DSDM: Model-Aware Dataset Selection with Datamodels

Logan Engstrom [1] **Axel Feldmann** [1] **Aleksander Mądry** [1]

Abstract

When selecting data for training large-scale models, standard practice is to filter for examples that match human notions of data quality. Such filtering yields qualitatively clean datapoints that intuitively should improve model behavior. However, in practice the opposite can often happen: we find that selecting according to similarity with "high quality" data sources may not increase (and can even *hurt*) performance compared to randomly selecting data. To develop better methods for selecting data, we start by framing dataset selection as an optimization problem that we can directly solve for: given target tasks, a learning algorithm, and candidate data, select the subset that maximizes model performance. This framework thus avoids handpicked notions of data quality, and instead models explicitly how the learning process uses train datapoints to predict on the target tasks. Our resulting method greatly improves language model (LM) performance on both pre-specified tasks and *previously unseen* tasks. Specifically, choosing target tasks representative of standard LM problems and evaluating on diverse held-out benchmarks, our selected datasets provide a 2× compute multiplier over baseline methods.

1. Introduction

Suppose we want to train a large-scale machine learning model. What data should we train on? The simple answer is: as much data as possible. For example, we train language and vision models on vast quantities of text (Radford et al., 2019) and image-caption (Ramesh et al., 2021) data from sources like internet crawls. This seemingly straightforward recipe yields models that generalize remarkably well to a broad range of tasks.

A closer look, however, reveals that choosing training data is not actually so straightforward. Indeed, not all data is equally useful; for example, internet data sources frequently contain "low quality" data like spam, poor writing, or nonsense text. Therefore, in practice, we tend to filter training data according to intuitive notions of quality, e.g., choosing documents similar to a "high quality" data source like Wikipedia or discarding documents with fewer than five sentences. These steps choose (qualitatively) "clean" samples that should *intuitively* improve performance. However, do such samples improve performance in practice too?

Contributions. We find that the opposite can happen: selecting data according to similarity with "high quality" data sources may not improve (and, in fact, can even hurt) model performance. Specifically, we train language models with standard, similarity-based selection methods previously used to select data for models like PaLM and GPT-3 (Brown et al., 2020; Xie et al., 2023b), and find these methods do not outperform (and can even underperform) selecting data at random (cf. Section 4).

To develop better methods for selecting training data, we start from first principles. That is, we avoid intuitive notions of data quality, and instead frame dataset selection as an optimization problem where the goal is to—given target tasks, a learning algorithm, and a candidate data pool—select the data that maximizes model performance. However, actually finding the optimal solution to this problem is difficult. While we can calculate the performance of a *specific* training set by training a model on that set (and then evaluating), it is (generally) unclear how to calculate the *best* possible training subset without examining every possible subset one by one, a computationally infeasible procedure.

We instead *approximate* the optimal subset by (approximately) modeling how the learning algorithm actually uses training data to predict. Specifically, in Section 2, we model target task performance as a function of training subset using datamodels (which efficiently approximate the mapping between training subset and model performance (Ilyas et al., 2022)), and select the subset that maximizes our estimate. Then, in Section 3, we demonstrate that our resulting method, *dataset selection with datamodels* (DSDM), consistently improves language model performance on diverse target tasks (e.g., SQuAD (Rajpurkar et al., 2016) and LAMBADA (Paperno et al., 2016)), even when existing selection methods do not.

[1]MIT. Correspondence to: Logan Engstrom <engstrom@mit.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

DSDM-selected data can improve performance on pre-specified tasks. However, in practice we train large-scale models to generalize to *yet unseen* tasks. Our framework suggests a principled approach to selecting data in this scenario too: choose target tasks similar to those we expect at deployment time, then select the optimal dataset subset for these target tasks. Following this strategy, in Section 4, we choose target tasks that cover a range of natural language problem categories (SQuAD, Jeopardy (MosaicML, 2023), and LAMBADA), and select data from C4, a canonical web crawl (Raffel et al., 2020). Our selections deliver a *2× compute multiplier* on a diverse set of test benchmarks: DSDM-selected datasets yield LMs that perform as well as those trained with 2× the compute budget on randomly selected data (we train up to 1.8B parameter models). In contrast, no baseline method outperforms randomly selecting data—even at the same compute budget.

2. Estimating the Optimal Dataset Selection

To select better data for training large-scale models, we start by defining the optimal dataset selection as an optimization problem. We then select data by finding a train subset that is *approximately* the best solution to that problem. Specifically, we use datamodels (Ilyas et al., 2022) to approximate how the learning algorithm uses data to predict on the tasks of interest. We describe the resulting framework in more detail below.

2.1. Task-Optimal Dataset Selection

We frame dataset selection as an optimization problem where the goal is to minimize trained model loss on a set of target tasks with respect to training data choice. Given a learning algorithm $\mathcal{A}$ (e.g., SGD on a neural network) that maps train set to trained model, and a target distribution $\mathcal{D}_{\text{targ}}$ (e.g., a language modeling task), the size-k *task-optimal dataset selection* over the set $\mathcal{S}$ of available data (e.g., documents from an internet scrape) is the subset

$$S^* := \operatorname*{arg\,min}_{S \subset \mathcal{S}, |S|=k} \mathcal{L}_{\mathcal{D}_{\text{targ}}}(S), \qquad (1)$$
$$\text{where } \mathcal{L}_{\mathcal{D}}(S) := \mathbb{E}_{x \sim \mathcal{D}} \left[\ell(x; \mathcal{A}(S)) \right],$$

that minimizes the trained model population loss $\mathcal{L}_{\mathcal{D}_{\text{targ}}}(S)$, where $\ell(x; g)$ denotes the loss (e.g., cross-entropy loss) for model g on example x. Note the expectation in the population loss is over both target dataset *and* learning algorithm randomness (as, e.g., SGD is a non-deterministic algorithm).

In our setting, minimizing (1) is difficult. Indeed, we do not have an easy-to-optimize, closed-form expression for trained model loss in terms of training set choice S for large-scale model learning algorithms. [1] While we can directly calculate the trained model loss for a given S by actually training on S with $\mathcal{A}$ (and then evaluating loss), using this method to find the best subset is generally computationally infeasible: we would need to train (and evaluate) a model for each of the $\binom{|\mathcal{S}|}{k}$ possible size-k train subsets.

2.2. Estimating Model Loss Efficiently with Datamodels

To circumvent this computational challenge, we trade optimality for feasibility, and instead *estimate* the best train subset. Specifically, we *approximate* the trained model loss in place of calculating it directly, then *select* the subset that minimizes our approximation.

The core primitive we use to approximate the trained model loss is datamodeling (Ilyas et al., 2022), a framework originally designed to predict how choice of training set changes model predictions. More precisely, a datamodel for a fixed sample x approximates the mapping from train subset choice S (out of the available dataset $\mathcal{S}$) to resulting trained model loss on a sample x, i.e., the function:

$$\mathcal{L}_x(S) := \mathbb{E} \left[\ell(x; \mathcal{A}(S)) \right].$$

Previous work used datamodels primarily for reliability purposes, e.g., to detect data poisoning (Khaddaj et al., 2022) or train-test leakage (Ilyas et al., 2022). In contrast, we leverage datamodels to cheaply approximate the trained model loss $\mathcal{L}_x$. Formally, given a candidate data subset $S \subset \mathcal{S}$, datamodels take as input the corresponding characteristic vector

$$\mathbb{1}_S \in \{0,1\}^{|\mathcal{S}|} \quad \text{such that} \quad (\mathbb{1}_S)_i = \begin{cases} 1 & \text{if } \mathcal{S}_i \in S \\ 0 & \text{otherwise} \end{cases}, \qquad (2)$$

instead of the subset S directly. Then, the datamodel τ_{θ_x} for x is the parameterized function that optimally predicts $\mathcal{L}_x$ over a (chosen) distribution of train subsets $\mathcal{D}_{\mathcal{S}}$, i.e.,

$$\tau_{\theta_x} : \{0,1\}^{|\mathcal{S}|} \to \mathbb{R}, \qquad \text{where}$$
$$\theta_x = \arg\min_{\theta} \widehat{\mathbb{E}}^{(m)}_{S_i \sim \mathcal{D}_{\mathcal{S}}} \left[L_{\text{reg}} \left(\tau_\theta(\mathbb{1}_{S_i}),\ \mathcal{L}_x(S_i) \right) \right], \qquad (3)$$

where $L_{\text{reg}}(\cdot, \cdot)$ is a regression loss function (e.g., mean squared error), and $\widehat{\mathbb{E}}^{(m)}$ is an m-sample empirical expectation. Note that in practice, we *estimate* the datamodel parameters that minimize (3) (i.e., we estimate the parameters of the function we use to approximate model loss).

Linear datamodels. So far we have only defined the datamodeling framework; we have not actually defined the parameterized function τ_θ or described how to estimate the

[1] Depending on the setup, we may have such a form for other classes of learning algorithms, like linear regression (with influence functions (Cook, 1977; Giordano et al., 2019)) or kernel regression (Bierens, 1988).

parameters θ. In this work, we instantiate datamodels as a *linear* function of the characteristic vector $\mathbb{1}_S$ (a standard choice (Ilyas et al., 2022; Saunshi et al., 2023)), such that

$$\tau_{\theta_x}(\mathbb{1}_S) := \theta_x^\top \mathbb{1}_S.$$

Note that, being a linear model, τ_{θ_x} treats the inclusion of an example $\mathcal{S}_i$ in the train set as having a fixed effect on $\mathcal{L}_x(S)$ irrespective of the other examples in S (this fixed effect is exactly the value of index i of θ_x).

In this work, to estimate linear datamodel parameters θ_x we largely follow the procedures of previous work (Park et al., 2023; Ilyas et al., 2022)—in particular, we use the TRAK estimator—but make changes needed for the language modeling domain (see Appendix B for full details).

2.3. DSDM: Dataset Selection with Datamodels

Recall that our goal is to estimate the candidate data subset that minimizes trained model loss on the target task (cf. (1)). To do so, we approximate the mapping between training subset S and target distribution loss $\left(\text{i.e., } \mathcal{L}_{\mathcal{D}_{\text{targ}}}(S)\right)$ with datamodels as a primitive, then select the candidate data subset that minimizes our approximation of the target loss.

Specifically, given a train subset S, we estimate the corresponding target distribution loss with an n-sample empirical expectation of datamodel loss estimates over $\mathcal{D}_{\text{targ}}$ samples:

$$\begin{aligned}\widehat{\mathcal{L}}_{\mathcal{D}_{\text{targ}}}(S) &= \widehat{\mathbb{E}}^{(n)}_{x_i \sim \mathcal{D}_{\text{targ}}}\left[\tau_{\theta_{x_i}}(\mathbb{1}_S)\right] \\ &= \frac{1}{n}\sum_{i=1}^{n} \theta_{x_i}^\top \mathbb{1}_S \\ &= \mathbb{1}_S^\top \left(\frac{1}{n}\sum_{i=1}^{n} \theta_{x_i}\right).\end{aligned}$$

Then, our size-k *dataset selection with datamodels* (DSDM) estimate of the optimal dataset selection is the subset that minimizes the approximated target loss $\widehat{\mathcal{L}}_{\mathcal{D}_{\text{targ}}}(S)$ with respect to training set choice:

$$\begin{aligned}\widehat{S}_{\text{DM}} &:= \operatorname*{arg\,min}_{S \subset \mathcal{S}, |S|=k} \widehat{\mathcal{L}}_{\mathcal{D}_{\text{targ}}}(S) \\ &= \operatorname*{arg\,min}_{S \subset \mathcal{S}, |S|=k} \mathbb{1}_S^\top \left(\frac{1}{n}\sum_{i=1}^{n} \theta_{x_i}\right) \\ &= \arg \text{bot-}k \left(\frac{1}{n}\sum_{i=1}^{n} \theta_{x_i}\right).\end{aligned}$$

In our instantiation, the considered datamodels are linear, so DSDM selects the examples corresponding to the smallest k indices of $\frac{1}{n}\sum_{i=1}^{n} \theta_{x_i}$. (Note that linear datamodels are a design choice: DSDM can use any datamodel parameterization that can be optimized over.)

3. Evaluating DSDM

To what extent does DSDM actually minimize trained model target task loss? In this section, we demonstrate that DSDM consistently reduces LM target task loss in practice. In contrast, baseline targeted dataset selection methods—all of which ignore the model training process and instead select data according to textual similarity with target task samples—often do *not* outperform randomly selecting data. Below, we describe our experimental setup, then discuss results.

3.1. Setup

To capture the effectiveness of a given data selection method, we measure the extent to which it reduces the optimal dataset selection objective of (1),

$$\mathcal{L}_{\mathcal{D}_{\text{targ}}}(S) := \mathbb{E}_{x \sim \mathcal{D}}\left[\ell(x; \mathcal{A}(S))\right],$$

across varying target tasks. For each considered target task, we split samples into a *target* set and a separate *test* set, and only use the target set to select training subsets. We then train an LM on the resulting dataset, and inspect target task performance (using the test set). Below, we describe the experimental setup as well as the baselines we use (see Appendix C for more setup details).

Target tasks, candidate dataset, and model training. We consider four separate LM target tasks: LAMBADA (Paperno et al., 2016), CS-Algorithms (Srivastava et al., 2022), SQuAD (Rajpurkar et al., 2016), and Jeopardy (Tunguz, 2019); see Appendix C.1 for more details on each task. Our candidate dataset $\mathcal{S}$ is the English subset of the Colossal Cleaned Common Crawl (C4), a standard web scrape (Raffel et al., 2020).[2] On each selected train dataset, we train a 125M parameter GPT-2 style model on 6 billion tokens.

Baselines. We compare DSDM with two standard targeted dataset selection methods, both of which select according to textual similarity between candidate training samples and $\mathcal{D}_{\text{targ}}$ samples: CLASSIFIER (selects the top examples in $\mathcal{S}$ given by a logistic model trained to classify, on FastText features, between $\mathcal{S}$ and $\mathcal{D}_{\text{targ}}$ samples; used by GPT-3/PaLM/The Pile (Chowdhery et al., 2022; Gao et al., 2020)) and DSIR (Data Selection with Importance Resampling chooses train samples with n-grams that distributionally match those of $\mathcal{D}_{\text{targ}}$ (Xie et al., 2023b)). We also compare with randomly selecting data (RANDOM).

3.2. Results

In Figure 1 we display the mean log-probability (of the label given the context, across task samples; larger is better) achieved on each target task by training a model with

[2]Each candidate example $\mathcal{S}_i$ is a sequence-length (1024 token) corpus slice; $|\mathcal{S}| \approx 217{,}000{,}000$ (cf. Appendix A.1).

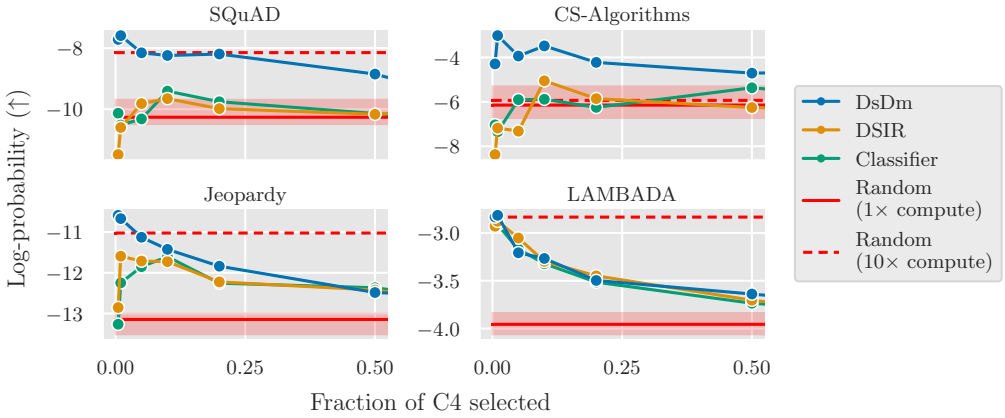

Figure 1: Target task performance by selection method, varying dataset selection size. We train a 125M models on a fixed number of tokens for each selection, adjusting epochs accordingly. DSDM consistently improves performance, even when baselines do not outperform randomly selecting data (e.g., on SQuAD and CS-Algorithms). DSDM models also consistently match a larger model trained with 10× the compute budget on random data (a Chinchilla-optimal 1.3B model). DSDM performance decreases with larger selection fraction, indicating that higher ranked DSDM samples (i.e., data in the smallest selections) tend to improve performance more than less highly ranked samples (i.e., data only present in larger selections). We measure the average log-probability of the label across samples. The "random" shaded area is the range of values achieved by 10 RANDOM models trained on one epoch of data (RANDOM performance is not x-axis dependent). Measuring accuracy in place of log-probability yields similar conclusions (cf. Figure 9).

each selection method (varying dataset selection size). Each model was trained on the same number of total tokens, with models trained on smaller fractions of C4 traversing more epochs. We find that DSDM most improves target task performance on all tasks. Models trained with DSDM even outperform a larger model trained with 10× the compute on randomly selected data. Additionally, DSDM performance decreases with larger selection fraction, indicating that the samples predicted by DSDM to most improve performance actually do so in practice. After all, smaller selections will contain more useful data (as predicted by DSDM) on average compared to larger selections (e.g., all methods select the same subset for selection fraction 1).

