint arXiv:2008.02637*, 2020.

Liang, P., Bommasani, R., Lee, T., Tsipras, D., Soylu, D., Yasunaga, M., Zhang, Y., Narayanan, D., Wu, Y., Kumar, A., Newman, B., Yuan, B., Yan, B., Zhang, C., Cosgrove, C., Manning, C. D., Ré, C., Acosta-Navas, D., Hudson, D. A., Zelikman, E., Durmus, E., Ladhak, F., Rong, F., Ren, H., Yao, H., Wang, J., Santhanam, K., Orr, L. J., Zheng, L., Yuksekgonul, M., Suzgun, M., Kim, N. S., Guha, N., Chatterji, N. S., Khattab, O., Henderson, P., Huang, Q., Chi, R., Xie, S. M., Santurkar, S., Ganguli, S., Hashimoto, T., Icard, T. F., Zhang, T., Chaudhary, V., Wang, W., Li, X., Mai, Y., Zhang, Y., and Koreeda, Y. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022.

Liang, P., Bommasani, R., Lee, T., Tsipras, D., Soylu, D., Yasunaga, M., Zhang, Y., Narayanan, D., Wu, Y., Kumar, A., Newman, B., Yuan, B., Yan, B., Zhang, C., Cosgrove, C. A., Manning, C. D., Re, C., Acosta-Navas, D., Hudson, D. A., Zelikman, E., Durmus, E., Ladhak, F., Rong, F., Ren, H., Yao, H., WANG, J., Santhanam, K., Orr, L., Zheng, L., Yuksekgonul, M., Suzgun, M., Kim, N., Guha, N., Chatterji, N. S., Khattab, O., Henderson, P., Huang, Q., Chi, R. A., Xie, S. M., Santurkar, S., Ganguli, S., Hashimoto, T., Icard, T., Zhang, T., Chaudhary, V., Wang, W., Li, X., Mai, Y., Zhang, Y., and Koreeda, Y. Holistic evaluation of language models. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL `https://openreview.net/forum?id=iO4LZibEqW`. Featured Certification, Expert Certification.

Longpre, S., Mahari, R., Chen, A., Obeng-Marnu, N., Sileo, D., Brannon, W., Muennighoff, N., Khazam, N., Kabbara, J., Perisetla, K., Wu, X., Shippole, E., Bollacker, K., Wu, T., Villa, L., Pentland, S., and Hooker, S. The data provenance initiative: A large scale audit of dataset licensing & attribution in ai, 2023. URL `https://arxiv.org/abs/2310.16787`.

Lozhkov, A., Li, R., Allal, L. B., Cassano, F., Lamy-Poirier, J., Tazi, N., Tang, A., Pykhtar, D., Liu, J., Wei, Y., Liu, T., Tian, M., Kocetkov, D., Zucker, A., Belkada, Y., Wang, Z., Liu, Q., Abulkhanov, D., Paul, I., Li, Z., Li, W.-D., Risdal, M., Li, J., Zhu, J., Zhuo, T. Y., Zheltonozhskii, E., Dade, N. O. O., Yu, W., Krauß, L., Jain, N., Su, Y., He, X., Dey, M., Abati, E., Chai, Y., Muennighoff, N., Tang, X., Oblokulov, M., Akiki, C.,

Marone, M., Mou, C., Mishra, M., Gu, A., Hui, B., Dao, T., Zebaze, A., Dehaene, O., Patry, N., Xu, C., McAuley, J., Hu, H., Scholak, T., Paquet, S., Robinson, J., Anderson, C. J., Chapados, N., Patwary, M., Tajbakhsh, N., Jernite, Y., Ferrandis, C. M., Zhang, L., Hughes, S., Wolf, T., Guha, A., von Werra, L., and de Vries, H. Starcoder 2 and the stack v2: The next generation, 2024. URL https://arxiv.org/abs/2402.19173.

Luo, H., Huang, H., Deng, Z., Liu, X., Chen, R., and Liu, Z. Bigbench: A unified benchmark for social bias in text-to-image generative models based on multimodal llm, 2024. URL https://arxiv.org/abs/2407.15240.

Meng. Anthropic memorization, 2025. URL https://x.com/mengk20/status/1904669936032899434. https://x.com/mengk20/status/1904669936032899434.

Mishra, M., Stallone, M., Zhang, G., Shen, Y., Prasad, A., Soria, A. M., Merler, M., Selvam, P., Surendran, S., Singh, S., Sethi, M., Dang, X.-H., Li, P., Wu, K.-L., Zawad, S., Coleman, A., White, M., Lewis, M., Pavuluri, R., Koyfman, Y., Lublinsky, B., de Bayser, M., Abdelaziz, I., Basu, K., Agarwal, M., Zhou, Y., Johnson, C., Goyal, A., Patel, H., Shah, Y., Zerfos, P., Ludwig, H., Munawar, A., Crouse, M., Kapanipathi, P., Salaria, S., Calio, B., Wen, S., Seelam, S., Belgodere, B., Fonseca, C., Singhee, A., Desai, N., Cox, D. D., Puri, R., and Panda, R. Granite code models: A family of open foundation models for code intelligence, 2024. URL https://arxiv.org/abs/2405.04324.

Myrzakhan, A., Bsharat, S. M., and Shen, Z. Open-llm-leaderboard: From multi-choice to open-style questions for llms evaluation, benchmark, and arena, 2024. URL https://arxiv.org/abs/2406.07545.

NVIDIA. Nemotron-4 340b instruct model. https://catalog.ngc.nvidia.com/orgs/nvidia/teams/nemo/models/nemotron-4-340b-instruct, 2024.

OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., Bello, I., Berdine, J., Bernadett-Shapiro, G., Berner, C., Bogdonoff, L., Boiko, O., Boyd, M., Brakman, A.-L., Brockman, G., Brooks, T., Brundage, M., Button, K., Cai, T., Campbell, R., Cann, A., Carey, B., Carlson, C., Carmichael, R., Chan, B., Chang, C., Chantzis, F., Chen, D., Chen, S., Chen, R., Chen, J., Chen, M., Chess, B., Cho, C., Chu, C., Chung, H. W., Cummings, D., Currier, J., Dai, Y., Decareaux, C., Degry, T., Deutsch, N., Deville, D., Dhar, A., Dohan, D., Dowling, S., Dunning, S., Ecoffet, A., Eleti, A., Eloundou, T., Farhi, D., Fedus, L., Felix, N., Fishman, S. P., Forte, J., Fulford, I., Gao, L., Georges, E., Gibson, C., Goel, V., Gogineni, T., Goh, G., Gontijo-Lopes, R., Gordon, J., Grafstein, M., Gray, S., Greene, R., Gross, J., Gu, S. S., Guo, Y., Hallacy, C., Han, J., Harris, J., He, Y., Heaton, M., Heidecke, J., Hesse, C., Hickey, A., Hickey, W., Hoeschele, P., Houghton, B., Hsu, K., Hu, S., Hu, X., Huizinga, J., Jain, S., Jain, S., Jang, J., Jiang, A., Jiang, R., Jin, H., Jin, D., Jomoto, S., Jonn, B., Jun, H., Kaftan, T., Łukasz Kaiser, Kamali, A., Kanitscheider, I., Keskar, N. S., Khan, T., Kilpatrick, L., Kim, J. W., Kim, C., Kim, Y., Kirchner, J. H., Kiros, J., Knight, M., Kokotajlo, D., Łukasz Kondraciuk, Kondrich, A., Konstantinidis, A., Kosic, K., Krueger, G., Kuo, V., Lampe, M., Lan, I., Lee, T., Leike, J., Leung, J., Levy, D., Li, C. M., Lim, R., Lin, M., Lin, S., Litwin, M., Lopez, T., Lowe, R., Lue, P., Makanju, A., Malfacini, K., Manning, S., Markov, T., Markovski, Y., Martin, B., Mayer, K., Mayne, A., McGrew, B., McKinney, S. M., McLeavey, C., McMillan, P., McNeil, J., Medina, D., Mehta, A., Menick, J., Metz, L., Mishchenko, A., Mishkin, P., Monaco, V., Morikawa, E., Mossing, D., Mu, T., Murati, M., Murk, O., Mély, D., Nair, A., Nakano, R., Nayak, R., Neelakantan, A., Ngo, R., Noh, H., Ouyang, L., O'Keefe, C., Pachocki, J., Paino, A., Palermo, J., Pantuliano, A., Parascandolo, G., Parish, J., Parparita, E., Passos, A., Pavlov, M., Peng, A., Perelman, A., de Avila Belbute Peres, F., Petrov, M., de Oliveira Pinto, H. P., Michael, Pokorny, Pokrass, M., Pong, V. H., Powell, T., Power, A., Power, B., Proehl, E., Puri, R., Radford, A., Rae, J., Ramesh, A., Raymond, C., Real, F., Rimbach, K., Ross, C., Rotsted, B., Roussez, H., Ryder, N., Saltarelli, M., Sanders, T., Santurkar, S., Sastry, G., Schmidt, H., Schnurr, D., Schulman, J., Selsam, D., Sheppard, K., Sherbakov, T., Shieh, J., Shoker, S., Shyam, P., Sidor, S., Sigler, E., Simens, M., Sitkin, J., Slama, K., Sohl, I., Sokolowsky, B., Song, Y., Staudacher, N., Such, F. P., Summers, N., Sutskever, I., Tang, J., Tezak, N., Thompson, M. B., Tillet, P., Tootoonchian, A., Tseng, E., Tuggle, P., Turley, N., Tworek, J., Uribe, J. F. C., Vallone, A., Vijayvergiya, A., Voss, C., Wainwright, C., Wang, J. J., Wang, A., Wang, B., Ward, J., Wei, J., Weinmann, C., Welihinda, A., Welinder, P., Weng, J., Weng, L., Wiethoff, M., Willner, D., Winter, C., Wolrich, S., Wong, H., Workman, L., Wu, S., Wu, J., Wu, M., Xiao, K., Xu, T., Yoo, S., Yu, K., Yuan, Q., Zaremba, W., Zellers, R., Zhang, C., Zhang, M., Zhao, S., Zheng, T., Zhuang, J., Zhuk, W., and Zoph, B. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

Oren, Y., Meister, N., Chatterji, N., Ladhak, F., and Hashimoto, T. B. Proving test set contamination in black box language models, 2023.

Rae, J. W., Borgeaud, S., Cai, T., Millican, K., Hoffmann, J., Song, F., Aslanides, J., Henderson, S., Ring, R., Young, S., Rutherford, E., Hennigan, T., Menick, J., Cassirer, A., Powell, R., van den Driessche, G., Hendricks, L. A., Rauh, M., Huang, P.-S., Glaese, A., Welbl, J., Dathathri, S., Huang, S., Uesato, J., Mellor, J., Higgins, I., Creswell, A., McAleese, N., Wu, A., Elsen, E., Jayakumar, S., Buchatskaya, E., Budden, D., Sutherland, E., Simonyan, K., Paganini, M., Sifre, L., Martens, L., Li, X. L., Kuncoro, A., Nematzadeh, A., Gribovskaya, E., Donato, D., Lazaridou, A., Mensch, A., Lespiau, J.-B., Tsimpoukelli, M., Grigorev, N., Fritz, D., Sottiaux, T., Pajarskas, M., Pohlen, T., Gong, Z., Toyama, D., de Masson d'Autume, C., Li, Y., Terzi, T., Mikulik, V., Babuschkin, I., Clark, A., de Las Casas, D., Guy, A., Jones, C., Bradbury, J., Johnson, M., Hechtman, B., Weidinger, L., Gabriel, I., Isaac, W., Lockhart, E., Osindero, S., Rimell, L., Dyer, C., Vinyals, O., Ayoub, K., Stanway, J., Bennett, L., Hassabis, D., Kavukcuoglu, K., and Irving, G. Scaling language models: Methods, analysis insights from training gopher, 2022.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. Squad: 100,000+ questions for machine comprehension of text, 2016. URL https://arxiv.org/abs/1606.05250.

Rajpurkar, P., Jia, R., and Liang, P. Know what you don't know: Unanswerable questions for SQuAD. In *Association for Computational Linguistics (ACL)*, 2018.

Rein, D., Hou, B. L., Stickland, A. C., Petty, J., Pang, R. Y., Dirani, J., Michael, J., and Bowman, S. R. Gpqa: A graduate-level google-proof q&a benchmark, 2023. URL https://arxiv.org/abs/2311.12022.

Roberts, J., Lee, T., Wong, C. H., Yasunaga, M., Mai, Y., and Liang, P. S. Image2struct: Benchmarking structure extraction for vision-language models. *Advances in Neural Information Processing Systems*, 37:115058–115097, 2024.

Russell, S. and Norvig, P. *Artificial Intelligence: A Modern Approach*. Prentice Hall Press, USA, 3rd edition, 2009. ISBN 0136042597.

Scale. Scale seal. https://scale.com/blog/leaderboard, 2024.

Shi, W., Ajith, A., Xia, M., Huang, Y., Liu, D., Blevins, T., Chen, D., and Zettlemoyer, L. Detecting pretraining data from large language models, 2023.

Shoeybi, M., Patwary, M., Puri, R., LeGresley, P., Casper, J., and Catanzaro, B. Megatron-lm: Training multi-billion parameter language models using model parallelism, 2020.

Snowflake. Arctic: Open, efficient foundation language models. https://www.snowflake.com/en/blog/arctic-open-efficient-foundation-language-models-snowflake/, 2024.

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., Madotto, A., Santilli, A., Stuhlmüller, A., Dai, A., 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., Ramírez, C. F., 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, 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., Rodola, 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., Jaimovitch-López, G., Betz, G., Gur-Ari, G., Galijasevic, H., Kim, H., Rashkin, H., Hajishirzi, H., Mehta, H., Bogar, H., Shevlin, H., Schütze, 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., Kocoń, 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. D., Gimpel, K., Omondi, K., Mathewson, K., 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., Colón, L. O., Metz, L., Şenel, L. K., Bosma, M., Sap, M., ter Hoeve, M., Farooqi, M., Faruqui, M., Mazeika, M., Baturan, M., Marelli, M., Maru, M., Quintana, M. J. R., 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., Swedrowski, 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., 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., LeBras, R., Liu, R., Jacobs, R., Zhang, R., Salakhutdinov, R., Chi, R., Lee, 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., Shyamolima, Debnath, 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. T., Shieber, S. M., 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., Prabhu, V. U., 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. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models, 2023. URL https://arxiv.org/abs/2206.04615.

Team, G., Georgiev, P., Lei, V. I., Burnell, R., Bai, L., Gulati, A., Tanzer, G., Vincent, D., Pan, Z., Wang, S., Mariooryad, S., Ding, Y., Geng, X., Alcober, F., Frostig, R., Omernick, M., Walker, L., Paduraru, C., Sorokin, C., Tacchetti, A., Gaffney, C., Daruki, S., Sercinoglu, O., Gleicher, Z., Love, J., Voigtlaender, P., Jain, R., Surita, G., Mohamed, K., Blevins, R., Ahn, J., Zhu, T., Kawintiranon, K., Firat, O., Gu, Y., Zhang, Y., Rahtz, M., Faruqui, M., Clay, N., Gilmer, J., Co-Reyes, J., Penchev, I., Zhu, R., Morioka, N., Hui, K., Haridasan, K., Campos, V., Mahdieh, M., Guo, M., Hassan, S., Kilgour, K., Vezer, A., Cheng, H.-T., de Liedekerke, R., Goyal, S., Barham, P., Strouse, D., Noury, S., Adler, J., Sundararajan, M., Vikram, S., Lepikhin, D., Paganini, M., Garcia, X., Yang, F., Valter, D., Trebacz, M., Vodrahalli, K., Asawaroengchai, C., Ring, R., Kalb, N., Soares, L. B., Brahma, S., Steiner, D., Yu, T., Mentzer, F., He, A., Gonzalez, L., Xu, B., Kaufman, R. L., Shafey, L. E., Oh, J., Hennigan, T., van den Driessche, G., Odoom, S., Lucic, M., Roelofs, B., Lall, S., Marathe, A., Chan, B., Ontanon, S., He, L., Teplyashin, D., Lai, J., Crone, P., Damoc, B., Ho, L., Riedel, S., Lenc, K., Yeh, C.-K., Chowdhery, A., Xu, Y., Kazemi, M., Amid, E., Petrushkina, A., Swersky, K., Khodaei, A., Chen, G., Larkin, C., Pinto, M., Yan, G., Badia, A. P., Patil, P., Hansen, S., Orr, D., Arnold, S. M. R., Grimstad, J., Dai, A., Douglas, S., Sinha, R., Yadav, V., Chen, X., Gribovskaya, E., Austin, J., Zhao, J., Patel, K., Komarek, P., Austin, S., Borgeaud, S., Friso, L., Goyal, A., Caine, B., Cao, K., Chung, D.-W., Lamm, M., Barth-Maron, G., Kagohara, T., Olszewska, K., Chen, M., Shivakumar, K., Agarwal, R., Godhia, H., Rajwar, R., Snaider, J., Dotiwalla, X., Liu, Y., Barua, A., Ungureanu, V., Zhang, Y., Batsaikhan, B.-O., Wirth, M., Qin, J., Danihelka, I., Doshi, T., Chadwick, M., Chen, J., Jain, S., Le, Q., Kar, A., Gurumurthy, M., Li, C., Sang, R., Liu, F., Lamprou, L., Munoz, R., Lintz, N., Mehta, H., Howard, H., Reynolds, M., Aroyo, L., Wang, Q., Blanco, L., Cassirer, A., Griffith, J., Das, D., Lee, S., Sygnowski, J., Fisher, Z., Besley, J., Powell, R., Ahmed, Z., Paulus, D., Reitter, D., Borsos, Z., Joshi, R., Pope, A., Hand, S., Selo, V., Jain, V., Sethi, N., Goel, M., Makino, T., May, R., Yang, Z., Schalkwyk, J., Butterfield, C., Hauth, A., Goldin, A., Hawkins, W., Senter, E., Brin, S., Woodman, O., Ritter, M., Noland, E., Giang, M., Bolina, V., Lee, L., Blyth, T., Mackinnon, I., Reid, M., Sarvana, O., Silver, D., Chen, A., Wang, L., Mag-