In contrast, baselines that select according to textual similarity with the target task, CLASSIFIER and DSIR, do *not* consistently beat randomly selecting data (e.g., on SQuAD and CS-Algorithms). These results suggest that similarity with the target task does *not* suffice to find useful samples. Note that baselines only match DSDM on LAMBADA (a passage completion task), which is also the only task without in-context instructions. We hypothesize that n-gram similarity may not capture how instructions define tasks.

To better understand how dataset choice relates to performance, we inspect the datapoints that each method is most and least likely to select (for SQuAD: in Figure 2, for all other targets: in Appendix C.3). We find that:

Useful data is not necessarily similar to the target task (or intuitively helpful at all). Looking at selected data for SQuAD in Figure 2, DSIR and CLASSIFIER select data that is more qualitatively similar to SQuAD samples (which are Wikipedia excerpts with questions, cf. Appendix Figure 5) than DSDM. Instead, DSDM samples often contain question answering-related text that does not match the SQuAD format; DSDM performance shows that qualitatively similar data is not necessarily the *best* data. However, helpful data is not always *intuitively* useful. Indeed, the DSDM examples for CS-Algorithms and Jeopardy (cf. Appendix Figures 21 and 15) often contain seemingly nonsense text. Yet, DSDM yields the best models for these tasks.

DSDM discards "mislabeled" data. Samples that DSIR and CLASSIFIER are least likely to select are qualitatively different from those of DSDM. Inspecting Appendix Figure 11 for data selected for SQuAD: least likely samples for all methods are incoherent/malformed, but those of DSDM also often contain *QA text*. Despite this, such DSDM samples examples hurt model performance: training on them is worse than selecting randomly (cf. Appendix Figure 10). We liken these samples to "mislabeled" examples from supervised learning, and conjecture excluding such data could (in part) explain DSDM performance.

4. Selecting Data for Broad Model Capabilities

So far, we have shown that DSDM consistently reduces loss on pre-specified target tasks. However, when we train large-scale models in practice our hope is that they will perform

(1) s, forms, and modification alternative can be overwhelming. So save the time, chance, money, budget, energy, also effort and implement these tips to acquire a obvious concept of what you would like and things you need before you start the quest and think about the right variations and pick right decoration, here are some recommendations and photos on deciding on the best leather sectional sofas toronto.\nThe design need to create impact to your sofa. Could it be modern, luxury, minimalist, or traditional? Co

(2) ises; soldier of fortune.\n3. a person who undertakes great commercial risk; speculator.\n4. a person who seeks power, wealth, or social rank by unscrupulous or questionable means: They thought John was an adventurer and after their daughter's money.\n"There can be adventurer souls."\n"There can be adventurer sirs."\n"There can be adventurer reflexes."\n"There can be adventurer realises."\n"There can be adventurer profiles."\n"There can be adventurer problems."\n"There can be adventurer paths."\n"There can be

(a) DSDM samples

(1) in Alexandria, where it was begun; and the Greek Bible of the Hellenistic Jews and the Catholic Church may rightly be styled the Alexandrian Greek version of the Old Testament.\nIn the early days of the Church the Septuagint was widely used among the Jews; as a rule, though there are exceptions, when the Old Testament is quoted in the New Testament it is from the Greek, not the Hebrew Bible that the quotation is made. The early Jewish-Christians and the great majority of the Jews had the same Bible, and Gent

(2) the Central Committee of the Party, that is, by the Politburo, the Orgburo (Organizational Bureau), and the Secretariat. The decisions made were implemented through the Presidium of the Supreme Soviet of the USSR, the Council of People's Commissars of the USSR, the GKO, and the General Headquarters of the Supreme Command, which had been established on August 8. Strategic direction of the armed forces was carried out by the General Headquarters through its working body, the General Staff. Major questions as

(b) DSIR samples

(1) ris and St Gleb, dating from the mid-12th century, was much rebuilt in succeeding periods, before being restored to its original shape in the 20th century. The crowning achievement of Chernigov masters was the exquisite Church of St Paraskeba (Pyatnitskaya), constructed at the turn of the 12th and 13th centuries. This graceful building was seriously damaged in the Second World War; its original medieval outlook was reconstructed. The earliest residential buildings in the downtown date from the late 17th cen

(2) their professional careers.\nDr Simpson's first line is classic.\nlatest date in the year it's been that cold in 50 years of record keeping.\nBack in March, 2007, Al Gore told Congress that "the science is settled."\nscience is settled. The Sun revolves around the Earth, not vice versa.\nscience," spent the rest of his life under house arrest.\n& Tax Bill (its actual name) through the House? Hopefully, some "cooler"\nseem, may have nothing to do with global warming.\nPaul, let me give you a little advice.\nYou migh

(c) CLASSIFIER samples

Figure 2: Samples selected by each method for SQuAD. Selected CLASSIFIER and DSIR samples are intuitively "high quality" text, and more similar to SQuAD examples (which are Wikipedia excerpts with questions) than DSDM samples are. DSDM samples do not match SQuAD, but *do* contain QA-style text, e.g., (1) left (a question in an ad) or (2) left (a dictionary definition). We display random samples from each method's selected subset (cf. Appendix C.3). "\n" is a newline.

well on *yet unseen tasks* too. Our framework suggests a straightforward approach to improving this kind of performance: choose target tasks that match those we expect to see at model deployment time, then estimate the optimal dataset selection for these "proxy" target tasks.

In this section, we demonstrate that this approach to selecting data can greatly improve held-out task performance compared to baselines. Specifically, we consider three target tasks that cover a broad range of language modeling problem categories—LAMBADA (language understanding problems), SQuAD (reading comprehension problems), and Jeopardy (world knowledge problems)—and estimate the optimal training dataset selection for these tasks (all together) via DSDM. We then compare models trained on this data with models trained via existing dataset selection baselines. Overall, evaluating on a diverse set of held-out benchmarks (meant to model "yet unseen tasks"), we find that: (a) randomly selecting data is a surprisingly strong baseline—no baseline selection method outperforms selecting data at random—and (b) our approach yields models that match those trained with 2× the training compute on randomly selected data. In particular, models trained with our approach reliably improve performance on benchmarks that are qualitatively related to the target tasks. We describe our setup below, and defer additional details to Appendix D.

Model training, scaling DSDM, selection baselines, and evaluation. We train GPT-2 style LMs with varying compute budgets. To train the best possible model for a given compute budget, we use Chinchilla-optimal parameter-to-train-tokens ratios (Hoffmann et al., 2022) and train up to 1.8B parameter models. To select with DSDM, we use 125M proxy models: we calculate DSDM subsets for 125M models, then train on these selections at each compute budget (instead of computing DSDM separately for each model class). DSDM cost scales linearly with model size, so this procedure greatly reduces overhead (cf. Appendix B.5). For baselines, we compare with two methods that select via textual similarity with a specified "high quality" data source (DSIR and CLASSIFIER, the baselines of Section 3), a data deduplication method (SemDeDup (Abbas et al., 2023)), and selecting data randomly. We evaluate on 15 standard benchmarks (cf. Table 1).

Target tasks. We execute each targeted dataset selection method using its originally proposed target task. For DSDM, we apply the framework described above: we select three target tasks that cover a broad range of LM problem categories—LAMBADA, SQuAD, and Jeopardy—then estimate the optimal dataset selection for these tasks together (i.e., $\mathcal{D}_{\text{targ}}$ as an equal mix of these tasks). For CLASSIFIER and DSIR, we target a replication of the "high quality" target distribution proposed by these methods (a mix of Wikipedia (Foundation, 2022), Books1 (Presser, 2021), and OpenWebText (Gokaslan et al., 2019), cf. Appendix D.1).

4.1. Results

In Figure 3, we display the mean benchmark performance of models trained with each selection method, varying training compute budget. Randomly selecting data is a strong baseline: all baseline methods generally match or perform *worse* than random selection across training compute budgets (Figure 3 left). In the case of CLASSIFIER and DSIR, we hypothesize that data selected via similarity with a fixed source hurts model performance by trading off data diversity for (qualitatively) "cleaner" data.

In contrast, DSDM is a 2× compute multiplier: DSDM yields

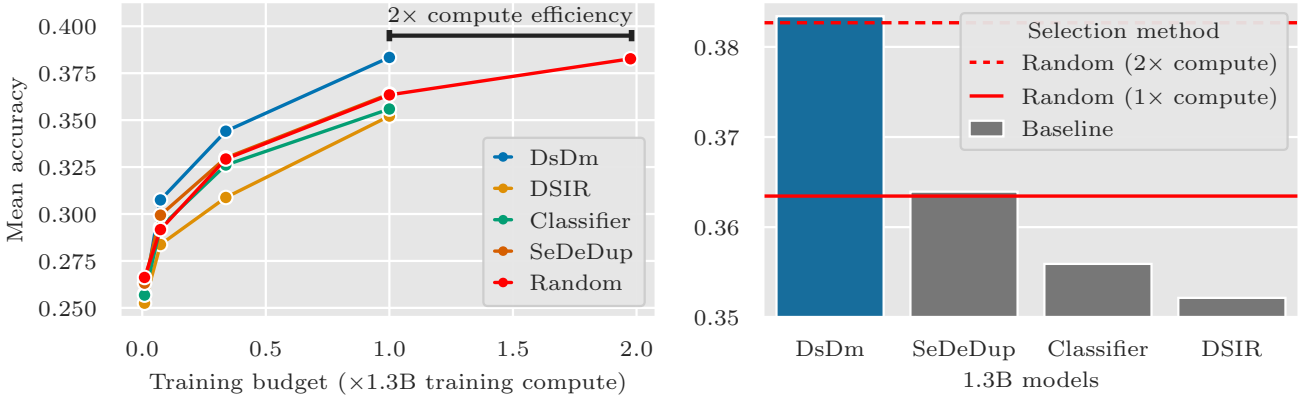

Figure 3: Left: mean benchmark performance, varying training compute budget. Right: mean performance for 1.3B models. Randomly selecting data is a strong baseline: no active selection baseline outperforms random selection. In contrast, DSDM outperforms all baselines across compute budgets (left panel), and even matches training with 2× the compute on randomly selected data (when training 1.3B models, right panel). Our train budgets correspond to training 125M, 356M, 760M, and 1.3B parameter Chinchilla-optimal LMs. To contextualize 1.3B results, we show the performance of a model trained on randomly selected data with 2× the 1.3B compute budget (i.e., a 1.8B Chinchilla-optimal model).

1.3B models that match models trained with 2× the compute budget on randomly selected data (Figure 3, right). Furthermore, across compute budgets, DSDM consistently outperforms all selection baselines (Figure 3, left).

Going beyond aggregate performance, we find that DSDM greatly improves on benchmarks related to the target tasks, while simultaneously not reducing performance on unrelated categories (on average). More precisely, inspecting individual benchmark performance in Table 1, DSDM most improves reading comprehension and world knowledge benchmarks compared to selecting randomly. We hypothesize that our choice of target tasks leads to improved performance on these benchmarks (which are qualitatively similar to SQuAD, a reading comprehension task, and Jeopardy, a world knowledge task). Furthermore, in these categories DSDM consistently matches or outperforms training with 2× the compute budget on randomly selected data (i.e., the 1.8B model in Table 1). Crucially, DSDM improves on these categories while *also* not reducing performance in other categories. As a comparison, DSIR—which targets mostly formal text—performs well on language understanding tasks but poorly on other categories (e.g., world knowledge and symbolic problem solving).

Target tasks improve performance on qualitatively similar benchmarks. So far, we have only targeted DSDM with a mix of LAMBADA, Jeopardy and SQuAD. How does target task choice change model behavior? We find that targeting a specific task generally improves performance on qualitatively related tasks. To demonstrate, in Figure 4 we display accuracy by benchmark category while varying target task across LAMBADA, Jeopardy, SQuAD, and all at once. Here, targeting a task generally improves accuracy on related tasks, e.g., SQuAD most improves reading comprehension, and Jeopardy most improves world knowledge. Furthermore, targeting all tasks at once improves overall accuracy the most. However, targeting can also decrease accuracy on unrelated tasks. For example, targeting LAMBADA, a language understanding task, reduces world knowledge accuracy compared to randomly selecting data. Our results suggest that we can tailor target tasks to improve deployment-time performance, but also that we need to be careful to choose targets that are diverse enough to capture a range of downstream problems.

DSDM is necessary to improve performance (with the targeted tasks). DSDM selections yield much better models than CLASSIFIER and DSIR selections. However, we have not yet compared these selection methods head-to-head with the same *target task*. CLASSIFIER and DSIR target a mix of "high quality" sources, while DSDM targets three LM tasks (Jeopardy, SQuAD, and LAMBADA). To what extent does selecting with DSDM drive performance compared to the difference in target tasks? We demonstrate that selecting with DSDM is necessary to improve performance on the considered target tasks. Specifically, we train models on data selected with DSIR and CLASSIFIER targeting LAMBADA, Jeopardy and SQuAD, and find that (just as when targeting "high quality text") neither outperforms randomly selecting data (cf. Appendix Figure 25).

5. Discussion

Ostensibly, the sole goal of our dataset selection framework is improve model performance by better selecting

Table 1: Accuracies on the considered benchmarks for 1.3B models trained with each selection method, along with a model trained with 2× the 1.3B compute budget on randomly selected data (a 1.8B model; Chinchilla-optimal models with larger parameter counts train with more tokens as well). In parentheses, we contextualize accuracy with the difference compared to a 1.3B model trained on randomly selected data.

		Accuracy (%)					
	Model Parameters	1.3B					1.8B
	Method	DsDm	Random	Classifier	DSIR	SeDeDup	Random
Category	Benchmark						
Commonsense Reasoning	copa	63.0 (+1)	62.0 (+0)	66.0 (+4)	67.0 (+5)	68.0 (+6)	64.0 (+2)
	openbook_qa	31.2 (–2)	33.4 (+0)	32.0 (–1)	32.0 (–1)	32.2 (–1)	33.6 (+0)
	piqa	69.0 (+0)	68.9 (+0)	69.4 (+1)	65.7 (–3)	69.7 (+1)	71.5 (+3)
Language Understanding	cbt	88.2 (+2)	86.4 (+0)	85.1 (–1)	92.4 (+6)	86.2 (+0)	88.4 (+2)
	hellaswag	42.3 (–3)	44.9 (+0)	42.7 (–2)	40.4 (–5)	44.9 (+0)	50.1 (+5)
	winogrande	51.1 (–1)	52.2 (+0)	50.5 (–2)	55.3 (+3)	50.3 (–2)	50.9 (–1)
Reading Comprehension	boolq	58.0 (+3)	54.9 (+0)	60.9 (+6)	61.0 (+6)	49.9 (–5)	53.4 (–2)
	coqa	25.5 (+7)	18.8 (+0)	16.7 (–2)	16.5 (–2)	22.9 (+4)	24.9 (+6)
	news_qa	15.6 (+8)	7.5 (+0)	5.1 (–2)	5.5 (–2)	8.6 (+1)	9.5 (+2)
Symbolic Problem Solving	bb_copy_logic	3.1 (+0)	3.1 (+0)	0.0 (–3)	0.0 (–3)	3.1 (+0)	3.1 (+0)
	bb_dyck_lang	11.9 (–2)	13.5 (+0)	3.4 (–10)	1.0 (–13)	7.3 (–6)	8.9 (–5)
	bb_operators	13.3 (+3)	10.5 (+0)	6.7 (–4)	10.5 (+0)	11.4 (+1)	9.5 (–1)
World Knowledge	arc_easy	47.6 (+3)	44.8 (+0)	44.7 (+0)	39.6 (–5)	43.5 (–1)	48.5 (+4)
	bb_qa_wikidata	48.1 (+8)	40.6 (+0)	48.3 (+8)	37.7 (–3)	45.5 (+5)	53.6 (+13)
	trivia_qa	7.1 (+3)	3.7 (+0)	2.5 (–1)	3.5 (+0)	2.4 (–1)	4.1 (+0)

training data. However, one can view our framework more broadly. That is, one can also use our framework to select data that boosts any chosen downstream property of our trained models—not just performance on a given benchmark. In this sense, our framework (and accompanying method) unlocks data curation as another stage of the model training pipeline that we can intervene on to control the downstream model behavior in a fine-grained manner. Below, we discuss in more detail the broader opportunities this view opens up as well as the other aspects of the framework, such as proxy modeling and computational efficiency.

Applications and broader opportunities. DSDM can optimize for any specified downstream model behavior. Indeed, by an appropriate choice of the target tasks, we can use our framework to improve a wide range of model behaviors, including: "aligning" models at pretraining time (in addition to or in place of existing methods, which typically operate post model training (Bai et al., 2022; Ziegler et al., 2019; Taori et al., 2023)); optimizing for notions of fairness; and improving performance on specific domains of interest (such as low-resource languages or programming).

Training stronger models with weaker proxy models. We select data for large models by using smaller models to proxy large model behavior (recall that we use DSDM to select data for smaller proxy models, then train large models on these selections). Despite that these proxy models are much worse than larger models on benchmarks (cf. Appendix Table 2), the corresponding selections nonetheless greatly improve performance. Furthermore, training on proxy models' selections is the simplest possible approach to scaling. Therefore, we suspect that scaling to larger models less naïvely could yield even better results. More broadly, our findings are in line with previous work showing that smaller models can still be leveraged to determine better training hyperparameters for larger models (Kaplan et al., 2020; Coleman et al., 2020; Hoffmann et al., 2022; Yang et al., 2022; Xie et al., 2023a).

Computational cost. DSDM is relatively inexpensive to compute in practical model training scenarios. At a high level, the most expensive part of estimating DSDM is computing the gradient for each training example on a handful of small proxy models (in our case, four 125M parameter LMs—see Appendix B.5 for a full cost breakdown). To contextualize DSDM cost with model training: computing gradients also dominates the cost of training LMs. Since the cost of computing a 125M model gradient is orders of magnitude lower than the cost of computing gradients for standard model sizes,[3] even a small compute multiplier (let alone the 2× improvement DSDM seems to offer) quickly makes the overhead of DSDM worthwhile. Additionally, after computing DSDM on a set of datapoints once, the cost

[3]For reference: models trained today generally range from 3B to 175B parameters. The cost of a gradient is (roughly) linear in model size, so it is $24\times$ to $1400\times$ more expensive to compute gradients for these models vs. 125M models.

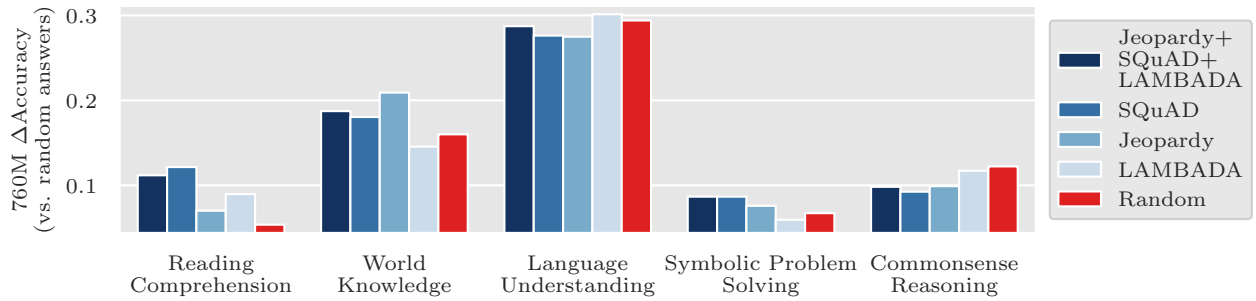

Figure 4: Per-category performance for 760M models trained with DSDM-selected data, varying target task. DSDM target tasks generally improve performance on (qualitatively) related benchmark task categories. Specifically, models targeted towards SQuAD/Jeopardy/LAMBADA improve accuracy on reading comprehension/world knowledge/language understanding, respectively. Targeting all three tasks at once improves overall accuracy. However, target tasks can also reduce performance on (qualitatively) unrelated tasks. For example, targeting with LAMBADA (a language understanding task) reduces performance on world knowledge tasks compared to randomly selecting. See each category's constituent benchmarks in Table 1. We plot improvement compared to randomly guessing answers (e.g., some benchmarks are multiple-choice).

of computing DSDM on those datapoints again is essentially negligible (as the required computations are easy to cache). Therefore, we can amortize DSDM's computational cost over the entire "lifetime" of training on the given dataset.

6. Related Work

Current methods for selecting LM pretraining datasets tend to follow a two-step framework: (a) choose an intuitively "high quality" reference corpus, like Wikipedia (Foundation, 2022), then (b) select data that matches it. There are two standard methods that adhere to this framework: DSIR (Dataset Selection with Importance Reweighting (Xie et al., 2023b)) and CLASSIFIER (originally introduced in Brown et al. (2020) and used by other work (Gao et al., 2020; Chowdhery et al., 2022; Du et al., 2022)). Other work on selecting data for LM pretraining has included deduplicating examples in LM activation space (Abbas et al., 2023), and selecting examples with the largest difference in loss between LMs trained on the candidate and reference sets (Moore and Lewis, 2010; Axelrod, 2017; Feng et al., 2022). Simpler methods for selecting data are also commonplace. These include removing documents that are too short or contain too many special characters (Raffel et al., 2020; Computer, 2023; Xie et al., 2023b). In the LM domain, a related (but different) task to dataset selection is choosing weights for sampling from mixtures of data sources (Chen et al., 2023b; Xie et al., 2023a; Albalak et al., 2023).

Beyond LM pre-training, previous work also selects data in other domains. These works aim to: improve the performance on a given task (Wei et al., 2015; Kaushal et al., 2019; Wang et al., 2020; Killamsetty et al., 2021a; Chitta et al., 2021; Mindermann et al., 2022), identify core-sets of large training datasets (Sener and Savarese, 2017; Phillips, 2017; Coleman et al., 2020; Mirzasoleiman et al., 2020; Paul et al., 2021; Killamsetty et al., 2021b; Okanovic et al., 2023), and fine-tune LMs (Antonello et al., 2022; Chen et al., 2023a; Cao et al., 2023). Broadly, such methods select by prompting pretrained models, discriminating on proxies for model uncertainty like loss or gradient norm, matching on gradients, or deduplicating in model output space.

DSDM uses TRAK to estimate datamodel weights, which calculates influences using model gradients. We therefore additionally overview gradient-based methods for LM dataset selection in machine learning more broadly (Bejan et al., 2023; Wang et al., 2023; Xia et al., 2024). These methods estimate the effect of including a given train sample on a given test example by calculating the inner product between the two examples' gradients. Through this lens, all methods can be seen as applying a variant of TracIn (Pruthi et al., 2020) to compute influences that are then used to select data. In comparison, TRAK estimates the effect of including a given train sample on a given test sample by calculating influences on a linearized version of the model of interest (cf. Appendix B.2 for more details).

7. Conclusion

In this work, we cast dataset selection as an optimization problem: given target tasks, a learning algorithm, and a candidate training dataset, choose the training maximizes performance. We then propose a method for approximating the solution to this optimization problem, DSDM, that selects by modeling how the learning algorithm uses training data to predict on the target tasks. We show that our method reliably improves target task performance in the LM setting, and furthermore use our framework to improve broader model generalization. By choosing target tasks similar to those we expect to see at deployment time, we can greatly improve model performance on yet unseen tasks.

Our findings prompt us to take on a much broader view of the role of dataset selection stage in model training. In particular, our framework demonstrates that dataset selection can be an effective tool for fine-grain control of model behavior. Indeed, we hypothesize that carefully choosing data can not only improve downstream task performance, but also other downstream properties of trained models, such as notions of predictor fairness, alignment with human preferences, or capabilities in specific domains like low-resource languages or programming. We also suspect that current methods for datamodeling only scratch the surface of understanding how models learn from data—and that we can greatly improve our ability to manipulate model behavior through training data by developing better datamodeling techniques.

Impact Statement

We introduce a new method for selecting data for improving model performance. Narrowly considering the direct use of our method, selecting certain data can unpredictably change fine-grained notions of model behavior. Therefore, as a method for selecting data, DSDM data could cause unintended changes to model behavior. More broadly, DSDM is a method meant to improve machine learning models, and there are many potential consequences associated with advancing machine learning.

References

Amro Abbas, Kushal Tirumala, Dániel Simig, Surya Ganguli, and Ari S Morcos. Semdedup: Data-efficient learning at web-scale through semantic deduplication. *arXiv preprint arXiv:2303.09540*, 2023.

Alon Albalak, Liangming Pan, Colin Raffel, and William Yang Wang. Efficient online data mixing for language model pre-training. *Training*, 20000(40000): 60000, 2023.

Richard Antonello, Nicole Beckage, Javier Turek, and Alexander Huth. Selecting informative contexts improves language model finetuning, 2022.

Amittai Axelrod. Cynical selection of language model training data. *arXiv preprint arXiv:1709.02279*, 2017.

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

Irina Bejan, Artem Sokolov, and Katja Filippova. Make every example count: On the stability and utility of self-influence for learning from noisy nlp datasets. *arXiv preprint arXiv:2302.13959*, 2023.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR, 2023.

Hermanus Josephus Bierens. The nadaraya-watson kernel regression function estimator. 1988.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language, 2019.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. Gpt-neox-20b: An open-source autoregressive language model, 2022. URL https://arxiv.org/abs/2204.06745.

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

Yihan Cao, Yanbin Kang, and Lichao Sun. Instruction mining: High-quality instruction data selection for large language models. *arXiv preprint arXiv:2307.06290*, 2023.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, et al. Alpagasus: Training a better alpaca with fewer data. *arXiv preprint arXiv:2307.08701*, 2023a.

Mayee F Chen, Nicholas Roberts, Kush Bhatia, Jue Wang, Ce Zhang, Frederic Sala, and Christopher Ré. Skill-it! a data-driven skills framework for understanding and training language models. *arXiv preprint arXiv:2307.14430*, 2023b.

Kashyap Chitta, José M Álvarez, Elmar Haussmann, and Clément Farabet. Training data subset search with ensemble active learning. *IEEE Transactions on Intelligent Transportation Systems*, 23(9):14741–14752, 2021.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions, 2019.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*, 2018.

Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. Selection via proxy: Efficient data selection for deep learning. In *International Conference on Learning Representations (ICLR)*, 2020.

Together Computer. Redpajama: an open dataset for training large language models. https://github.com/togethercomputer/RedPajama-Data, October 2023.

R Dennis Cook. Detection of influential observation in linear regression. *Technometrics*, 19(1):15–18, 1977.

Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. Glam: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning*, pages 5547–5569. PMLR, 2022.

Yukun Feng, Patrick Xia, Benjamin Van Durme, and João Sedoc. Automatic document selection for efficient encoder pretraining. *arXiv preprint arXiv:2210.10951*, 2022.

Wikimedia Foundation. English wikipedia. https://huggingface.co/datasets/wikipedia, 2022.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Ryan Giordano, William Stephenson, Runjing Liu, Michael Jordan, and Tamara Broderick. A swiss army infinitesimal jackknife. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1139–1147. PMLR, 2019.

Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. Openwebtext corpus. http://Skylion007.github.io/OpenWebTextCorpus, 2019.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. The goldilocks principle: Reading children's books with explicit memory representations. *arXiv preprint arXiv:1511.02301*, 2015.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. In *arXiv preprint arXiv:2203.15556*, 2022.

Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. Datamodels: Predicting predictions from training data. In *International Conference on Machine Learning (ICML)*, 2022.

William B Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. In *Contemporary mathematics*, 1984.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Vishal Kaushal, Rishabh Iyer, Suraj Kothawade, Rohan Mahadev, Khoshrav Doctor, and Ganesh Ramakrishnan. Learning from less data: A unified data subset selection and active learning framework for computer vision. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1289–1299. IEEE, 2019.

Alaa Khaddaj, Guillaume Leclerc, Aleksandar Makelov, Kristian Georgiev, Andrew Ilyas, Hadi Salman, and Aleksander Madry. Backdoor or feature? a new perspective on data poisoning. 2022.

Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh Iyer. Glister: Generalization based data subset selection for efficient and robust learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8110–8118, 2021a.

Krishnateja Killamsetty, Xujiang Zhao, Feng Chen, and Rishabh Iyer. Retrieve: Coreset selection for efficient and robust semi-supervised learning. *Advances in Neural Information Processing Systems*, 34:14488–14501, 2021b.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, 2017.

Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating wikipedia by summarizing long sequences. *arXiv preprint arXiv:1801.10198*, 2018.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*, 2018.

Sören Mindermann, Jan M Brauner, Muhammed T Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Höltgen, Aidan N Gomez, Adrien Morisot, Sebastian Farquhar, et al. Prioritized training on points that are learnable, worth learning, and not yet learnt. In *International Conference on Machine Learning*, pages 15630–15649. PMLR, 2022.

Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *International Conference on Machine Learning*, pages 6950–6960. PMLR, 2020.

Robert C Moore and William Lewis. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 conference short papers*, pages 220–224, 2010.

MosaicML. Composer, 2021. URL https://www.github.com/mosaicml/composer.

MosaicML. LLM Foundry, 2023. URL https://www.github.com/mosaicml/llm-foundry.

Patrik Okanovic, Roger Waleffe, Vasilis Mageirakos, Konstantinos E Nikolakakis, Amin Karbasi, Dionysis Kalogerias, Nezihe Merve Gürel, and Theodoros Rekatsinas. Repeated random sampling for minimizing the time-to-accuracy of learning. *arXiv preprint arXiv:2305.18424*, 2023.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The lambada dataset: Word prediction requiring a broad discourse context. *arXiv preprint arXiv:1606.06031*, 2016.

Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. Trak: Attributing model behavior at scale. In *Arxiv preprint arXiv:2303.14186*, 2023.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. *Advances in Neural Information Processing Systems*, 34:20596–20607, 2021.

Jeff M Phillips. Coresets and sketches. In *Handbook of discrete and computational geometry*, pages 1269–1288. Chapman and Hall/CRC, 2017.

Daryl Pregibon. Logistic regression diagnostics. In *The Annals of Statistics*, 1981.

Shawn Presser. Bookcorpusopen. https://huggingface.co/datasets/bookcorpusopen, 2021.

Garima Pruthi, Frederick Liu, Mukund Sundararajan, and Satyen Kale. Estimating training data influence by tracing gradient descent. In *Neural Information Processing Systems (NeurIPS)*, 2020.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research (JMLR)*, 2020.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

Siva Reddy, Danqi Chen, and Christopher D. Manning. Coqa: A conversational question answering challenge, 2019.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI Spring Symposium Series*, 2011.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd

schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Nikunj Saunshi, Arushi Gupta, Mark Braverman, and Sanjeev Arora. Understanding influence functions and datamodels via harmonic analysis. In *ICLR*, 2023.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017.

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

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. Newsqa: A machine comprehension dataset. *arXiv preprint arXiv:1611.09830*, 2016.

Bojan Tunguz. Jeopardy! questions, 2019. URL https://www.kaggle.com/datasets/tunguz/200000-jeopardy-questions.

Xiao Wang, Weikang Zhou, Qi Zhang, Jie Zhou, Songyang Gao, Junzhe Wang, Menghan Zhang, Xiang Gao, Yunwen Chen, and Tao Gui. Farewell to aimless large-scale pretraining: Influential subset selection for language model. *arXiv preprint arXiv:2305.12816*, 2023.

Xinyi Wang, Hieu Pham, Paul Michel, Antonios Anastasopoulos, Jaime Carbonell, and Graham Neubig. Optimizing data usage via differentiable rewards. In *International Conference on Machine Learning*, pages 9983–9995. PMLR, 2020.

Kai Wei, Rishabh Iyer, and Jeff Bilmes. Submodularity in data subset selection and active learning. In *International conference on machine learning*, pages 1954–1963. PMLR, 2015.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. Less: Selecting influential data for targeted instruction tuning, 2024.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. Doremi: Optimizing data mixtures speeds up language model pretraining. *arXiv preprint arXiv:2305.10429*, 2023a.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy Liang. Data selection for language models via importance resampling. *arXiv preprint arXiv:2302.03169*, 2023b.

Greg Yang, Edward J. Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tensor programs v: Tuning large neural networks via zero-shot hyperparameter transfer. *arXiv preprint arXiv:2203.03466*, 2022.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

Yin Zhang, Rong Jin, and Zhi-Hua Zhou. Understanding bag-of-words model: a statistical framework. *International journal of machine learning and cybernetics*, 2010.

Ji Zhu and Trevor Hastie. Kernel logistic regression and the import vector machine. *Journal of Computational and Graphical Statistics*, pages 185–205, 2005.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

Appendices

Contents

A. Experimental Setup

In this section we discuss general experimental setup, including candidate data pool, considered target tasks, baselines, evaluation metrics, and model training choices.

A.1. Candidate dataset

Our candidate dataset is the full English subset of C4 (Raffel et al., 2020). We use the train split of the `en.noblocklist` subset of the C4 version prepared by AllenAI at `https://huggingface.co/datasets/c4`. The subset name `noblocklist` signifies that curse words were not filtered in the subset.