giore, L., Chang, O., Attaluri, N., Thornton, G., Chiu, C.-C., Bunyan, O., Levine, N., Chung, T., Eltyshev, E., Si, X., Lillicrap, T., Brady, D., Aggarwal, V., Wu, B., Xu, Y., McIlroy, R., Badola, K., Sandhu, P., Moreira, E., Stokowiec, W., Hemsley, R., Li, D., Tudor, A., Shyam, P., Rahimtoroghi, E., Haykal, S., Sprechmann, P., Zhou, X., Mincu, D., Li, Y., Addanki, R., Krishna, K., Wu, X., Frechette, A., Eyal, M., Dafoe, A., Lacey, D., Whang, J., Avrahami, T., Zhang, Y., Taropa, E., Lin, H., Toyama, D., Rutherford, E., Sano, M., Choe, H., Tomala, A., Safranek-Shrader, C., Kassner, N., Pajarskas, M., Harvey, M., Sechrist, S., Fortunato, M., Lyu, C., Elsayed, G., Kuang, C., Lottes, J., Chu, E., Jia, C., Chen, C.-W., Humphreys, P., Baumli, K., Tao, C., Samuel, R., dos Santos, C. N., Andreassen, A., Rakićević, N., Grewe, D., Kumar, A., Winkler, S., Caton, J., Brock, A., Dalmia, S., Sheahan, H., Barr, I., Miao, Y., Natsev, P., Devlin, J., Behbahani, F., Prost, F., Sun, Y., Myaskovsky, A., Pillai, T. S., Hurt, D., Lazaridou, A., Xiong, X., Zheng, C., Pardo, F., Li, X., Horgan, D., Stanton, J., Ambar, M., Xia, F., Lince, A., Wang, M., Mustafa, B., Webson, A., Lee, H., Anil, R., Wicke, M., Dozat, T., Sinha, A., Piqueras, E., Dabir, E., Upadhyay, S., Boral, A., Hendricks, L. A., Fry, C., Djolonga, J., Su, Y., Walker, J., Labanowski, J., Huang, R., Misra, V., Chen, J., Skerry-Ryan, R., Singh, A., Rijhwani, S., Yu, D., Castro-Ros, A., Changpinyo, B., Datta, R., Bagri, S., Hrafnkelsson, A. M., Maggioni, M., Zheng, D., Sulsky, Y., Hou, S., Paine, T. L., Yang, A., Riesa, J., Rogozinska, D., Marcus, D., Badawy, D. E., Zhang, Q., Wang, L., Miller, H., Greer, J., Sjos, L. L., Nova, A., Zen, H., Chaabouni, R., Rosca, M., Jiang, J., Chen, C., Liu, R., Sainath, T., Krikun, M., Polozov, A., Lespiau, J.-B., Newlan, J., Cankara, Z., Kwak, S., Xu, Y., Chen, P., Coenen, A., Meyer, C., Tsihlas, K., Ma, A., Gottweis, J., Xing, J., Gu, C., Miao, J., Frank, C., Cankara, Z., Ganapathy, S., Dasgupta, I., Hughes-Fitt, S., Chen, H., Reid, D., Rong, K., Fan, H., van Amersfoort, J., Zhuang, V., Cohen, A., Gu, S. S., Mohananey, A., Ilic, A., Tobin, T., Wieting, J., Bortsova, A., Thacker, P., Wang, E., Caveness, E., Chiu, J., Sezener, E., Kaskasoli, A., Baker, S., Millican, K., Elhawaty, M., Aisopos, K., Lebsack, C., Byrd, N., Dai, H., Jia, W., Wiethoff, M., Davoodi, E., Weston, A., Yagati, L., Ahuja, A., Gao, I., Pundak, G., Zhang, S., Azzam, M., Sim, K. C., Caelles, S., Keeling, J., Sharma, A., Swing, A., Li, Y., Liu, C., Bostock, C. G., Bansal, Y., Nado, Z., Anand, A., Lipschultz, J., Karmarkar, A., Proleev, L., Ittycheriah, A., Yeganeh, S. H., Polovets, G., Faust, A., Sun, J., Rrustemi, A., Li, P., Shivanna, R., Liu, J., Welty, C., Lebron, F., Baddepudi, A., Krause, S., Parisotto, E., Soricut, R., Xu, Z., Bloxwich, D., Johnson, M., Neyshabur, B., Mao-Jones, J., Wang, R., Ramasesh, V., Abbas, Z., Guez, A., Segal, C., Nguyen, D. D., Svensson, J., Hou, L., York, S., Milan, K., Bridgers, S., Gworek, W., Tagliasacchi, M., Lee-Thorp, J., Chang, M., Guseynov, A., Hartman, A. J., Kwong, M., Zhao, R., Kashem, S., Cole, E., Miech, A., Tanburn, R., Phuong, M., Pavetic, F., Cevey, S., Comanescu, R., Ives, R., Yang, S., Du, C., Li, B., Zhang, Z., Iinuma, M., Hu, C. H., Roy, A., Bijwadia, S., Zhu, Z., Martins, D., Saputro, R., Gergely, A., Zheng, S., Jia, D., Antonoglou, I., Sadovsky, A., Gu, S., Bi, Y., Andreev, A., Samangooei, S., Khan, M., Kocisky, T., Filos, A., Kumar, C., Bishop, C., Yu, A., Hodkinson, S., Mittal, S., Shah, P., Moufarek, A., Cheng, Y., Bloniarz, A., Lee, J., Pejman, P., Michel, P., Spencer, S., Feinberg, V., Xiong, X., Savinov, N., Smith, C., Shakeri, S., Tran, D., Chesus, M., Bohnet, B., Tucker, G., von Glehn, T., Muir, C., Mao, Y., Kazawa, H., Slone, A., Soparkar, K., Shrivastava, D., Cobon-Kerr, J., Sharman, M., Pavagadhi, J., Araya, C., Misiunas, K., Ghelani, N., Laskin, M., Barker, D., Li, Q., Briukhov, A., Houlsby, N., Glaese, M., Lakshminarayanan, B., Schucher, N., Tang, Y., Collins, E., Lim, H., Feng, F., Recasens, A., Lai, G., Magni, A., Cao, N. D., Siddhant, A., Ashwood, Z., Orbay, J., Dehghani, M., Brennan, J., He, Y., Xu, K., Gao, Y., Saroufim, C., Molloy, J., Wu, X., Arnold, S., Chang, S., Schrittwieser, J., Buchatskaya, E., Radpour, S., Polacek, M., Giordano, S., Bapna, A., Tokumine, S., Hellendoorn, V., Sottiaux, T., Cogan, S., Severyn, A., Saleh, M., Thakoor, S., Shefey, L., Qiao, S., Gaba, M., yiin Chang, S., Swanson, C., Zhang, B., Lee, B., Rubenstein, P. K., Song, G., Kwiatkowski, T., Koop, A., Kannan, A., Kao, D., Schuh, P., Stjerngren, A., Ghiasi, G., Gibson, G., Vilnis, L., Yuan, Y., Ferreira, F. T., Kamath, A., Klimenko, T., Franko, K., Xiao, K., Bhattacharya, I., Patel, M., Wang, R., Morris, A., Strudel, R., Sharma, V., Choy, P., Hashemi, S. H., Landon, J., Finkelstein, M., Jhakra, P., Frye, J., Barnes, M., Mauger, M., Daun, D., Baatarsukh, K., Tung, M., Farhan, W., Michalewski, H., Viola, F., de Chaumont Quitry, F., Lan, C. L., Hudson, T., Wang, Q., Fischer, F., Zheng, I., White, E., Dragan, A., baptiste Alayrac, J., Ni, E., Pritzel, A., Iwanicki, A., Isard, M., Bulanova, A., Zilka, L., Dyer, E., Sachan, D., Srinivasan, S., Muckenhirn, H., Cai, H., Mandhane, A., Tariq, M., Rae, J. W., Wang, G., Ayoub, K., FitzGerald, N., Zhao, Y., Han, W., Alberti, C., Garrette, D., Krishnakumar, K., Gimenez, M., Levskaya, A., Sohn, D., Matak, J., Iturrate, I., Chang, M. B., Xiang, J., Cao, Y., Ranka, N., Brown, G., Hutter, A., Mirrokni, V., Chen, N., Yao, K., Egyed, Z., Galilee, F., Liechty, T., Kallakuri, P., Palmer, E., Ghemawat, S., Liu, J., Tao, D., Thornton, C., Green, T., Jasarevic, M., Lin, S., Cotruta, V., Tan, Y.-X., Fiedel, N., Yu, H., Chi, E., Neitz, A., Heitkaemper, J., Sinha, A., Zhou, D., Sun, Y., Kaed, C., Hulse, B., Mishra, S., Georgaki, M., Kudugunta, S., Farabet, C., Shafran, I., Vlasic, D., Tsitsulin, A., Ananthanarayanan, R., Carin, A., Su, G., Sun, P., V, S., Carvajal, G., Broder, J., Comsa, I., Re-