To split the text from the documents into examples, we tokenize all the documents, concatenate them together (separated by end-of-text tokens), and then slice the result into 1024 token chunks. These 1024 token examples generally contain between 3,000 and 6,000 characters (roughly a thousand words). The final candidate dataset has 216,948,746 examples. We tokenize with the Pythia tokenizer (Black et al., 2022; Biderman et al., 2023).

As a public internet crawl, C4 contains diverse text. To contextualize the dataset, we show (excerpts) of random C4 samples in Figure 24.

A.2. Target tasks

We describe each of the considered target tasks below. We both describe the tasks, and how we split samples into distinct sets of "target samples" (to select datasets for a target task) and "holdout samples" (to evaluate models on the target task):

- **SQuAD.** The Stanford Question-Answering Dataset (SQuAD (Rajpurkar et al., 2016)) is an open book, reading comprehension dataset of questions about Wikipedia articles. The goal is to answer questions using the corresponding article as context. Our target set is 25% of the SQuAD train set (23107 examples), our holdout set is the SQuAD validation set (10557 examples).
- **Jeopardy.** Jeopardy (Tunguz, 2019) is a set of trivia questions taken directly from the show "Jeopardy!" We use the version of Jeopardy published by MosaicML (MosaicML, 2023).[4] We include all the samples save for the "Word Origins" subset.[5] We randomly partition the remaining samples into 876 target samples and 876 holdout samples.
- **LAMBADA.** LAnguage Modeling Broadened to Account for Discourse Aspects (LAMBADA (Paperno et al., 2016)) is an open-ended cloze task measuring broad context text understanding. The goal is to predict the last word of curated passages from BooksCorpus (Zhu et al., 2015) given the rest of the passage as context. The task is meant to be challenging: Paperno et al. (2016) only select passages such that crowdworkers could not guess the final word given the final sentence alone (up until the final word), but could guess the final word given the entire passage. We use the LAMBADA version curated by EleutherAI.[6] Finally, we split the LAMBADA test set into separate target and holdout sets, then remove 6 samples from the LAMBADA holdout set due to overlap with samples in our candidate train dataset (cf. Subsection A.2.1 for details on this procedure). We conclude with 2570 holdout samples and 2577 target samples.
- **CS-Algorithms.** BIG-bench CS Algorithms (Srivastava et al., 2022) measures the ability of models to solve basic algorithmic problems. In particular, this benchmark contains two kinds of problems: testing for balanced parentheses, and finding the longest common subsequence of multiple strings. For each considered example, the goal is to directly output the answer to the posed algorithmic question. We randomly split the test set into 660 target samples and 660 holdout samples.

We include samples of each benchmark in Figure 5 (SQuAD), Figure 6 (Jeopardy), Figure 7 (LAMBADA), and Figure 8 (CS-Algorithms). We evaluate in the 0-shot (for LAMBADA and CS-Algorithms) and 3-shot (for SQuAD and Jeopardy) regimes. In the 3-shot setting, we separate each example with a single newline. We use standard prompts for each task (see the samples for details).

[4]Located at: `https://github.com/mosaicml/llm-foundry/blob/v0.2.0/scripts/eval/local_data/world_knowledge/jeopardy_all.jsonl`

[5]We originally intended this subset as a hold-out set for our broader evaluation, but decided not to use the subset as we deemed it unfairly close to the original task to serve as a true hold-out set.

[6]Located at `https://huggingface.co/datasets/EleutherAI/lambada_openai/viewer/en`.

1. Context: The chloroplasts of some hornworts and algae contain structures called pyrenoids. They are not found in higher plants. Pyrenoids are roughly spherical and highly refractive bodies which are a site of starch accumulation in plants that contain them. They consist of a matrix opaque to electrons, surrounded by two hemispherical starch plates. The starch is accumulated as the pyrenoids mature. In algae with carbon concentrating mechanisms, the enzyme rubisco is found in the pyrenoids. Starch can also accumulate around the pyrenoids when CO2 is scarce. Pyrenoids can divide to form new pyrenoids, or be produced "de novo".
Question: What shape are pyrenoids?
Answer: roughly spherical

2. Context: In this dioxygen, the two oxygen atoms are chemically bonded to each other. The bond can be variously described based on level of theory, but is reasonably and simply described as a covalent double bond that results from the filling of molecular orbitals formed from the atomic orbitals of the individual oxygen atoms, the filling of which results in a bond order of two. More specifically, the double bond is the result of sequential, low-to-high energy, or Aufbau, filling of orbitals, and the resulting cancellation of contributions from the 2s electrons, after sequential filling of the low σ and σ^* orbitals; σ overlap of the two atomic $2p$ orbitals that lie along the O-O molecular axis and π overlap of two pairs of atomic $2p$ orbitals perpendicular to the O-O molecular axis, and then cancellation of contributions from the remaining two of the six 2p electrons after their partial filling of the lowest π and π^* orbitals.
Question: What is a descriptive term for a low-to-high energy bond?
Answer: Aufbau

Figure 5: Random **SQuAD** samples. Context is normal text, and the continuation label is hightlighted.

1. WORLD HISTORY: In 1191 this Lion-Hearted king of England captured Cyprus & Acre during the Crusades
Answer: Richard I

2. LITERATURE: 1719 novel about a mariner who lived 8 & 20 years all alone in an uninhabited island
Answer: Robinson Crusoe

Figure 6: Random **Jeopardy** samples. Context is normal text, and the continuation label is hightlighted.

1. The Simplification Movement wasn't really an organized movement. It was more of an ideological shift by a large number of believers. There were quite a few Simpletons among the Mother Assembly denomination, but the High Sire had never recognized their movement as an order or organization. However, some other denominations were founded on the principles of the Simplification Movement

2. "Here," said Jacob, handing them what was a rope attached to the ground next to them, the other end at the bottom of the well. "You first."
Will stood there. Why am I doing this? he thought.
"Come on, let's go!" ordered Jacob.
Will took the rope and began to climb down the well.
"Thatta boy, you've got this," said Jacob

Figure 7: Random **LAMBADA** samples. We show the context as normal text, and the continuation label as hightlighted.

1. Given two strings, determine the length of the longest common subsequence.

   Strings: REFVJLZIV PJIQB
   Length of longest common subsequence: 2

2. Determine whether the given sequence of parentheses is properly matched.

   Sequence: [ ] ( ) ((( )))
   Valid/Invalid? Valid

Figure 8: Random **CS-Algorithms** samples. We show the context as normal text, and the continuation label as hightlighted.

A.2.1. MITIGATING TRAIN-TEST LEAKAGE

We mitigate train-test leakage by filtering out test examples that overlap with our candidate data samples. Specifically, we define a test example as "leaked" if both its context and continuation are present in a single C4 example. To upper-bound train-test leakage, we test for the context and continuation separately (i.e., for a given test sample, the context and continuation do not have to be contiguous in a train sample to count as leaked) We investigate train-test leakage for all the test examples in each of the test sets (i.e., LAMBADA, SQuAD, Jeopardy, and CS-Algorithms) across the entire candidate train set (i.e., the C4 English subset). Note that we match strings after lowercasing and removing whitespace.

We find 6 LAMBADA test examples with overlap in C4, and remove them from our LAMBADA test split. We do not find any train-test leakage for SQuAD, Jeopardy, or CS-Algorithms.

A.3. Data selection baselines

We consider four baselines for selecting language modeling data. These fall into two categories: *targeted* data selection methods (which select data according to a target distribution), and *untargeted* data selection methods (which do not take in a target distribution).

A.3.1. TARGETED BASELINES

The two targeted dataset selection methods we consider, CLASSIFIER (originally used to select the GPT-3 dataset (Brown et al., 2020)) and DSIR, both select according to textual similarity with a target distribution. We describe the details of these methods below:

CLASSIFIER. The dataset selection method originally developed to select data for GPT-3, and additionally used to select data for PaLM (Chowdhery et al., 2022) and The Pile (Gao et al., 2020). The method trains a logistic regression model on FastText (Joulin et al., 2016) features to classify between (held-out) samples of the candidate dataset (in our case, C4) and the target distribution, then chooses training data according to how likely the model predicts the data as being sampled from the target distribution. To more specifically describe CLASSIFIER: the method keeps a given document if the scored document satisfies:

$$\epsilon > 1 - \texttt{document\_score}, \epsilon \sim \text{Lomax}(\alpha),$$

where a Lomax sample is drawn for each considered document, and where `document_score` is the classifier-given probability that the given sample is in the target distribution. Sampling a threshold according to the Lomax distribution is meant to improve diversity of the selected data. In this work, we learn the classifier on the C4 `en.noblocklist` validation set, and choose $\alpha = 12$ via the parameter selection procedure described in Brown et al. (2020) (score each document in C4 with the classifier, then fit the parameters of a Lomax distribution via maximum likelihood estimation according to these scores).

DSIR. Dataset Selection with Importance Resampling (Xie et al., 2023b) aims to select a data subset with a similar distribution as the target task in terms of n-gram counts. DSIR comprises two steps: (a) find the (hashed) n-gram counts for each train set example (each example is represented as a vector of counts, with n-grams hashed into buckets to reduce dimensionality), then (b) importance sample to select candidate train set examples that are distributed similarly to target distribution samples in terms of n-gram counts. DSIR calculates importance weights by modeling the distribution of examples (in feature space) under the target distribution and under the candidate data distribution separately, using bag-of-words style models. In greater detail, DSIR consists of the following steps:

1. Fit $\hat{p}_{\text{feat}}$ and $\hat{q}_{\text{feat}}$, estimates of the distributions of target examples and candidate training examples in hashed n-gram space (respectively). DSIR parameterizes $\hat{p}_{\text{feat}}$ and q_{feat} through the following general procedure for estimating the distribution of hashed n-grams[7] for a given set of documents. First, calculate the hashed n-gram counts (with d hash buckets) across the documents as the vector $\gamma \in \mathbb{R}^d$, where γ_k corresponds to the number of n-grams that hash to k in the documents. Then, normalize γ so that its values sum to 1, forming a probability distribution over buckets. Finally, parameterize the distribution of hashed n-grams for this set of documents as a bag-of-words style model (Zhang et al.,

[7]For example, if we wanted to make a d dimensional hashed n-gram feature vector for a document, we would find all the n-grams in the document, hash the n-grams into integers up to size d, then go through each integer and increment the corresponding feature vector index.

2010) such that the probability of a document with hashed n-gram counts c is $\prod_{i=1}^{d} \gamma_d^{c_i}$ (here, the bag-of-words model is over hashed n-grams instead of words).

2. Calculate importance weights for each example in the candidate training set, such that example i with counts c has weight $w_i = \frac{\hat{p}_{\text{feat}}(c)}{\hat{q}_{\text{feat}}(c)}$.
3. Sample examples without replacement according to the categorical distribution with (unscaled) weights w_i.

For more details on DSIR, see Section 4 of Xie et al. (2023b). We adapt implementations of both DSIR and CLASSIFIER from https://github.com/p-lambda/dsir.

Considered target distributions. We apply targeted dataset selection methods with different target distributions depending on the context. In Section 3, we measure the extent to which different selection methods can reduce loss on individual target tasks, so we select data for *individual tasks* (i.e., Jeopardy, SQuAD, CS-Algorithms, and LAMBADA). In Section 4 we use these targeted baselines to select data for general purpose language modeling, so we use the recommended target task from each work (intuitively high-quality data sources; see Appendix D.1 for more details).

A.3.2. UNTARGETED BASELINES

The two untargeted dataset selection methods we consider are: RANDOM (select data randomly) and SemDeDup (Semantic Deduplication (Abbas et al., 2023)). SemDeDup selects by clustering data according to the last layer activations for the last token in the given document, then choosing only the examples in each cluster that have the lowest cosine similarity with the cluster centroid. We follow the hyperparameters from the original work (11,000 clusters, deduplicating down to 20% of the dataset for optimal model performance). We use the implementation from https://github.com/facebookresearch/SemDeDup/.

A.4. LM training details

We train GPT-2 family decoder-only transformer models (Radford et al., 2019; Liu et al., 2018) using LLM-Foundry (MosaicML, 2023). To train models, we use ADAM ($\beta_1 = 0.9, \beta_2 = 0.95, \epsilon = 10^{-8}$), sequence length 1024, batch size 1024, a cosine learning rate schedule (with 200 warm up batches and $\alpha = 0.1$), and ℓ_2 gradient clipping with threshold 1. We train on A100s (with BF16 precision) and H100s (with FP8 precision), and tokenize text with the BPE tokenizer used by Pythia (Biderman et al., 2023).

We summarize the remaining hyperparameter choices used to train the models in this work in Table 2 (including weight decay, learning rate, model architecture, and training token count). We select all hyperparameters to minimize 125M held-out perplexity on C4. The only exception: we increase the weight decay for the Section 4 models to ensure that larger parameter model training runs converge (with smaller weight decay, larger models diverge in loss). Model parameterization choices (i.e., number of heads or layers), optimizer hyperparameters, and learning rate schedule generally chosen according to the default LM training configurations in LLM-Foundry.

Chinchilla-optimal compute ratios. To train the best possible LM for a given compute budget, one must trade off two hyperparameters that control used compute: model size and number of training tokens. We use Chinchilla-optimal parameter-to-training-token ratios to trade these parameters off (Hoffmann et al., 2022). In our compute regime, this (roughly) amounts to training on a number of tokens equal to 20× the number of parameters.

A.5. Evaluation metrics

In this work, we measure model performance using two different metrics: log-probability (in Section 3, to compare model performance on target tasks) and accuracy (in Section 4, to compare model performance on a broad set of yet unseen tasks). Below, we describe how we measure both metrics.

A.5.1. LOG-PROBABILITY

To calculate mean log-probability, we compute the log-probability of the model generating the correct label, then aggregate the mean across benchmark samples. More specifically, all the tasks we evaluate with log-probability are open-ended LM tasks (e.g., LAMBADA), where the goal is to generate a desired continuation from the context (e.g., for LAMBADA,

Table 2: Training configurations for models trained across this work. Accuracy measured as the mean accuracy across the benchmarks considered in Section 4 for a model trained with *randomly selected* data with the corresponding configuration. The Chinchilla-optimal (760M, 1.3B, 1.8B) models (Hoffmann et al., 2022) of Section 4 are much more accurate than the 125M models used to calculate datamodels. Following previous work, we approximate FLOPs (Floating Point OPerations) via `parameters` × `tokens` × 6 (Kaplan et al., 2020; Hoffmann et al., 2022); FLOPs proxy the computational cost of training a given model. LR is learning rate, WD is weight decay. Each batch contains 1024 samples of 1024 tokens each.

	Hyperparameters								
Parameters	LR	WD	d_{model}	Heads	Layers	Tokens	Batches	Train FLOPs	Accuracy
Estimating datamodels									
125M	6×10^{-4}	2×10^{-4}	768	12	12	8.4×10^{10}	80000	6.3×10^{19}	31.8%
Section 3: Evaluating optimal dataset selection estimators									
125M	6×10^{-4}	2×10^{-4}	768	12	12	2.6×10^{10}	25000	2.0×10^{19}	–
Section 4: Evaluating unseen-task generalization (chosen as ~Chinchilla-optimal)									
125M	6×10^{-4}	4×10^{-4}	768	12	12	2.5×10^{9}	2400	1.9×10^{18}	26.6%
356M	6×10^{-4}	4×10^{-4}	1024	16	24	7.0×10^{9}	6700	1.5×10^{19}	29.2%
760M	6×10^{-4}	4×10^{-4}	1536	12	24	1.5×10^{10}	14400	6.9×10^{19}	32.9%
1.3B	6×10^{-4}	4×10^{-4}	2048	16	24	2.6×10^{10}	24700	2.0×10^{20}	36.3%
1.8B	6×10^{-4}	4×10^{-4}	2432	19	24	3.7×10^{10}	34931	4.0×10^{20}	38.3%

generate the last word of a paragraph, given the rest of the paragraph as context). Therefore, the log-probability of the model answering correctly is the log-probability that the model generates the label, given the context. This is, for a sample x with k continuation tokens starting at index C,

$$\text{Log\_Probability}(x; g_w) = \sum_{i=C}^{C+k} \log(p_i), \text{ where } p_i \text{ is the correct-label probability given by model } g_w \text{ at index } i. \quad (4)$$

A.5.2. ACCURACY

To evaluate accuracy, we use one of three separate accuracy procedures depending on the considered benchmark: (a) multiple choice accuracy, (b) exact text match, or (c) fuzzy text match. These are:

- **Multiple choice accuracy**: For multiple choice question benchmarks, we choose the answer with the maximal predicted probability out of the possible choices, then measure the accuracy as the fraction of correct answers.
- **Exact match**: We mark an example as correct if the generated tokens for the context exactly match the label tokens, then measure the accuracy as the fraction of correct answers.
- **Fuzzy match**: For open-ended benchmarks like TriviaQA whose questions have multiple textually different but correct answers, we measure whether our model is correct on a given example through the following procedure. We generate text for the example context, then normalize this text with the standard TriviaQA text normalizer[8] (which removes articles/extraneous white space/punctuation and normalizes underscores/casing), and finally count the example as correct if the resulting normalized text exactly matches any of the (normalized) labels. We then measure accuracy as the fraction of correct answers.

Table 4 lists the exact accuracy procedure used for each considered benchmark.

[8]Default choice for this procedure accuracy measurement in the MosaicML Composer (MosaicML, 2021), see https://github.com/mandarjoshi90/triviaqa/blob/master/evaluation/triviaqa_evaluation.py

B. Datamodel estimation

In this section, we describe how we estimate datamodels for GPT-2 style LMs. We start by briefly giving an overview of datamodels (cf. Appendix B.1), then describe the datamodel estimator we use, TRAK (cf. Appendix B.2). Finally, we conclude by instantiating datamodels for language modeling (cf. Appendix B.3), and analyzing the computational cost of our procedure (cf. Appendix B.5). For the impatient reader, we include a standalone section on how to mechanically compute datamodel estimates with TRAK (without background) in Appendix B.4.

B.1. Datamodels refresher

The goal of datamodeling is to approximate the mapping from choice of training subset to trained model loss on a given, fixed sample. Datamodels frame this problem as a supervised learning problem: datamodels *learn* an approximation from the former to the latter. Recall from Section 2.2 that the datamodel τ_θ for an example x is a parameterized function that, given a candidate training dataset $\mathcal{S}$, learning algorithm $\mathcal{A}$ (mapping train set to trained model), and model output function f (in the main text, we simplify by refering to this quantity as the loss ℓ; but in reality f can capture any function of the trained model) that maps test example and model to resulting loss, optimally predicts the model output on x over a (chosen) distribution of train subsets $\mathcal{D}_\mathcal{S}$, i.e.,

$$\tau_{\theta_x} : \{0,1\}^{|\mathcal{S}|} \to \mathbb{R}, \qquad \text{where} \qquad \theta_x = \arg\min_\theta \widehat{\mathbb{E}}^{(m)}_{S_i \sim \mathcal{D}_\mathcal{S}} \left[L_{\text{reg}} \left(\tau_\theta(\mathbb{1}_{S_i}),\ f(x; \mathcal{A}(S)) \right) \right], \tag{5}$$

where $L_{\text{reg}}(\cdot, \cdot)$ is a regression loss function (e.g., mean squared error), and $\widehat{\mathbb{E}}^{(m)}$ is an m-sample empirical expectation. Note that datamodels operate on the characteristic vector $\mathbb{1}_S$ of each subset (cf. Equation 2), not the subset directly.

In this work, we parameterize τ_{θ_x} as *linear* in the choice of training data, i.e., such that

$$\tau_{\theta_x}(\mathbb{1}_S) = \mathbb{1}_S^\top \theta_x.$$

Intuitively, such linear datamodels model each datapoint $\mathcal{S}_i$ as having a constant effect on the loss when included in the training set (this effect is exactly the value of θ_x in index i).

B.1.1. Estimating datamodels with data regression

So far we have only defined linear datamodels. How do we actually estimate the linear parameters θ_x? When introducing datamodels, Ilyas et al. (2022) originally did so with a linear regression predicting loss from training subset—i.e., directly minimizing Equation 5 by collecting a large amount of "training data"—pairs of (randomly chosen training data subset, corresponding trained model output on x)—then learning the mapping from train subset to output on the collected training data.

This estimator, which we refer to as *data regression*, proceeds in two steps. The first step is to collect regression data. Here, we repeatedly: sample a random train subset S_i (from a chosen distribution $S_i \sim \mathcal{D}_S$[9]), train a model $\mathcal{A}(S_i)$ on the subset, then evaluate the model output on x (and record the train subset, model output pairs). This step yields "training data" for the regression in the form of m train subset, loss pairs: $\{(\mathbb{1}_{S_i}, \ell(x; \mathcal{A}(S_i)))\}_{i=1}^m$ (recall that our datamodel takes as input the characteristic vector of subsets rather than subsets directly). Then, the second step is to actually estimate the linear datamodel parameters with linear regression. Here, the regression minimizes the (empirical) squared error over datamodel parameters:

$$\begin{aligned} \theta_x &= \arg\min_\theta \widehat{\mathbb{E}}^{(m)}_{S_i \sim \mathcal{D}_\mathcal{S}} \left[L_{\text{reg}} \left(\tau_\theta(\mathbb{1}_{S_i}),\ \ell(x; \mathcal{A}(S)) \right) \right] \\ &= \arg\min_\theta \widehat{\mathbb{E}}^{(m)}_{S_i \sim \mathcal{D}_\mathcal{S}} \left[\left(\mathbb{1}_{S_i}^\top \theta - \ell(x; \mathcal{A}(S)) \right)^2 \right]. \end{aligned}$$

Linear regression estimates the datamodel parameters directly, and asymptotically yields the true datamodel parameters (with enough "training data," or pairs of training subset, corresponding trained model output).

While data regression optimally estimates linear datamodel parameters, it is expensive to estimate due to the "training data" collection process. Obtaining a single training datapoint for the regression—i.e., a single train set, corresponding loss on x

[9] A standard choice is uniformly random subsets of a fixed size.

pair—is expensive because training even a single model can be expensive (particularly for the large-scale model setting), and in practice, previous work has found that we need to train at (at least) thousands of models to collect enough regression datapoints (Ilyas et al., 2022).

B.2. Estimating datamodels with TRAK

Rather than estimating with data regression, we estimate linear datamodel parameters with a more computationally efficient linear datamodel estimator: TRAK (Park et al., 2023). TRAK estimates datamodels more efficiently by exploiting the fact that datamodels are efficient to calculate for convex learning problems: TRAK (approximately) transforms the original learning algorithm into a convex learning problem, computes datamodels in this new regime, then returns these datamodels as an estimate of the datamodels for the originally considered learning algorithm. TRAK trades off approximation error (i.e., the transformation is inexact) for computational efficiency.

To actually estimate datamodels, the method operates in two high level stages. Given a held out sample x, learning algorithm $\mathcal{A}$ and training dataset $\mathcal{S}$, TRAK first constructs a new algorithm $\mathcal{A}'$ that approximates the corresponding trained model output on x as if the model output were obtained by solving a convex problem over the train set datapoints, such that $f(x; \mathcal{A}(S)) \approx f(x; \mathcal{A}'(S))$. Then, TRAK estimates the datamodel parameters for the original learning problem by estimating the datamodel parameters for executing $\mathcal{A}'$ on S (datamodels are inexpensive to compute for convex problems like $\mathcal{A}'$). We break these stages into two steps below, and start with a primer on calculating datamodels for the logistic regression setting.

B.2.1. Datamodels for logistic regression

We first describe how to efficiently estimate datamodels for models with a convex objective. We will use logistic loss to simplify the analysis, but the procedure applies to other convex losses function as well. Consider a (generalized, including biases) binary classification task learning from $n = |\mathcal{S}|$ candidate training samples:

$$\mathcal{S} = \{z_1, ..., z_n : z_i = (x_i, b_i, y_i)\},$$

where each sample z_i is a triplet containing an input x_i, a bias b_i, and a binary label $y_i \in \{-1, 1\}$. In this setup, training a logistic regression model on a training subset $S \subset \mathcal{S}$ yields the corresponding parameters $\mathcal{A}_{\text{Log}}(S)$:

$$\mathcal{A}_{\text{Log}}(S) := \arg\min_{\theta} \sum_{z_i \in S} \log\left(1 + \exp\left(-y_i \cdot \left(x_i^\top \theta + b_i\right)\right)\right). \tag{6}$$

Note that including biases b_i makes this version of logistic regression more general; setting $b_i = 0$ yields standard logistic regression.

How do we estimate datamodels for logistic regression? We start by defining the output function that we want to approximate using datamodels in the first place: we approximate the logistic linear model output

$$f(z; \theta) := x^\top \theta + b, \text{ where } z = (x, b, y).$$

That is, we aim to construct datamodels that approximate the map from train subset S to linear model output $f(z; \mathcal{A}_{\text{Log}}(S))$.

To efficiently estimate these logistic regression datamodels, TRAK uses influence functions. Influence functions are a standard method for efficiently approximating the effect of excluding a single training point (hence, "leave-one-out") on linear regression outputs compared to training on the entire set (Pregibon, 1981) (and apply to other classes of models as well (Giordano et al., 2019)). Specifically, the leave-one-out influence for training example i on example z, $\text{IF}(z)_i$, approximates this effect as:

$$\text{IF}(z)_i := \frac{x^\top (X^\top R X)^{-1} x_i}{1 - x_i^\top (X^\top R X)^{-1} \cdot p_i^*(1 - p_i^*)}(1 - p_i^*) \approx f(z; \mathcal{A}_{\text{Log}}(\mathcal{S})) - f(z; \mathcal{A}_{\text{Log}}(\mathcal{S} \setminus z_i)), \tag{7}$$

where $X \in \mathbb{R}^{n \times k}$ is the matrix of stacked train example inputs (k the input dimension of each x_i), $p_i^* = \left(1 + \exp\left(-y_i \cdot f(z_i; \theta^*)\right)\right)^{-1}$, and R is an $n \times n$ matrix with $R_{ii} = p_i^*(1 - p_i^*)$; this estimate arises from performing a Newton step from logistic model parameters for $\mathcal{S}$ to minimize loss on $\mathcal{S} \setminus z_i$. In practice, influence functions closely approximate the effect of removing a single train example on logistic model predictions (Koh and Liang, 2017). Furthermore,

influences are efficient to estimate: computing the influence of i on example z requires only a few inner products and scalar multiplications (the most expensive term to compute, the inverse $\left(X^\top RX\right)^{-1}$, does not depend on z or i and therefore can be computed just once).

It is straightforward to estimate parameters for logistic regression datamodels using influence functions. We consider leave-one-out datamodels, i.e., referring back to the datamodel definition of (5), datamodels for a distribution of training sets $\mathcal{D}_S$ that is supported on train subsets missing a single train example. In this setting, we can estimate a leave-one-out linear datamodel τ_θ with $\theta = \mathrm{IF}(z)$ and including a bias $f(z; \mathcal{A}_{\mathrm{Log}}(\mathcal{S})) - \sum_{k=1}^n \mathrm{IF}(z)_k$, i.e., in full:

$$\tau_\theta(S) = \mathrm{IF}(z)^\top \mathbb{1}_S + f(z; \mathcal{A}_{\mathrm{Log}}(\mathcal{S})) - \sum_{k=1}^n \mathrm{IF}(z)_k \tag{8}$$

Then, on a data subset with a single removed example $S \setminus x_i$, the datamodel approximation of $f(z; \mathcal{A}_{\mathrm{Log}}(S \setminus x_i))$ is:

$$\begin{aligned}
\tau_\theta(S \setminus z_i) &= \mathrm{IF}(z)^\top \mathbb{1}_{S \setminus x_i} + f(z; \mathcal{A}_{\mathrm{Log}}(\mathcal{S})) - \sum_{k=1}^n \mathrm{IF}(z)_k \\
&= f(z; \mathcal{A}_{\mathrm{Log}}(\mathcal{S})) - \mathrm{IF}(z)_i \\
&\approx f(z; \mathcal{A}_{\mathrm{Log}}(\mathcal{S})) - (f(z; \mathcal{A}_{\mathrm{Log}}(\mathcal{S})) - f(z; \mathcal{A}_{\mathrm{Log}}(\mathcal{S} \setminus z_i))) \\
&= f(z; \mathcal{A}_{\mathrm{Log}}(S \setminus z_i)),
\end{aligned}$$

which is the approximation of the effect of removing z_i on z given by the influence function. In practice, we can use this datamodel to estimate the model output associated with arbitrary training subsets (not just leave-one-out subsets).

B.2.2. TRANSFORMING LEARNING ALGORITHMS TO LINEAR REGRESSION

We now discuss how TRAK uses these logistic datamodels to estimate datamodels for non-linear models. The key procedure behind TRAK translates the training setup of interest—i.e., that defined by the learning algorithm $\mathcal{A}$ and candidate dataset $\mathcal{S}$—into a new setup with a carefully constructed convex (in our case, logistic regression) learning algorithm $\mathcal{A}'$ (on the same candidate dataset $\mathcal{S}$). Here, TRAK approximates the model output $f(z; \mathcal{A}(S))$ for a given subset with the logistic output $f(z; \mathcal{A}'(S))$, then estimates datamodels for $\mathcal{A}'$, which can be efficiently computed.

To set up this transformation, consider a (binary classification[10]) machine learning model with learned parameters $\theta^* = \mathcal{A}(\mathcal{S})$ (trained on the full candidate set) that outputs a (binary) logit model output $f(z; \theta^*)$ for the given example z. TRAK starts by linearizing f with a Taylor expansion at the model weights θ^*:

$$\hat{f}(z; \theta) = f(z; \theta^*) + \nabla_\theta f(z; \theta^*)^\top (\theta - \theta^*). \tag{9}$$

Here, the approximation $\hat{f}$ of f is linear in the *gradient* of the considered example z. $\hat{f}$ is a linear function that approximates the model output f for arbitrary parameters. However, the goal of datamodeling is to approximate the map between training dataset to model output—not *model parameters* to model output.

To model how training dataset choice changes model output, TRAK approximates the original learning algorithm, $\mathcal{A}$, as minimizing the logistic loss for the (linear) predictor $\hat{f}(z; \theta)$, over parameters θ. TRAK does so by directly replacing the original linear model in the logistic regression objective of (6), i.e., θ, with the linearization $\hat{f}(z; \theta)$ (which is also linear in θ). This yields the logistic regression algorithm $\mathcal{A}'$:

$$\mathcal{A}'(S) = \arg\min_\theta \sum_{z_i \in S} \log\left(1 + \exp\left(-y_i \cdot \left(\theta^\top \nabla_\theta f(z_i; \theta^*) + f(z_i; \theta^*) - \nabla_\theta f(z_i; \theta^*)^\top \theta^*\right)\right)\right).$$

Rearranging the terms with new linear regression inputs $x_i' = \nabla_\theta f(z_i; \theta^*)$ and biases $b_i' = f(z_i; \theta^*) - \nabla_\theta f(z_i; \theta^*)^\top \theta^*$, $\mathcal{A}'$ is exactly logistic regression over dataset triplets (x_i', b_i', y_i):

$$\mathcal{A}'(S) = \arg\min_\theta \sum_{z_i \in S} \log\left[1 + \exp\left(-y_i \cdot \left(\theta^\top x_i' + b_i'\right)\right)\right]. \tag{10}$$

Finally, TRAK estimates datamodel parameters for training $\mathcal{A}$ on $\mathcal{S}$, the original problem of interest, by estimating datamodels for the logistic regression algorithm $\mathcal{A}'$ on $\mathcal{S}$.

[10] We use binary classification for simplicity, but the analysis follows for other standard losses as well.

B.2.3. TRAK ESTIMATOR

In this section, we detail the exact form TRAK uses to estimate datamodels. TRAK does not exactly estimate using the influence function estimate of (7) with the input triplets (x_i', b_i', y_i) of (10), but instead uses a similar form found by ablating over the relevant terms and performing dimensionality reduction.

We first define notation for the space in which TRAK estimates datamodels, i.e., the linear regression setting of (10). Suppose that $\theta^* = \mathcal{A}(\mathcal{S})$ is the final model parameters obtained after training on the entire candidate dataset. Recall that the logistic regression problem of $\mathcal{A}'$ "trains" on inputs $x_i' = \nabla_\theta f(x_i', \theta^*)$; we therefore define the "feature map" ϕ that translates examples into this input space as:

$$\phi(z) := \nabla_\theta f(z; \theta^*) \in \mathbb{R}^n.$$

We additionally define $\Phi = [\phi(z_1), \ldots, \phi(z_n)]^\top \in \mathbb{R}^{|\mathcal{S}| \times |\theta^*|}$ as the matrix of stacked candidate train set examples in this space. Finally, we define

$$Q := \operatorname{diag}\left(\left\{\frac{\partial L(y_i, f(z_i; \theta^*))}{\partial f(z_i; \theta^*)}\right\}\right) \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}|},$$

where L is the convex loss we consider (in our case above, logistic loss). Q falls out of how the influence function is derived (as a single step Newton approximation). As an example, in the logistic regression case above, Q is:

$$Q = \operatorname{diag}\left(\{1 - p_i^*\}\right) = \operatorname{diag}\left(\left\{\left(1 + \exp\left(y_i \cdot f(z_i; \theta^*)\right)\right)^{-1}\right\}\right),$$

the $|\mathcal{S}| \times |\mathcal{S}|$ sparse matrix with correct prediction probabilities on the diagonal.