pina, A., Wong, W., Chen, W. W., Hawkins, P., Filonov, E., Loher, L., Hirnschall, C., Wang, W., Ye, J., Burns, A., Cate, H., Wright, D. G., Piccinini, F., Zhang, L., Lin, C.-C., Gog, I., Kulizhskaya, Y., Sreevatsa, A., Song, S., Cobo, L. C., Iyer, A., Tekur, C., Garrido, G., Xiao, Z., Kemp, R., Zheng, H. S., Li, H., Agarwal, A., Ngani, C., Goshvadi, K., Santamaria-Fernandez, R., Fica, W., Chen, X., Gorgolewski, C., Sun, S., Garg, R., Ye, X., Eslami, S. M. A., Hua, N., Simon, J., Joshi, P., Kim, Y., Tenney, I., Potluri, S., Thiet, L. N., Yuan, Q., Luisier, F., Chronopoulou, A., Scellato, S., Srinivasan, P., Chen, M., Koverkathu, V., Dalibard, V., Xu, Y., Saeta, B., Anderson, K., Sellam, T., Fernando, N., Huot, F., Jung, J., Varadarajan, M., Quinn, M., Raul, A., Le, M., Habalov, R., Clark, J., Jalan, K., Bullard, K., Singhal, A., Luong, T., Wang, B., Rajayogam, S., Eisenschlos, J., Jia, J., Finchelstein, D., Yakubovich, A., Balle, D., Fink, M., Agarwal, S., Li, J., Dvijotham, D., Pal, S., Kang, K., Konzelmann, J., Beattie, J., Dousse, O., Wu, D., Crocker, R., Elkind, C., Jonnalagadda, S. R., Lee, J., Holtmann-Rice, D., Kallarackal, K., Liu, R., Vnukov, D., Vats, N., Invernizzi, L., Jafari, M., Zhou, H., Taylor, L., Prendki, J., Wu, M., Eccles, T., Liu, T., Kopparapu, K., Beaufays, F., Angermueller, C., Marzoca, A., Sarcar, S., Dib, H., Stanway, J., Perbet, F., Trdin, N., Sterneck, R., Khorlin, A., Li, D., Wu, X., Goenka, S., Madras, D., Goldshtein, S., Gierke, W., Zhou, T., Liu, Y., Liang, Y., White, A., Li, Y., Singh, S., Bahargam, S., Epstein, M., Basu, S., Lao, L., Ozturel, A., Crous, C., Zhai, A., Lu, H., Tung, Z., Gaur, N., Walton, A., Dixon, L., Zhang, M., Globerson, A., Uy, G., Bolt, A., Wiles, O., Nasr, M., Shumailov, I., Selvi, M., Piccinno, F., Aguilar, R., McCarthy, S., Khalman, M., Shukla, M., Galic, V., Carpenter, J., Villela, K., Zhang, H., Richardson, H., Martens, J., Bosnjak, M., Belle, S. R., Seibert, J., Alnahlawi, M., McWilliams, B., Singh, S., Louis, A., Ding, W., Popovici, D., Simicich, L., Knight, L., Mehta, P., Gupta, N., Shi, C., Fatehi, S., Mitrovic, J., Grills, A., Pagadora, J., Petrova, D., Eisenbud, D., Zhang, Z., Yates, D., Mittal, B., Tripuraneni, N., Assael, Y., Brovelli, T., Jain, P., Velimirovic, M., Akbulut, C., Mu, J., Macherey, W., Kumar, R., Xu, J., Qureshi, H., Comanici, G., Wiesner, J., Gong, Z., Ruddock, A., Bauer, M., Felt, N., GP, A., Arnab, A., Zelle, D., Rothfuss, J., Rosgen, B., Shenoy, A., Seybold, B., Li, X., Mudigonda, J., Erdogan, G., Xia, J., Simsa, J., Michi, A., Yao, Y., Yew, C., Kan, S., Caswell, I., Radebaugh, C., Elisseeff, A., Valenzuela, P., McKinney, K., Paterson, K., Cui, A., Latorre-Chimoto, E., Kim, S., Zeng, W., Durden, K., Ponnapalli, P., Sosea, T., Choquette-Choo, C. A., Manyika, J., Robenek, B., Vashisht, H., Pereira, S., Lam, H., Velic, M., Owusu-Afriyie, D., Lee, K., Bolukbasi, T., Parrish, A., Lu, S., Park, J., Venkatraman, B., Talbert, A., Rosique, L., Cheng, Y., Sozanschi, A., Paszke, A., Kumar, P., Austin,

J., Li, L., Salama, K., Kim, W., Dukkipati, N., Baryshnikov, A., Kaplanis, C., Sheng, X., Chervonyi, Y., Unlu, C., de Las Casas, D., Askham, H., Tunyasuvunakool, K., Gimeno, F., Poder, S., Kwak, C., Miecnikowski, M., Mirrokni, V., Dimitriev, A., Parisi, A., Liu, D., Tsai, T., Shevlane, T., Kouridi, C., Garmon, D., Goedeckemeyer, A., Brown, A. R., Vijayakumar, A., Elqursh, A., Jazayeri, S., Huang, J., Carthy, S. M., Hoover, J., Kim, L., Kumar, S., Chen, W., Biles, C., Bingham, G., Rosen, E., Wang, L., Tan, Q., Engel, D., Pongetti, F., de Cesare, D., Hwang, D., Yu, L., Pullman, J., Narayanan, S., Levin, K., Gopal, S., Li, M., Aharoni, A., Trinh, T., Lo, J., Casagrande, N., Vij, R., Matthey, L., Ramadhana, B., Matthews, A., Carey, C., Johnson, M., Goranova, K., Shah, R., Ashraf, S., Dasgupta, K., Larsen, R., Wang, Y., Vuyyuru, M. R., Jiang, C., Ijazi, J., Osawa, K., Smith, C., Boppana, R. S., Bilal, T., Koizumi, Y., Xu, Y., Altun, Y., Shabat, N., Bariach, B., Korchemniy, A., Choo, K., Ronneberger, O., Iwuanyanwu, C., Zhao, S., Soergel, D., Hsieh, C.-J., Cai, I., Iqbal, S., Sundermeyer, M., Chen, Z., Bursztein, E., Malaviya, C., Biadsy, F., Shroff, P., Dhillon, I., Latkar, T., Dyer, C., Forbes, H., Nicosia, M., Nikolaev, V., Greene, S., Georgiev, M., Wang, P., Martin, N., Sedghi, H., Zhang, J., Banzal, P., Fritz, D., Rao, V., Wang, X., Zhang, J., Patraucean, V., Du, D., Mordatch, I., Jurin, I., Liu, L., Dubey, A., Mohan, A., Nowakowski, J., Ion, V.-D., Wei, N., Tojo, R., Raad, M. A., Hudson, D. A., Keshava, V., Agrawal, S., Ramirez, K., Wu, Z., Nguyen, H., Liu, J., Sewak, M., Petrini, B., Choi, D., Philips, I., Wang, Z., Bica, I., Garg, A., Wilkiewicz, J., Agrawal, P., Li, X., Guo, D., Xue, E., Shaik, N., Leach, A., Khan, S. M., Wiesinger, J., Jerome, S., Chakladar, A., Wang, A. W., Ornduff, T., Abu, F., Ghaffarkhah, A., Wainwright, M., Cortes, M., Liu, F., Maynez, J., Terzis, A., Samangouei, P., Mansour, R., Kepa, T., Aubet, F.-X., Algymr, A., Banica, D., Weisz, A., Orban, A., Senges, A., Andrejczuk, E., Geller, M., Santo, N. D., Anklin, V., Merey, M. A., Baeuml, M., Strohman, T., Bai, J., Petrov, S., Wu, Y., Hassabis, D., Kavukcuoglu, K., Dean, J., and Vinyals, O. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, 2024a. URL https://arxiv.org/abs/2403.05530.

Team, R., Ormazabal, A., Zheng, C., de Masson d'Autume, C., Yogatama, D., Fu, D., Ong, D., Chen, E., Lamprecht, E., Pham, H., Ong, I., Aleksiev, K., Li, L., Henderson, M., Bain, M., Artetxe, M., Relan, N., Padlewski, P., Liu, Q., Chen, R., Phua, S., Yang, Y., Tay, Y., Wang, Y., Zhu, Z., and Xie, Z. Reka core, flash, and edge: A series of powerful multimodal language models, 2024b. URL https://arxiv.org/abs/2404.12387.

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 Lam-

ple, G. Llama: Open and efficient foundation language models. *arXiv*, 2023.

Vu, T.-T., He, X., Haffari, G., and Shareghi, E. Koala: An index for quantifying overlaps with pre-training corpora. *arXiv preprint arXiv:2303.14770*, 2023.

White, C., Dooley, S., Roberts, M., Pal, A., Feuer, B., Jain, S., Shwartz-Ziv, R., Jain, N., Saifullah, K., Naidu, S., Hegde, C., LeCun, Y., Goldstein, T., Neiswanger, W., and Goldblum, M. Livebench: A challenging, contamination-free llm benchmark, 2024. URL https://arxiv.org/abs/2406.19314.

Writer. Writer helm results, 2024. URL https://writer.com/blog/palmyra-helm-benchmark/. accessed on 10/10/2024.

x.ai. Announcing grok 2. https://x.ai/blog/grok-2, 2024.

Yang, A., Yang, B., Hui, B., Zheng, B., Yu, B., Zhou, C., Li, C., Li, C., Liu, D., Huang, F., Dong, G., Wei, H., Lin, H., Tang, J., Wang, J., Yang, J., Tu, J., Zhang, J., Ma, J., Yang, J., Xu, J., Zhou, J., Bai, J., He, J., Lin, J., Dang, K., Lu, K., Chen, K., Yang, K., Li, M., Xue, M., Ni, N., Zhang, P., Wang, P., Peng, R., Men, R., Gao, R., Lin, R., Wang, S., Bai, S., Tan, S., Zhu, T., Li, T., Liu, T., Ge, W., Deng, X., Zhou, X., Ren, X., Zhang, X., Wei, X., Ren, X., Liu, X., Fan, Y., Yao, Y., Zhang, Y., Wan, Y., Chu, Y., Liu, Y., Cui, Z., Zhang, Z., Guo, Z., and Fan, Z. Qwen2 technical report, 2024. URL https://arxiv.org/abs/2407.10671.

Yang, S., Chiang, W.-L., Zheng, L., Gonzalez, J. E., and Stoica, I. Rethinking benchmark and contamination for language models with rephrased samples, 2023.

A. Scoring Details

For developers that do not openly release their training data, we provide additional explanation below as to why their transparency regarding train-test overlap is meaningful; for quantification of the degree of train-test overlap for each of these developers, see their associated technical reports.

OpenAI reports its train-test overlap analysis in the GPT-4 Technical Report (OpenAI et al., 2024). OpenAI reports results for GPT-4 on 8 public test sets and shared train-test overlap analysis for 6 of these test sets. OpenAI et al. (2024) states *"We measure cross-contamination between our evaluation dataset and the pre-training data using substring match. Both evaluation and training data are processed by removing all spaces and symbols, keeping only*

characters (including numbers). For each evaluation example, we randomly select three substrings of 50 characters (or use the entire example if it's less than 50 characters). A match is identified if any of the three sampled evaluation substrings is a substring of the processed training example. This yields a list of contaminated examples. We discard these and rerun to get uncontaminated scores."

Meta reports its train-test overlap analysis in the Llama 3 Technical Report (Dubey et al., 2024, Section 5.1.4). Dubey et al. (2024) write: *"Singh et al. (2024) propose to select contamination detection methods empirically, based on which method results in the largest difference between the 'clean' part of the dataset and the entire dataset, which they call estimated performance gain. For all our evaluation datasets, we score examples based on 8-gram overlap, a method that was found by Singh et al. (2024) to be accurate for many datasets. We consider an example of a dataset D to be contaminated if a ratio TD of its tokens are part of an 8-gram occurring at least once in the pre-training corpus. We select TD separately for each dataset, based on which value shows the maximal significant estimated performance gain across the three model sizes."* Meta reports train-test overlap for Llama 3.1 models on AGIEval, BIG-Bench Hard, BoolQ, CommonSenseQA, GSM8K, HellaSwag, MATH, NaturalQuestions, OpenBookQA, PiQA, QuaC, SiQA, SQuAD, Winogrande, and WorldSense."

Writer released train-test overlap statistics for Palmyra X after receiving a request from the authors. Writer ran a script [6] provided by the authors over its pretraining and instruction-tuning data to evaluate train-test overlap via an n-gram analysis. Writer publishes its results on HELM Lite (Writer, 2024), which includes 9 public test sets, and Writer reported train-test overlap on each of the public test sets included in HELM as well as others. Writer found some degree of train-test overlap for 13 of the 27 test sets on which it ran the script (APPS, CivilComments, CNN/Daily Mail, EntityMatching, HumanEval, ICE, LegalSupport, MATH, NarrativeQA, RAFT, The Pile, WikiFact, XSum).

Alibaba released train-test overlap statistics for Qwen2 via an update to its technical report after receiving a request from the authors (Yang et al., 2024, Section 5.2.6). Yang et al. (2024) conducted an analysis of Qwen2's training set following OpenAI's approach, writing that in addition to n-gram matching *"we also applied another constraint based on the longest common subsequence (LCS). Specifically, we first remove all symbols and punctuation from both the test and training sequences and perform*

---

[6]This script is publicly available and attached in the supplementary material–we encourage other developers to run it over their training sets and publicly report the results.

tokenization. For a training sequence st, we remove it if there is a test sequence se such that $|LCS(st, se)| \geq 13$ and $|LCS(st, se)| \geq 0.6 \times \min(|st|, |se|)$. To assess the potential effects of leaking data on the test performance, we follow OpenAI (2023) to construct a strict non-contaminated test set to check if there is a significant performance degradation after strict decontamination. Specifically, we construct the non-contaminated test set by excluding any sample which has 13-gram overlap with the pre-training or the post-training data (without constraint on LCS), and then compute the corresponding metric on the test set.” Alibaba reports results for Qwen2-72B on 14 public test sets (MMLU, GPQA, TheoremQA, HumanEval, MBPP, MultiPL-E, IFEval, LiveCodeBench v1, GSM8K, MATH, MT-Bench, MixEval, ArenaHard, and AlignBench) and reported train-test overlap on 8 of the public test sets (MMLU, GPQA, HumanEval, MBPP, MultiPL-E, GSM8K, MATH, and IFEval).

Apple   reports train-test overlap statistics for Apple Intelligence on 24 benchmarks (MMLU, GSM8K, HellaSwag, WinoGrande, NarrativeQA, Natural Questions, Open-BookQA, MATH_CoT, LegalBench, MedQA, WMT-2014, IFEval, AlpacaEval, ArenaHard, Berkeley Functional Calling, arc_challenge, arc_easy, lambada, piqa, sciq, triviaQA, webqs, HumanEval, MultiPLE-Swift); of these, Apple prefiltered its training data against 12 (MMLU, GSM8K, HellaSwag, WinoGrande, OpenBookQA, arc_challenge, arc_easy, lambada, piqa, sciq, triviaQA, webqs), filtering documents upon 4-13 gram collisions unless the n-gram reaches a “common-usage” threshold of 1000 (Gunter et al., 2024). Apple provided specificity about the benchmarks for which its training data was prefiltered after receiving a request from the authors.