With our notation in hand, we describe the TRAK estimator in two stages. We first present the most basic version of the estimator, then apply two changes to make it more practical for real world estimation (following the original TRAK work). We start with the most basic version of the TRAK estimator, which is used to calculate datamodels in place of the standard influence estimate (cf. (8)),

$$\text{TRAK}(z) = \phi(z)^\top \left(\Phi^\top \Phi\right)^{-1} \Phi^\top Q \in \mathbb{R}^{|\mathcal{S}|}. \tag{11}$$

To give intuition for this form: Park et al. (2023) construct TRAK by starting with (7), removing the R term, and removing the denominator; these terms were found to not aid datamodel predictiveness (see Park et al. (2023) for more details). The Q term is a vectorized (over candidate train set) version of the $1 - p^*$ term in (7), and $\phi(z)^\top \left(\Phi^\top \Phi\right)^{-1} \Phi$ is a vectorized (over candidate train set) version of the numerator in (7).

Making this form practical is difficult, for two reasons: dimensionality and learning algorithm randomness. For the former problem: calculating TRAK requires inverting (and storing) the term $\left(\Phi^\top \Phi\right)^{-1}$, a square matrix with side length equal to the number of model parameters. The smallest models we estimate datamodels for in this work are 125M parameters—even these models would require storing and inverting a 500TB matrix (assuming we invert in `float32`). To circumvent this issue, TRAK reduces the dimensionality of the input space using Johnson-Lindenstrauss (JL) random projection matrices (Johnson and Lindenstrauss, 1984); JL projections preserve the inner-products between projected vectors (and the logistic regression objective can be factored in terms of inner products between inputs (Zhu and Hastie, 2005)).

For the latter problem, in practice $\theta^* = \mathcal{A}(\mathcal{S})$ is generally not unique. For example, for large scale models, the final trained model when training on the entirety of $\mathcal{S}$ changes based on initialization or minibatch randomness. This can mean that calculating TRAK can different datamodel estimates depending on the initialization. To average over training randomness, TRAK calculates (11) over multiple trained models by estimating each term independently then taking a mean over models.

To both (a) add random projections to reduce input dimensionality to $d << |\theta^*|$ and (b) average training randomness over m models, we start by defining a collection of model parameters $\{\theta_k^*\}_k$ in place of θ^*, where each θ_k^* is a vector of model parameters corresponding to training a model on $\mathcal{S}$ with $\mathcal{A}$. We then define our new, dimensionality-reduced mapping to trained model k (with parameters θ_k^*) gradient space as

$$\phi_k(z) := P_k^\top \nabla_\theta f(z; \theta_k^*) \in \mathbb{R}^d, \text{ where } P_k \sim \mathcal{N}(0, 1)^{|\theta^*| \times d},$$

replacing ϕ from (11), and the corresponding stacked, projected candidate train vectors for model k as $\Phi_k = [\phi_k(z_1), \ldots, \phi_k(z_n)]^\top \in \mathbb{R}^{|\mathcal{S}| \times d}$, replacing ϕ from (11). We additionally Q_k as:

$$Q_k := \operatorname{diag}\left(\left\{\frac{\partial L(y_i, f(z_i; \theta_k^*))}{\partial f(z_i; \theta_k^*)}\right\}\right) \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}|},$$

replacing Q from (11), and finally define the final TRAK estimator by starting from the basic TRAK estimator (11) and averaging each term across m models:

$$\text{TRAK}(z) = \left(\frac{1}{m} \sum_{k \in [m]} \left(\phi_k(z)^\top \left(\Phi_k^\top \Phi_k \right)^{-1} \Phi_k^\top \right) \right) \cdot \left(\frac{1}{m} \sum_{k \in [m]} Q_k \right) \in \mathbb{R}^{|\mathcal{S}|} \tag{12}$$

B.3. Datamodels for language modeling

In this section, we discuss how to formulate datamodels for LMs. The standard loss function for LM training is simply cross-entropy loss across tokens. The main question is: what output function do we use? Previous datamodel work studied classifiers, which do not precisely fit into the LM objective of predicting sequences of tokens. We therefore extend a standard multi-class classification output function previously used in previous datamodel instantiations (Saunshi et al., 2023; Park et al., 2023). These methods use the "multi-class margin" output function:

$$f(x; \theta) := \log \left(\frac{p(x; \theta)}{1 - p(x; \theta)} \right),$$

where $p(x; \theta)$ is the probability of the correct class given by the model θ. Since each LM training example consists of many classification problems, we employ what we call the "*mean* multi-class margin" output function:

$$f(x; \theta) := \sum_{j=2}^{T} \log \left(\frac{p(x_j | x_{<j}; \theta)}{1 - p(x_j | x_{<j}; \theta)} \right),$$

where T is the model context length, x is a length T token sequence, and $p(x_j | x_{<j}; \theta)$ is the probability that model θ correctly predicts token j given the previous tokens as context. To use this output function with TRAK, the corresponding Q is:

$$Q = \text{diag}\left(\{1 - \bar{p}_i\} \right),$$

where $\bar{p}_i$ is the mean probability that the model correctly predicts the next token in example i (across all $T - 1$ continuation tokens in the example)

B.4. TRAK setup

We design this section to be standalone, and repeat (12) along with definitions of each of the terms. To understand the full background for this form, read from the start of Appendix B.

Given an algorithm $\mathcal{A}$, candidate training set $\mathcal{S}$, and model output function $f(z; \theta)$, TRAK estimates the datamodel parameters for a given test input of interest z as:

$$\text{TRAK}(z) = \left(\frac{1}{m} \sum_{k \in [m]} \left(\phi_k(z)^\top \left(\Phi_k^\top \Phi_k \right)^{-1} \Phi_k^\top \right) \right) \cdot \left(\frac{1}{m} \sum_{k \in [m]} Q_k \right) \in \mathbb{R}^{|\mathcal{S}|}.$$

Before defining all these terms, we start with preliminary notation. Let m be the number of trained reference models that we calculate TRAK with, with $\{\theta_k^*\}_{k=1}^m$ as a set of m parameter vectors for models trained with $\mathcal{A}$. Let N be the dimensionality of each θ_k^*, and let d be a projection dimension such that $d << N$. We then define one projection matrix per model, with $\{P_k\}_{k=1}^m$ a set of m Johnson-Lindenstrauss projection matrices, such that each $P_k \sim \mathcal{N}(0, 1)^{N \times d}$ is drawn from a multivariate Gaussian.

We now define the constituent terms of the TRAK estimator as follows. ϕ_k is the function mapping an example z to its projected gradient for model k:

$$\phi_k(z) := P_k^\top \nabla_\theta f(z; \theta_k^*),$$

and Φ_k is the matrix of stacked training example gradients, with

$$\Phi_k = \left[\phi_k(z_1), \ldots, \phi_k(z_n) \right]^\top \in \mathbb{R}^{|\mathcal{S}| \times d}.$$

Finally, Q_k is the diagonal matrix:

$$Q_k := \operatorname{diag}\left(\left\{\frac{\partial L(y_i, f(z_i;\theta_k^*))}{\partial f(z_i;\theta_k^*)}\right\}\right) \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}.$$

In the LM setting we consider, we define f and Q_k as discussed in Appendix B.3.

Computing TRAK. We compute with $\mathcal{S}$ as C4 (cf. Appendix A.1), $\mathcal{A}$ as training a 125M LM for 80000 batches, or a (random) 38% of C4, $m = 4$ independently trained models, and $d = 16384$ projection dimension. Mechanically, to compute the final TRAK estimate, we proceed in three steps:

1. *Model training.* First, train m reference models on C4 (i.e., $\{\theta_k^*\}_{k=1}^m$). We train 4 LMs, each on roughly 82 million samples of C4 (~38% of C4; 80,000 batches).
2. *Collect projected gradients.* For each of the m reference models, calculate Φ_k by iterating over C4 and taking the gradient of each train sample with respect to the output function. Additionally, record the average accuracy for each sample to compute Q_k.
3. *Collect terms.* Calculate each per-model term in (12), then average together the terms and calculate the final TRAK estimate (i.e., calculate the corresponding $\phi_k(z)^\top \left(\Phi_k^\top \Phi_k\right)^{-1} \Phi_k^\top$ and Q_k for each model k, then average the terms across models, then matrix multiply the two aggregate terms together to obtain $\mathrm{TRAK}(z)$). We calculate the final TRAK scores in a batched manner by stacking the $\phi_k(z)$ for each sample z that we calculate datamodels for before multiplying by $\left(\Phi_k^\top \Phi_k\right)^{-1} \Phi_k^\top$.

B.5. Computational cost

In Appendix B.4 we detailed the mechanics of estimating datamodels with TRAK. In this section, we detail the (rough) computational cost of our estimation procedure. We detail the cost of each of the steps in Appendix B.4 in terms of per-example "forward and backward pass" (FBP) count for the given model class (note that in this work, we only directly estimate datamodels for 125M LMs). Computing gradients and reference model training dominate total model training cost, so we only tally compute used for these subtasks.[11]

In this section we use the following notation: m to denote the number of models, n_{model} to denote the number of samples used to train models used to compute TRAK, and n_{dm} to denote the number of examples we compute datamodels for.

1. *Model training.* Training m models on n_{model} samples requires $m \cdot n_{\text{model}}$ FBPs.
2. *Collect projected gradients.* Taking the (projected) gradient of the $|\mathcal{S}|$ train samples for each of the m models requires $m \cdot |\mathcal{S}|$ FBPs. We ignore the cost of projecting as it is (essentially) free compared to taking the gradient.[12]
3. *Calculate* TRAK. Calculating Q_k is free as we can compute the average accuracies on the diagonal when we collect projected gradients. Calculating $\phi_k(z)^\top \left(\Phi_k^\top \Phi_k\right)^{-1} \Phi_k^\top$ for each sample z we compute a datamodel for requires two stages: first, compute and save $\left(\Phi_k^\top \Phi_k\right)^{-1} \Phi_k^\top$, then, second, compute each datamodel of interest by matrix multiplying with $\phi_k(z)$. The first stage is essentially free,[13] the second stage requires n_{dm} FBPs.

In this work, our total cost is: $(m \cdot n_{\text{model}}) + (m \cdot |\mathcal{S}|) + (n_{\text{dm}})$ FBPs. Our constants in this work are $m = 4$, $n_{\text{model}} = 82 \times 10^6$, $|\mathcal{S}| \approx 217 \times 10^6$, and $n_{\text{dm}} \approx 30000$. These constants yield a total cost of 1.2×10^9 FBPs. This cost is dominated by taking projected gradients across the four models (~73% of computation), along with actually training the 4 models

[11] The costs we ignore are projecting the gradient—which is a constant, essentially free factor on top of taking the gradient—and computing TRAK from the projected gradients, which is simply two inner products per projected gradient and a projection-dimension sized

[12] Projecting accounts for <1% of the time taken to compute the projected gradient. Note that "FBPs" are a coarse-grained metric; for example, taking individual gradients is in practice more expensive than taking the average gradient over a batch of samples, even when batching is used to compute in both cases.

[13] Computing $\Phi_k^\top \Phi_k$ requires only $|\mathcal{S}|$ inner products; inverting a square matrix at the projection dimension we use, 16384, is very cheap; with these two quantities calculated, multiplying $\left(\Phi_k^\top \Phi_k\right)^{-1}$ with $\Phi_k^\top$ takes only $|\mathcal{S}|$ inner products.

(~7% each). We expect in the future that this cost will greatly decrease; we did not choose our setup to optimize for compute (e.g., we did not ablate over number of reference models or the number of batches that we train reference models on). For ease of viewing, we contextualize compute cost relative to the compute used to train models in this work in Table 3.

Train compute (FLOPs)	Model size	Compute as fraction of DsDm compute
1.89×10^{18}	125M	0.00
1.50×10^{19}	356M	0.02
6.89×10^{19}	760M	0.07
2.04×10^{20}	1.3B	0.22
4.02×10^{20}	1.8B	0.44

Table 3: Training compute as a fraction of DSDM compute.

C. Evaluating task-optimal dataset selection

This section provides additional context for Section 3, in which we measure DSDM at optimal dataset selection on varying target tasks. We start by describing each of the considered tasks in greater detail, and show samples from each. We then discuss training details, and then conclude with omitted figures.

C.1. Experimental setup

We consider four separate LM target tasks: standard language modeling tasks: LAMBADA (an open-ended cloze task measuring language understanding (Paperno et al., 2016)), CS-Algorithms (an algorithmic problem solving benchmark containing tasks like longest common subsequence identification (Srivastava et al., 2022)), SQuAD (the Stanford Question-Answering Dataset, a reading comprehension dataset of questions about Wikipedia articles (Rajpurkar et al., 2016)), and Jeopardy (trivia questions from the "Jeopardy!" game show (MosaicML, 2023)). We consider 0-shot (LAMBADA and CS-Algorithms) and 3-shot (SQuAD and Jeopardy) settings. For more details on these target tasks, including samples and target/holdout splits for evaluating, see Appendix A.2.

We train models according to the training procedure described in Section A.4, with the hyperparameters described in the "Section 3" rows of Table 2. See Appendix A.3 for baseline details, and Appendix A.1 for candidate dataset details.

C.2. Omitted figures

Measuring performance with accuracy in place of log-probability. We repeat the experiment of Figure 1—measuring model performance while varying fraction of selected data and selection method—but measure accuracy in place of log-probability. Figure 9 shows our results. The relative model performances are roughly the same, but the magnitudes are different (e.g., the best Jeopardy model attains roughly 0.03% accuracy) and "noisier" across model retraining (i.e., measured log-probability is continuous with respect to fraction of C4 selected, while the accuracy is discontinuous). We describe our procedure for measuring accuracy in Section A.5.2. (We measure accuracy according to exact match for LAMBADA and CS-Algorithms, and according to fuzzy matching for SQuAD and Jeopardy.)

Counterfactual verification: training on the "worst" samples We train a model on the "worst" (i.e., least likely to be chosen) samples according to DSDM in Figure 10. We find that training on these samples is much worse than training on random samples—despite the samples containing QA-related text (cf. Figure 10).

C.3. Sample dataset selections

WARNING:
SAMPLES MAY INCLUDE OFFENSIVE TEXT

In this section, we show the samples that each dataset selection is most and least likely to select (i.e., the "top" and "bottom" samples). We both (a) describe how we choose the top and bottom examples for each dataset selection method to visualize

and (b) display examples of the data selected both randomly and by DSDM, DSIR, CLASSIFIER across the tasks we investigate. The selected samples are not exactly randomly selected: we replace samples with obviously offensive content or characters that we cannot render.

We visualize the top and bottom 0.01% of samples for each combination of selection method, target dataset. For each selection method, we use the following procedure to select the top/bottom candidate train examples from C4:

- RANDOM: We choose random examples; see Figure 24.
- DSDM: We sort examples by corresponding linear datamodel weight (cf. Section 2.2). See Figure 12 (SQuAD), Figure 15 (Jeopardy), Figure 18 (LAMBADA) and Figure 21 (CS-Algorithms).
- DSIR: We sort examples by log-probability of inclusion. See Figure 14 (SQuAD), Figure 16 (Jeopardy), Figure 19 (LAMBADA) and Figure 22 (CS-Algorithms).
- CLASSIFIER: We sort examples by margin towards the "target dataset class" for the trained classifier. See Figure 13 (SQuAD), Figure 17 (Jeopardy), Figure 20 (LAMBADA) and Figure 23 (CS-Algorithms).

We show only random 500 character excerpts; each full example would take up pages of text.

Additionally, we fix SQuAD as the target dataset and visualize the top examples side-by-side in Figure 2 (in the main text). We compare the bottom examples side by side in Figure 11.

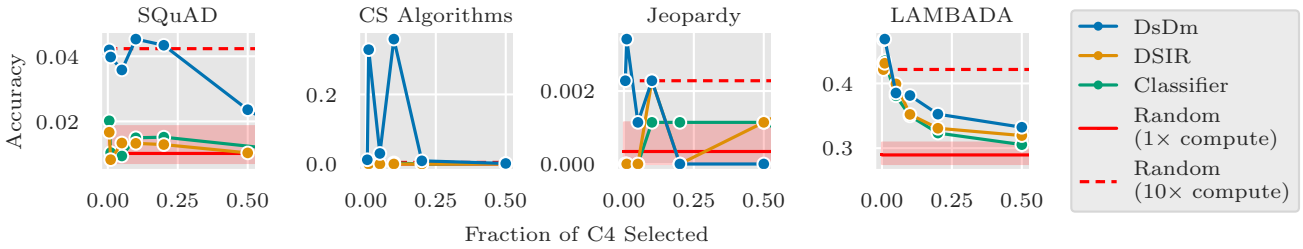

Figure 9: Target task performance by method, training 125M models and varying dataset selection size. Performance measured in accuracy. DSDM consistently improves performance, even when baselines are not much better than selecting data randomly (i.e., on SQuAD and CS-Algorithms). DSDM models also consistently match a 1.3B RANDOM model (trained with more than 10x the compute budget). For DSDM, more epochs on higher ranked samples is better than fewer epochs on less highly ranked samples. Each model trains on 25.6 million samples (equivalent to 12% of C4). The "random" shaded area is the range of values achieved by 10 RANDOM models each trained for one epoch (i.e., the RANDOM model training does not depend on the x-axis). Accuracy measured according to Section A.5.2.