B. Protocol

As the training data for many language models is private, we propose a protocol (see Figure 1) to coordinate between the model provider (first party) and the external entity (third party) to report train-test overlap statistics. Through this protocol, relevant train-test overlap statistics can be made public while respecting the access controls placed on the underlying training data.

B.1. Notation

The training data, denoted D_{train}, is a set of examples where each $x_{\text{train}} \in D_{\text{train}}$ is a sequence of tokens. The test data consists of one or more test sets D_{test}, where D_{test} is a set of examples where each instance $(x_{\text{test}}, Y_{\text{test}}) \in D_{\text{test}}$ consists of an input sequence of tokens x_{test} and a reference set of token sequences $y_{\text{test}} \in Y_{\text{test}}$ representing the ground truth response. For instance, for a Q&A test set, input would be

the question whereas reference are the answer choices. We concatenate the reference set into a single sequence, so we will refer to the reference set as a single reference token sequence y_{test} hereafter.

B.2. Steps

We divide the above described process of computing train-test overlap metrics between a first-party actor with access to the training set and a third-party actor in the following manner (depicted in Figure 1):

1. The third-party actor(s) release test data, **compute-instance-overlap-statistics**, **aggregate-overlap-statistics**, **aggregate-metrics** publicly to instantiate the protocol.

2. The first-party actor uses **compute-instance-overlap-statistics** on their D_{train} and publicly available test data to output instance-level overlap statistics.

3. The third-party actor uses **aggregate-overlap-statistics** on the instance-level overlap statistics to produce test-set-level statistics, which are published as meaningful overlap information between D_{train} and test data.

In terms of costs, we distinguish **compute-instance-overlap-statistics** as the computationally expensive step relative to **aggregate-overlap-statistics** as training sets are often several terabytes of data. This enables rapid iteration with different aggregation methods on the instance-level overlap stats.

C. Protocol Instantiation

While this protocol is generic with respect to the particular method of computing train-test overlap, we instantiate it with n-gram overlap as it is the predominant method used by model providers and computationally inexpensive (Brown et al., 2020a; Touvron et al., 2023). Intuitively, for large enough n, the higher the overlap between the n-grams in D_{train} and the n-grams in D_{test} the larger the likelihood that the training set is “contaminated”, *i.e.*, contains significant portions of the test set. A variety of metrics aimed at quantifying this intuition were proposed in earlier works, each proposing a different metric over the intersection between n-grams from D_{train} and D_{test}. We denote the family of such overlap metrics as $\mathcal{F}$, and conceptually organize some of the previously proposed overlap metrics $f \in \mathcal{F}$ in Section **??**. Below, we use the abstract form $f \in \mathcal{F}$ and instantiate the protocol in Figure 1 using n-gram overlap with the following test data, **compute-instance-overlap-statistics**, **aggregate-overlap-statistics**:
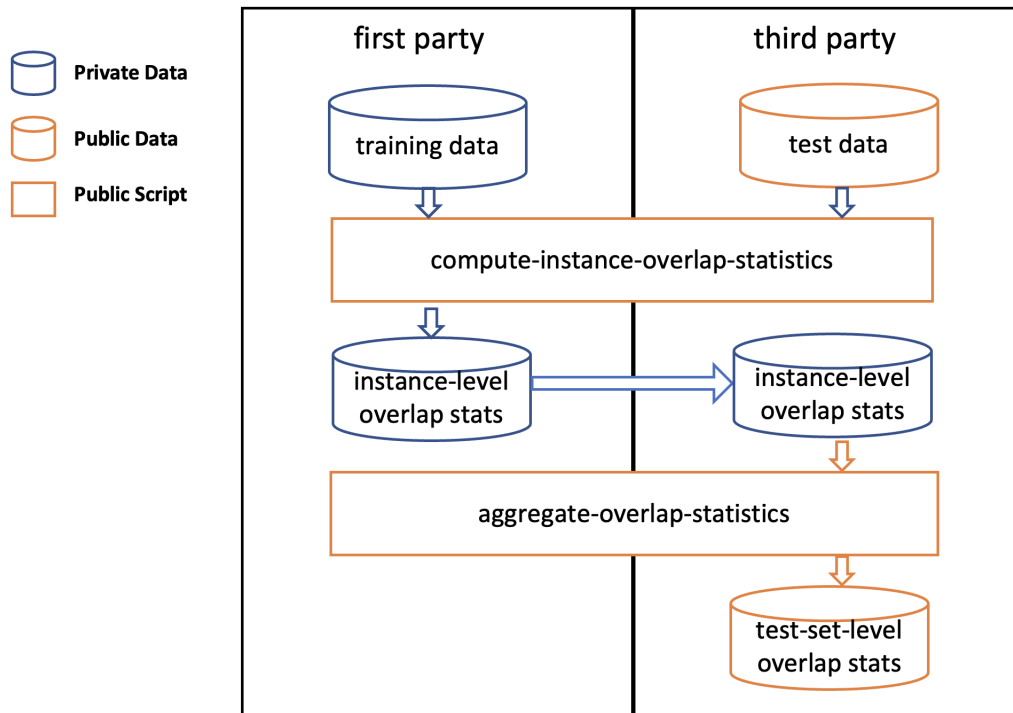
Figure 1: **Protocol for computing train-test overlap.** The first party takes private training data and public test data to compute overlap statistics at the instance level. This is sent to the third party, which aggregates statistics to the dataset level and publishes the results publicly.

1. The instantiated test data are HELM scenarios (Liang et al., 2022).

2. The instantiated **compute-instance-overlap-statistics** takes in their private D_{train} and test data, and for each D_{test} in HELM, outputs the n-gram overlap between each instance of the test set and the entire training set. In order to gain insight on the effect of training set overlap with test input x_{test} versus training set overlap with test reference y_{test}, this function separately outputs the overlap for the input: n-grams$(D_{\text{train}}) \cap$ n-grams(x_{test}) and for the reference: n-grams$(D_{\text{train}}) \cap$ n-grams(y_{test}), so overall 2 sets of n-grams are generated per D_{test}. The overlapping n-gram sets are then used to generate metrics: for a given overlap metric $f \in \mathcal{F}$ the per-instance overlap scores: $\forall (x_{\text{test}}, y_{\text{test}}) \in D_{\text{test}}$: f(n-grams$(D_{\text{train}}) \cap$ n-grams$(x_{\text{test}}))$; f(n-grams$(D_{\text{train}}) \cap$ n-grams$(y_{\text{test}}))$. We present several previous instantiations of $f \in \mathcal{F}$ in Section D.

3. The instantiated **aggregate-overlap-statistics** takes in the per-instance scores and then aggregates the statistics for public release. We present these aggregations in the results section. These aggregations are the only public information released regarding D_{train}.