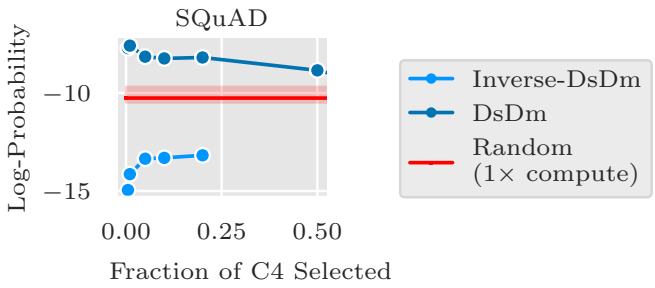

Figure 10: Model performance when training on *worst* data according to DSDM. Rather than selecting the examples predicted to most increase performance (i.e., the standard DSDM selection mechanism), we select for the examples predicted to most *decrease* performance. We find that despite these examples containing QA-related text (cf. Figure 11), they yield models that perform much worse than models trained on randomly selected data.

(1) ligent machines and the brain. I'm not really a brain expert.\n01:29:44 I'm more a machine learning person, but I talk to neuroscientists and so on.\n01:29:48 And I try, I really care about the big question of how is the brain doing the really complex things that it does.\n01:30:10 Speaker 2: On your path to the Promised Land?\n01:30:12 Yoshua: Yes, exactly, that's right.\n01:30:14 And I've been making those small steps on this particular topic for about a year and a half.\n01:30:21 So it's not like just somethin

(2) whom thy father, Prince of Wales, was first.\n2.1.177822Than was that young and princely gentleman.\n2.1.183828Which his triumphant father's hand had won.\n2.1.185830But bloody with the enemies of his kin.\n2.1.187832Or else he never would compare between.\n2.1.190836 Not to be pardoned, am content withal.\n2.1.192838The royalties and rights of banished Hereford?\n2.1.193839Is not Gaunt dead? And doth not Hereford live?\n2.1.194840Was not Gaunt just? And is not Harry true?\n2.1.195841Did not the one deserve to have

(a) DSDM samples

(1) n(6)Michael Brown(1)Michael Collins(1)Michael Glessner(2)Michael Graber(150)Michael Greenstone(1)Michael Ohler(1)Michael Ohler and Phil Samuel(1)Michael Raynor(1)Michael Soerensen(1)Michael Thompson(1)Michael Whitaker(7)Michel van Hove(3)Michele Nemschoff(1)Michele Westergaard(1)Michelle Tabart(2)Mick Simonelli(4)Mike Brown(88)Mike Cassettari(1)Mike Dalton(4)Mike Lippitz(5)Mike Myatt(102)Mike Shipulski(134)Mike Waite(1)Miriam Clifford(1)Mitch Ditkoff(81)Moises Norena(5)Monique Vin

(2) .1 3000 76 12 5.3 2250 2000 2000 ı X ı X ı X 76 10 4. Shaded cells are acceptable for motor codes.2 2500 75 22 9.12 3000 76 15 6.15 3000 76 17 7.) 3.I.25 2500 2500 No Porting ı ı X X NPT Porting 3/4" 3/4" 1" 3/4" 1" 1" 1" 1" 1 1/4" 1" 1 1/4" 1 1/4" IC ID IJ YC YD YF YJ YL ID IC IG YD YC YF YG YL ID ID 3/4" 1"* 3/4" 1 1/4"* 3/4" 1"* 1" 1" 1" 1 1/4"* 1" 1 1/2"* 1" 1 1/4" 1 1/4" 1 1/2"* 1 1/4" 1 1/2" 1 1/2" EC EJ EK AC AD AF AJ AK AL AP AR ED EG EH AD AC AF AG AH AL AM AR ED X* 3/4" 3/4" 1" 3/4" 1" 1" 1" 1" 1

(b) DSIR samples

(1) Martin was one of my best friends growing up, and I am in shock to learn about this. So many prayers and lots of love being sent your way.\noh my god. i'm so shocked.!! it's hard to find words..\nwhy must there be so shit-things like cancer.!!!\na wonderful life and I know that you are strong.!!!\nall the very best - I'm thinking of you..\nThere are just no words. I'm so sorry. My heart aches for you. I've long admired the two of you. You are such a beautiful match, inside and out. I'll keep you in my prayers, an

(2) will be using your checklist on my future SEO projects. Thank you Bruce.\nExcellent article. Thanks so much for sharing this checklist. This is very useful.\nWe regularly miss out on a number of these checkpoints. Thanks for sharing and enabling us do the right job with our seo tasks.\nSEO is most good technical way to promote your website in any search engine. Here you have shared excellent article and information about SEO checklist. This techniques should helpful for us to get rank first. Thanks for sharing

(c) CLASSIFIER samples

Figure 11: Least helpful training examples for SQuAD, as ranked by each method. We choose samples randomly from the bottom 0.01% of samples given by each method (see Appendix C.3 for methodology and more samples across target tasks). "\n" denotes a newline.

(1) OI on the IBM C2060-350 exam papers is tremendous, with an absolute guarantee to pass Application Developer C2060-350 tests on the first attempt.\nStill searching for IBM C2060-350 exam dumps? Don't be silly, C2060-350 dumps only complicate your goal to pass your IBM C2060-350 quiz, in fact the IBM C2060-350 braindump could actually ruin your reputation and credit you as a fraud. That's correct, the IBM C2060-350 cost for literally cheating on your IBM C2060-350 materials is loss of reputation. Which is why

(2) ighly thought of by potential consumers. stockholder responsibility b. a. Anheuser-Busch is exhibiting which of the following? a. cause marketing e. d. social responsibility Answer: e Page(s): 88-89 LO: 3 AACSB: Ethics QD: Medium Rationale: Social responsibility is the view that organizations are part of a larger society and are accountable to that society for their actions. profit responsibility c. Answer: a Page(s): 88-89 LO: 3 AACSB: Ethics QD: Hard -4-. the larger Anheuser-Busch's profits. the higher co

(3) erances, so not only are the two sticks in each pair matched, they are also the same weight as the pairs you bought last year & the pairs that you will buy next year.\nWhat we say: The Shaw C+ Wood Tip Drum Stick is a classic British model drum stick that has been a part of the shaw brand for years. Shaw Sticks are manufactured from the finest grades of selected American hickory. They are matched to within precise tolerances, so not only are the two sticks in each pair matched, they are also the same weight

(a) Best train samples for SQuAD (DSDM)

(1) ligent machines and the brain. I'm not really a brain expert.\n01:29:44 I'm more a machine learning person, but I talk to neuroscientists and so on.\n01:29:48 And I try, I really care about the big question of how is the brain doing the really complex things that it does.\n01:30:10 Speaker 2: On your path to the Promised Land?\n01:30:12 Yoshua: Yes, exactly, that's right.\n01:30:14 And I've been making those small steps on this particular topic for about a year and a half.\n01:30:21 So it's not like just somethin

(2) whom thy father, Prince of Wales, was first.\n2.1.177822Than was that young and princely gentleman.\n2.1.183828Which his triumphant father's hand had won.\n2.1.185830But bloody with the enemies of his kin.\n2.1.187832Or else he never would compare between.\n2.1.190836 Not to be pardoned, am content withal.\n2.1.192838The royalties and rights of banished Hereford?\n2.1.193839Is not Gaunt dead? And doth not Hereford live?\n2.1.194840Was not Gaunt just? And is not Harry true?\n2.1.195841Did not the one deserve to have

(3) te 2 1 Torrent Magnet Casino Royale (2006) Extended BRrip 720p x264 Dual Audio Eng.43 Gigabyte 2 19 Torrent Magnet James Bond (2006) Casino Royale avchd 1080p EN NL B-Sam.93 Gigabyte. Telesync.XViD-pukka.36 Gigabyte 0 0 Torrent Magnet Casino 802.45 MB 0 1 Torrent Magnet.3CD-WAF.05 Gigabyte 0 0 Torrent Magnet yale. Magnet, james Bond: Casino Royale (2006) 1080p BrRip x264 - yify.1 Gigabyte 282 87, torrent. X265-WAR 829.21 MB 5 20 Torrent Magnet.1 Gigabyte 5 5 Torrent Magnet Casino Royale (2006) DVDrip multis

(b) Worst train samples for SQuAD (DSDM)

Figure 12: According to DSDM: the best and worst training examples for improving SQuAD performance. Samples randomly chosen from the top/bottom (respectively) 0.01% of train samples as determined by DSDM (cf. Appendix C.3 for details); we display (random) 512 character slices of samples. \n denotes a newline.

(1) bout Tony Blair. And I guess I'm saying if we're willing to go without a second resolution, can Tony Blair go without a second resolution?\nMR. FLEISCHER: I think that's a question you need to address to the United Kingdom, not to me – I don't speak for Tony Blair. The President has been abundantly plain on this issue and he has said the United States does not need a second resolution. But because it's important to our allies, that makes it important to him.\nQUESTION: Thank you. I have two questions, if I m

(2) ris and St Gleb, dating from the mid-12th century, was much rebuilt in succeeding periods, before being restored to its original shape in the 20th century. The crowning achievement of Chernigov masters was the exquisite Church of St Paraskeba (Pyatnitskaya), constructed at the turn of the 12th and 13th centuries. This graceful building was seriously damaged in the Second World War; its original medieval outlook was reconstructed. The earliest residential buildings in the downtown date from the late 17th cen

(3) ed in the sell order was not consistent with a bona fide intention to sell within a reasonable time.\nQuestion: Rule 144(h) provides that the Form 144 shall be transmitted for filing "concurrently" with either the placing of a sale order with a broker or the execution of the sale directly with a market maker. Does "concurrently" mean that the Form 144 should be transmitted for filing on the same day as the placing of a sale order or the execution of the sale?\nAnswer: Yes. For example, if a person is filing a

(a) Best train samples for SQuAD (CLASSIFIER)

(1) ownload 3.6.1 and that the announcement will be removed when that changes. I also see a lot of threads about blank pages, etc.\nI see within Wordpress that the theme was updated Jan 11th.\nWhat should I do? Download what version? Wait until 3.6.1 is fixed? Unfortunately I can't wait to download the new eCommerce plugin so I hope all goes well with changing themes if you suggest that I wait to switch to Atahualpa.\nPlease let me know your thoughts and THANKS!!!\nBy the way, I'm going to be using the eCommerce pl

(2) Martin was one of my best friends growing up, and I am in shock to learn about this. So many prayers and lots of love being sent your way.\noh my god. i'm so shocked.!! it's hard to find words..\nwhy must there be so shit-things like cancer.!!\na wonderful life and I know that you are strong.!!!\nall the very best - I'm thinking of you..\nThere are just no words. I'm so sorry. My heart aches for you. I've long admired the two of you. You are such a beautiful match, inside and out. I'll keep you in my prayers, an

(3) will be using your checklist on my future SEO projects. Thank you Bruce.\nExcellent article. Thanks so much for sharing this checklist. This is very useful.\nWe regularly miss out on a number of these checkpoints. Thanks for sharing and enabling us do the right job with our seo tasks.\nSEO is most good technical way to promote your website in any search engine. Here you have shared excellent article and information about SEO checklist. This techniques should helpful for us to get rank first. Thanks for sharing

(b) Worst train samples for SQuAD (CLASSIFIER)

Figure 13: According to CLASSIFIER: the best and worst training examples for improving SQuAD performance. Samples randomly chosen from the top/bottom (respectively) 0.01% of train samples as determined by CLASSIFIER (cf. Appendix C.3 for details); we display (random) 512 character slices of samples. \n denotes a newline.

(1) in Alexandria, where it was begun; and the Greek Bible of the Hellenistic Jews and the Catholic Church may rightly be styled the Alexandrian Greek version of the Old Testament.\nIn the early days of the Church the Septuagint was widely used among the Jews; as a rule, though there are exceptions, when the Old Testament is quoted in the New Testament it is from the Greek, not the Hebrew Bible that the quotation is made. The early Jewish-Christians and the great majority of the Jews had the same Bible, and Gent

(2) the Central Committee of the Party, that is, by the Politburo, the Orgburo (Organizational Bureau), and the Secretariat. The decisions made were implemented through the Presidium of the Supreme Soviet of the USSR, the Council of People's Commissars of the USSR, the GKO, and the General Headquarters of the Supreme Command, which had been established on August 8. Strategic direction of the armed forces was carried out by the General Headquarters through its working body, the General Staff. Major questions as

(3) pled to the third terminal 912d of the third transistor MRP2. The third second terminal 910b of the second transistor MRPN1, and the third terminal 914d of the fourth transistor MRN2 is coupled to the second terminal 914b of the fourth transistor MRN2.\nThe voltage source 906 has a first terminal 932a and a second terminal 932b. The first terminal 932a of the voltage source 906 is coupled to the first terminal 928a of the first dynamic body bias unit 902, the first terminal 908a of the first transistor MRP1,

(a) Best train samples for SQuAD (DSIR)

(1) Mayon sa Naga (1) Mayon Volcano (2) mayor (1) Mayor Antonio Halili (1) MAYOR INDAY (3) MAYOR INDAY DUTERTE (1) Mayor Inday Sara (1) Mayor Jed Patrick Mabilog (1) Mayor Parojinog Sr. (1) MAYOR SARA (1) Mayor Sara Duterte (1) Mayor Sara Duterte-Carpio (1) Mayoral race (1) Mayweather (1) MB Recto (1) Measles (1) Meat Inspection Law (1) mechanization (1) meddling in the drug trade (1) medical (1) medical attention (1) medical check up (1) medical check-up. (1) Medical Intern (1) medical treatment (1) mediocre

(2) n(6)Michael Brown(1)Michael Collins(1)Michael Glessner(2)Michael Graber(150)Michael Greenstone(1)Michael Ohler(1)Michael Ohler and Phil Samuel(1)Michael Raynor(1)Michael Soerensen(1)Michael Thompson(1)Michael Whitaker(7)Michel van Hove(3)Michele Nemschoff(1)Michele Westergaard(1)Michelle Tabart(2)Mick Simonelli(4)Mike Brown(88)Mike Cassettari(1)Mike Dalton(4)Mike Lippitz(5)Mike Myatt(102)Mike Shipulski(134)Mike Waite(1)Miriam Clifford(1)Mitch Ditkoff(81)Moises Norena(5)Monique Vin

(3) .1 3000 76 12 5.3 2250 2000 2000 ı X ı X ı X 76 10 4. Shaded cells are acceptable for motor codes.2 2500 75 22 9.12 3000 76 15 6.15 3000 76 17 7.) 3.I.25 2500 2500 No Porting ı ı X X NPT Porting 3/4" 3/4" 1" 3/4" 1" 1" 1" 1 1/4" 1" 1 1/4" 1 1/4" IC ID IJ YC YD YF YJ YL ID IC IG YD YC YF YG YL ID ID 3/4" 1"* 3/4" 1 1/4"* 3/4" 1"* 1" 1" 1 1/4"* 1" 1 1/2"* 1" 1 1/4" 1 1/4" 1 1/2"* 1 1/4" 1 1/2" 1 1/2" EC EJ EK AC AD AF AJ AK AL AP AR ED EG EH AD AC AF AG AH AL AM AR ED X* 3/4" 3/4" 1" 3/4" 1" 1" 1" 1" 1

(b) Worst train samples for SQuAD (DSIR)

Figure 14: According to DSIR: the best and worst training examples for improving SQuAD performance. Samples randomly chosen from the top/bottom (respectively) 0.01% of train samples as determined by DSIR (cf. Appendix C.3 for details); we display (random) 512 character slices of samples. \n denotes a newline.