Note that there are additional complexities in terms of the protocol. For instance, different providers are comfortable with sharing varying levels of instance information. For simplicity, we note that almost all providers were willing to share instance-level metrics without sharing instance-ids or n-grams. One provider was willing to directly share n-grams, which enabled us to compute metrics on our own without their input (hence saving their time and reducing the need for us to reach out after modifying the protocol). With another, we were only able to run an earlier version of the protocol where we had only binary overlap and have not been able to successfully rerun the protocol since.

D. Train-test overlap metrics

In this section we formally present the train-test overlap metrics. We begin by surveying the existing methods that were used for computing n-gram based overlap metrics for leading LLMs, and from which we derive three core metrics.

D.1. Existing Methods

We surveyed literature on how providers have evaluated train-test overlap on existing LMs to establish a baseline for what providers are familiar and comfortable with. Table 3 contains information on how leading LLMs have computed overlap. All the methods were ngram-based and 13-tokens

was the most common. From n-grams, providers computed either binary, jaccard, or token overlap. Binary overlap is the most common and simplest method, which simply marks an instance as overlapping if there is a single overlapping n-gram, and not overlapping otherwise. In contrast, jaccard and token overlap compute a score based on what portion of a given instance is overlapping. Jaccard overlap takes the fraction of the number of n-grams that are overlapping over the total number of n-grams for a given test example. Token overlap counts the number of overlapping tokens over the total number of tokens for a given test instance, which avoids double counting any given token. We provide an example of binary, jaccard, and token overlap below and then define the metrics more formally later. We compute binary, jaccard, and token overlap for the training data as they can be derived from the same primitives (n-grams) and thus require minimal additional compute.

Note that there are additional variations that we do not capture here. For instance, the GPT-3 prefiltering stage computed the frequency of 13-grams within the training set and filtered out those with a frequency greater than 10 (Brown et al., 2020a). We explored filtering and weighting by frequency, but did not find the effect to be sufficiently meaningful to justify the added complexity. We do not capture other variants, e.g. GPT-4 only subsamples 3 50-character segments (OpenAI, 2023) and Llama 2 introduces skip-grams (Touvron et al., 2023)

D.2. Metric definition

Based on our literature review, we choose to compute binary, jaccard, and token overlap on 13-grams.

We define the above metrics formally by specifying various overlap metric functions $f \in \mathcal{F}$, where $\mathcal{F}$ is the family of functions f that take a set of sequences of tokens D_1 and a token sequence x as input and produce a real value between 0 and 1.

We take D_1 as D_{test} and $x \in D_{\text{train}}$. When running the algorithm, we load the n-grams associated with the entire D_{test} into memory and iterate through the test set n-gram by n-gram to compute overlap. While we've previously noted that instances in D_{test} can have two components, input and reference, we will define metrics in terms of token sequence x for simplicity.

Binary overlap marks an instance as overlapping if there is at least a single overlapping n-gram. Mathematically, it is defined as:

$$f_{\text{binary}}(D_{\text{train}}, x) = \min(|\text{n-grams}(x) \cap \text{n-grams}(D_{\text{train}})|, 1)$$

Jaccard overlap measures how many n-grams are overlapping for an instance and divides by the total number of

Table 3: **Overlap Metrics of Existing Models.** Here we note how train-test overlap was computed for selected existing models. In addition to analyzing overlap after training, certain providers filtered training or test data based on overlap. All methods are n-gram based using tokens, besides GPT-4 which is based on characters.

Model	Measurement	Type	Stage
Llama 2 (Touvron et al., 2023)	>10 tokens	Token	Post-training analysis
PALM (Chowdhery et al., 2022b)	8 tokens	Jaccard	Post-training analysis
GPT-3 (Brown et al., 2020a)	13 tokens	Binary	Pre-training filtering
GPT-3 (Brown et al., 2020a)	8-13 tokens	Binary	Post-training analysis
GPT-4 (OpenAI, 2023)	50 characters	Binary	Post-training analysis
Gopher (Rae et al., 2022)	13 tokens	Jaccard	Pre-training filtering
OPT-IML (Iyer et al., 2023)	13 tokens	Binary	Post-training filtering
Megatron GPT2 (Shoeybi et al., 2020)	8 tokens	Binary	Post-training analysis

---

Example of Overlap Metrics:
Example sentence: "this is a fake example sentence for showing how we compute metrics"
Example overlapping n-grams (for $n = 3$): [("this is a"), ("is a fake"), ("for showing how")]

Metric	Value	Explanation
Binary overlap	1	There is at least one overlapping n-gram.
Jaccard overlap	$3/10$	There are 3 overlapping n-grams out of 10 total n-grams.
Token overlap	$7/12$	There are 7 overlapping tokens out of a total of 12 tokens.

---

n-grams in that instance. Jaccard overlap is defined as:

$$f_{\text{Jaccard}}(D_{\text{train}}, x) = \frac{|\text{n-grams}(x) \cap \text{n-grams}(D_{\text{train}})|}{|\text{n-grams}(x)|}$$

Token overlap measures how many tokens are overlapping for an instance and divides by the total number of tokens in that instance. A token is overlapping if it is associated with at least one overlapping n-gram. This is similar to jaccard, but does not double count tokens for contiguous n-grams. Mathematically: Let tokens(x) denote the tokens $x_1, x_2, ..., x_k$ corresponding to x.
Let tokens-intersect(x, D) denote the tokens corresponding to n-grams$(x) \cap$ n-grams(D) without duplication of any x_i. That is, for a given n-gram at starting index i and length n, each of the tokens $x_i, x_{i+1}, ..., x_{i+n-1}$ are included in tokens-intersect(x, D) if the n-gram exists in n-grams(D), without duplicates for any given index. For instance if n-grams starting at i and $i + 1$ are both in n-grams(D), tokens-intersect(x, D) includes only a single instance of x_{i+1}, even though x_i is associated with the n-grams at both i and $i + 1$.

$$f_{\text{Token}}(D_{\text{train}}, x) = \frac{|\text{tokens-intersect}(x, D_{\text{train}})|}{|\text{tokens}(x)|}$$

D.3. Metric Aggregation

We aggregate the metrics in two ways. **Possible overlap** instances are those with any overlap in either the input or reference, *i.e.*, binary overlap value of 1. We report the scores separately between input and reference rather than the union. **Likely overlap** instances are those where both the input and reference overlap of an instance is "dirty". Similar to Touvron et al. (2023), we define "dirty" as a token overlap score of 0.8 or greater, though we simply define any other score as "clean" for simplicity.

D.4. Algorithm

Algorithm for computing overlapping n-grams and frequencies. Code will be released on GitHub.

Algorithm 1 Compute Overlapping N-grams

Require: Test set examples $x_{\text{test}} \in D_{\text{test}}$
Require: Train set examples $x_{\text{train}} \in D_{\text{train}}$
 1: Read the test set examples $x_{\text{train}} \in D_{\text{train}}$ into a hash table h into memory
 2: **for** $x_{\text{train}} \in D_{\text{train}}$ **do**
 3:     Split it into n-grams $n_1, \ldots, n_k$
 4:     **for** each n-gram n_j **do**
 5:         **if** n_j exists in the hash table h **then**
 6:             Mark n_j as overlapping
 7:         **end if**
 8:     **end for**
 9: **end for**
10: Output the overlapping n-grams, and associated test examples

---