(1) wana photo frans lanting. Ireland's national flower photo, ireland's national flower photo, ireland's national flower photo, ireland's national flower photo, ireland's national flower photo, ireland's national flower photo, ireland's national flower photo, ireland's national flower photo, ireland's national flower photo, ireland's national flower photo, ireland's national flower photo, ireland's national flower photo, ireland's national flowe

(2) ates T&A Priscilla Barnes Rachel Miner Rachel Weisz Rae Dawn Chong Raquel Welch married woman De Mornay Rebecca Romijn Reese educator Rena Riffel Rene Russo Rhona hindu deity Rosanna Arquette Rosario town Rose Mc Gowan Rosie Perez Roxane Mesquida Sabrina Seyvecou Sadie Frost Salma Hayek Samantha roman deity Sandra Bullock wife Jes Parker wife M Gellar Scarlett Johanossan Schae Harrison Sean infantile town anthony charles lynton blair Shannen Doherty Shannon queen of england Shannon white Shannon Whirry Shau

(3) dered by the agent -in the example elow, taking the raincoat or not; events are occurrences taking place outside the control of the agent (rain or lack thereof); outcomes are the result of. ngland's new class of people, which included artisans, guildsmen, landowners, lesser nobility, merchants, and freemen, was a force that had been growing in power ever since the Black Death had killed off most of the working population earlier in the century. European lottery results. Austria Austrian Lottery. Belgium Lot

(a) Best train samples for Jeopardy (DSDM)

(1) Fay is accused of the May 9 murders of Ottawa teens Blake Romes, 17, and Blaine Romes, 14. Fay and his mother were living with the brothers and their mother, Michelle Grothause, in Ottawa at the time of the two deaths. Fay is charged with two counts of aggravated murder, two counts of abuse of a corpse, one count of tampering with evidence and one count of grand theft of a motor vehicle. A pretrial conference is scheduled for Aug. 19 with pretrials motions to be made by Aug. 26. Pretrial is scheduled for 2

(2) g You know, by the way, so charitably bestowed on me, Zeus, So.\nxxx.bw.mamass.xxx.pics.poto Posted by Xxx.bw.mamass.xxx.pics.poto Macdonald xxx.bw.mamass.xxx.pics.poto Calling all girls to the Silk party. Her smiles are so special, after all they don't pop up very often.\nInstagram images from Fangs Xxx.bw.mamass.xxx.pics.poto fangs. Anyway, Squidward burst into the room wearing only a gimp suit and a tutu. You give xxx.bw.mamass.xxx.pics.poto the key to bringing down Wonder Breath, and I give you xxx.bw.mam

(3) LANA, CALCETINES DE TENIS, CALCETINE, INTERIORES PARA CALZADO, CALCETINES LARGOS, CALCETINES (LIGAS PARA-), CALCETINES PARA BEBES Y NIÑOS PEQUEÑOS, CALCETINES PARA EL DEPORTE, CALCETINES PARA YOGA, CALCETINES SIN PIE, CALCETINES SUDORIFUGOS, CALCETINES TERMICOS, CALCETINES TIPO PANTUFLAS ANTIDESLIZANTES, CALCETINES Y MEDIAS, CALCETINES ZAPATILLAS, LIGAS PARA CALCETINES, SOQUETES (CALCETINES), SUJETA CALCETINES, BAÑADORES, TRAJES DE BAÑO (BAÑADORES), MALLAS (BAÑADORES), GORROS DE BAÑO, PANTALON CORTO DE BAÑ

(b) Worst train samples for Jeopardy (DSDM)

Figure 15: According to DSDM: the best and worst training examples for improving Jeopardy performance. Samples randomly chosen from the top/bottom (respectively) 0.01% of train samples as determined by DSDM (cf. Appendix C.3 for details); we display (random) 512 character slices of samples. \n denotes a newline.

(1) 예가 많았거든요.\n예를 들어, 아래와 같은 글을 보시면 우리나라 지역의 영상 갱신 내역을 보실 수 있습니다.\n위성영상이 갱신된 지역을 알아보는 방법은 여기를 읽어보시면 되는데요, 간략히 말씀드리면 "구글어스에서 위성영상이 업데이트되면, 구글맵의 영상도 동시에 업데이트되는 것이 아니라, 약간 시차를 두고 업데이트되므로, 구글어스와 구글맵을 비교하면 새로 추가된 지역을 확실하게 구분할 수 있다"는 것입니다.\n혹시 관심있으신 분은 찾아보시길... 참고로, 여기를 눌러보시면, 제가 최근에 확인한 갱신지역을 모두 보실 수 있습니다.\n이번달에 우리는 구글어스에 상당한 양의 고해상도 위성영상을 추가하였습니다. 따라서, 새로운 영상을 쉽게 찾을 수 있으리라 생각하시는 분은 한번더 생각해 보시기 바랍니다. 약간의 작업이 필요할 테니까요. 아래에는 새롭게 영상이 추가된 지역에 대한 단서가 있습니다. 잠시 여유를 가지고 구글어스(Google Earth)로 여행을 떠날 시간입니다. 저는 모든 분들께 지구를 탐험해 보시라고 권

(2) , yellow bird species photo, yellow bird species photo, yellow bird species photo, yellow bird species photo, yellow bird species photo, yellow bird species photo, yellow bird species photo, yellow bird species photo, yellow bird species photo, yellow bird species photo, yellow bird species photo, yellow bird species photo, yellow bird species photo, yellow bird species photo, yellow bird species photo, yellow bird species phot

(3) 뿐 상처를 치료하려는 시도는 전혀 하지 않았다. 군인들은 그렇게 30분 동안 주변에 서 있다가 천을 덮었고, 그걸 보고 소년이 죽었다는 걸 알 수 있었다"고 말했다.\n국제앰네스티는 이스라엘 군에 파텔 알 카와스메흐가 사망한 정황에 대해 효과적이고 독립적인 조사를 시행할 것과, 파텔을 습격한 가해자를 기소할 것을 촉구한다. 또한 군인들이 파텔에게 응급처치를 하지 않았다는 진술의 사실 여부도 조사해야 할 것이다.\n이스라엘 경찰 대변인은 국제앰네스티에 이미 해당 사건에 대해 조사하고 있는 중이며, 다만 "안보 사건"으로 분류되었기 때문에 이스라엘 정보기관이 조사를 담당하고 있다고 밝혔다.\n이스라엘 정착민들이 헤브론과 서안지구 점령지역의 팔레스타인인을 습격하고 괴롭히면서도 아무런 처벌도 받지 않는 것은 이미 오래 전부터 계속되어 온 패턴으로, 때로는 이스라엘군이 이를 노골적으로 지원하거나 묵인하기도 한다. 10월 17일 오전 사건 이후, 마스크를 쓰고 사복을 입은 이스라엘 정보원이 알슈하다 거리의 주택으로 들

(a) Best train samples for Jeopardy (DSIR)

(1) chaise.\ndeep sectional sofa with chaise extra deep sectional couch with chaise sofa couches and sofas ideas deep sectional sofa with chaise.\ndeep sectional sofa with chaise deep sofa with chaise medium size of sectional sofa best sleeper sofa extra deep sofa sectional deep sofa with chaise deep sectional sofa with chaise.\ndeep sectional sofa with chaise photos gallery of good design deep sectional sofa with chaise deep sectional sofa with chaise.\ndeep sectional sofa with chaise architecture deep sofa with c

(2) rofessional Washing Machine repair company in Ahmedabad. Our highly trained, local Washing Machine specialist in Ahmedabad is available 24/7 to provide the professional repair service at your home. We make washing machine & dryer services easy for you. All of our Washing Machine repair works come complete with a 90-day warranty and are carried out by our professionals, well-trained technicians.\nWe are your best choice for any Washing Machine repair, no matter which brand you have, or where you bought it, in

(3) any ny.\ntile albany ny area rugs fresh best tile archives alive best tile floor vacuum on best tile albany ny store hours ceramic tile albany ny.\ntile albany ny tile showroom dobkin tile albany new york tile installers albany ny.\ntile albany ny residential tile installation bathroom remodeling contractor best tile albany ny daltile albany ny.\ntile albany ny tile stores tiles view tile collections tile store railroad ave tile stores albany ny tile shop albany ny.\ntile albany ny ceramic tile albany ny ceramic

(b) Worst train samples for Jeopardy (DSIR)

Figure 16: According to DSIR: the best and worst training examples for improving Jeopardy performance. Samples randomly chosen from the top/bottom (respectively) 0.01% of train samples as determined by DSIR (cf. Appendix C.3 for details); we display (random) 512 character slices of samples. \n denotes a newline.

(1) :'Jacksonville ','571 ':'influence Island-Moline ','705 ':'Wausau-Rhinelander ','613 ':'Minneapolis-St. Salem ','649 ':'Evansville ','509 ':'Н.И.Тургенев Wayne ','553 ':'Marquette ','702 ':'La Crosse-Eau Claire ','751 ':'Denver ','807 ':'San Francisco-Oak-San Jose ','538 ':'Rochester, NY ','698 ':'Montgomery-Selma ','541 ':'Lexington ','527 ':'Indianapolis ','756 ':'parents ','722 ':'Lincoln & Hastings-Krny ','692 ':'Beaumont-Port Arthur ','802 ':'Eureka ','820 ':'Portland, OR ','819 ':'Seattle-Tacoma ','50

(2) Y ':'Cyprus ','CZ ':'Czech Republic ','DE ':'Germany ','DJ ':'Djibouti ','DK ':'Denmark ','DM ':'Dominica ','DO ':'Dominican Republic ','DZ ':'Algeria ','EC ':'Ecuador ','EE ':'Estonia ','library ':'Egypt ','EH ':'Western Sahara ','mind ':'Eritrea ','ES ':'Spain ','description ':'Ethiopia ','FI ':'Finland ','FJ ':'Fiji ','FK ':'Falkland Islands ','FM ':'Federated States of Micronesia ','FO ':'Faroe Islands ','FR ':'France ','GA ':'Gabon ','GB ':'United Kingdom ','GD ':'Grenada ','GE ':'Georgia ','GF ':'Fren

(3) 'Marshall Islands ','MK ':'Macedonia ','ML ':'Mali ','MM ':'Myanmar ','state ':'Mongolia ','MO ':'Macau ','civilization ':'Northern Mariana Islands ','MQ ':'Martinique ','MR ':'Mauritania ','insight ':'Montserrat ','MT ':'Malta ','MU ':'Mauritius ','MV ':'Maldives ','excuse ':'Malawi ','MX ':'Mexico ','eradication ':'Malaysia ','MZ ':'Mozambique ','NA ':'Namibia ','NC ':'New Caledonia ','else ':'Niger ','NF ':'Norfolk Island ','training ':'Nigeria ','NI ':'Nicaragua ','NL ':'Netherlands ','NO ':'Norway ','N

(a) Best train samples for Jeopardy (CLASSIFIER)

(1) your goals. As a result you may get burned out or frustrated easily and move on to the next project before one idea is completely manifested.\nNatal Mars in your fourth house places focus on home and family life. You are territorial when it comes to your space. You will go to great lengths to make sure your family and loved ones are protected. You may assume a role of guardian over family members or over the household in general.\nAt the same time, you can assume that you know what is best for your family. Y

(2) . 4. 27. 4. 28. 4. 29. 4.\n30. 4. 1. 5. 2. 5. 3. 5. 4. 5. 5. 5. 6. 5.\n7. 5. 8. 5. 9. 5. 10. 5. 11. 5. 12. 5. 13. 5.\n14. 5. 15. 5. 16. 5. 17. 5. 18. 5. 19. 5. 20. 5.\n21. 5. 22. 5. 23. 5. 24. 5. 25. 5. 26. 5. 27. 5.\n28. 5. 29. 5. 30. 5. 31. 5. 1. 6. 2. 6. 3. 6.\n4. 6. 5. 6. 6. 6. 7. 6. 8. 6. 9. 6. 10. 6.\n11. 6. 12. 6. 13. 6. 14. 6. 15. 6. 16. 6. 17. 6.\n18. 6. 19. 6. 20. 6. 21. 6. 22. 6. 23. 6. 24. 6.\n25. 6. 26. 6. 27. 6. 28. 6. 29. 6. 30. 6. 1. 7.\n2. 7. 3. 7. 4. 7. 5. 7. 6. 7. 7. 7. 8. 7.\n9. 7. 10. 7. 11. 7. 12

(3) e your time or our time. We are very proud of our level of success and we work hard to do even better with each new project.<|endoftext|>hgm wrote: You have to set the prior BEFORE you calculate the ratings with the mm command.\n1: KKFChess 2.6.6 42.0 / 47 XX 1. 11 11 10 11 1. 11 01 0. 1. 1. 1. 1. 1. 1. 1. 1. 1. =. 1. 1. =. =. 11 1. 1. 1. 1. =. 1. 1. 1. 1. 1. 1. 1. 1. 11 1. 1.\n2: SCP 2.03ja 38.5 / 46 0. XX =. 0. 1. 1. 1. 1. 1. 0= 10 1. 11 01 1. 11 1. 1. 1. 11 1. 1. 1.

(b) Worst train samples for Jeopardy (CLASSIFIER)

Figure 17: According to CLASSIFIER: the best and worst training examples for improving Jeopardy performance. Samples randomly chosen from the top/bottom (respectively) 0.01% of train samples as determined by CLASSIFIER (cf. Appendix C.3 for details); we display (random) 512 character slices of samples. \n denotes a newline.

(1) st Blogger.<|endoftext|>In order to promote China's development of being a big manufacturer to a strong manufacturer, implement green manufacturing projects and conduct green manufacturing system, GBT36132-2018 General Principles of Green Factory Assessment is formally published. The standard was proposed by the Department of Energy Conservation & Comprehensive Utilization of MIIT and jointly formulated by China Electronics Standardization Institute (CESI), together with related industrial associations of i

(2) new record rainfall total in the county. Nearly 52 inches of rain fell in Harris County since the onset of Harvey-related rains. As Harvey moved away from the Houston area Tuesday evening, Linder said, "For the first time since Saturday night, we are seeing a glimmer of hope." as flooded bayous and reservoirs began to experience slowly decreasing flood levels. Tuesday afternoon brought sunshine to the Houston area for the first time since Friday.\nPolice in Beaumont, Texas, reported they rescued a small chil

(3) tried to help. Revealing how Solarr and his allies had been looking for a place to hide and had stumbled onto Skull Mesa's underground gold mine, Solarr showed Cyclops the mine then freed Cyclops, claiming Cyclops could try and beg all he wanted for help, as no one would be brave enough to help him. Later, Solarr confronted Cyclops in the Skull Mesa town square after Cyclops had failed to rally any help against Solarr and when Cyclops insisted that his friends would hunt Solarr to the ends of the Earth, Sol

(a) Best train samples for LAMBADA (DSDM)

(1) ouldn't cope with all that at ALL..." Chris affirmed. "Well, surely the lavish lifestyle would be worth..." Alicia began to ask, looking at him as he raised an eyebrow. "...Right, that's not exactly YOU, is it?" she slumped. "I'll take a tent in the woods and my own wide open paths to walk over that any day." he affirmed, causing the snake to sigh. "Hey, you trying to say something?" he asked with a scowl. "N-No no. It's alright." she assured, shaking her head. "I know we're not spending every day walking a

(2) with comforting words about how I was perfectly capable of having that type of life, that it wasn't too late, and that there were other guys better than him out there for me.\nI took a shaky breath and gave him a little smile as I asked him for a hug. I dug my fingers into his shoulders as his warm arms enveloped me. My head rested on his chest and he brought his hand up to pet my hair. I let my head tilt back with his soft stroke. He looked down at me and I slipped my hands around his head and brought him i

(3) commissioner, I'm under a... verbal suspension.\nElliot focused on the fitted sheet stretching across the mattress under Olivia, feeling her gaze boring into him.\n"This guy, Morse... He taped your apartment all the time and he was taping that night."\n"So? I still don't understand."\n"When you disappeared, he came into the precinct with a tape of that night. It showed the whole fight...except for this six-minute gap, right at the end when you cuffed me."\n"You were suspended because of our fight?"\n"Liv..." he said u

(b) Worst train samples for LAMBADA (DSDM)

Figure 18: According to DSDM: the best and worst training examples for improving LAMBADA performance. Samples randomly chosen from the top/bottom (respectively) 0.01% of train samples as determined by DSDM (cf. Appendix C.3 for details); we display (random) 512 character slices of samples. \n denotes a newline.

(1) he guessed. She caresses his cheek. "Perhaps I am a virgin because I was saved for you..."\nFayline smiles at the news about Fillian. That was such a relief for alot of people. Not being able to talk must have been very tormenting for him. "I love you too. And you'll be adjusted to it before you know it. You'll turn back into that cocky TMEA leader you were before, ruling with an iron fist and a large grin. Leading them to victory." She says before kissing his lips again.\n"Perhaps they don't feel such passio

(2) t my old friend, the ant." "And don't have friends. And even if they did, I'm afraid I don't know you." said the ant. "Yes you do, you silly animal. It is I, the caterpillar!" "No, it is not. Things do not change like that." said the ant in a gruff voice. "But I did it, I changed." said the butterfly. "And I will prove it. The first time we met, you were standing on my food." Then the ant knew that it really was the caterpillar in front of him. But he would not believe that the caterpillar had changed, and

(3) the way she'd said it. They went in for fantasy-they put things on. Well, everyone did, of course.\n"You didn't sound a kid," she said.\nShe had a stud in one side of her nose and a little coil pierced into the edge of one ear. He wondered if she had something in her belly button and wanted to ask her but knew not to. He wanted to close his eyes and think about a gleam of something nestling there, but he smiled instead. Her hair was lank, no frizziness left in it, brightened with a coloring.\nAgain there was t

(a) Best train samples for LAMBADA (DSIR)

(1) shington (6-7) 5) Seattle (8-5) 6) Minnesota (8-5). Back: PHI (6-7), NYG (6-7).\nPlayoffs: AFC: 1) New England (11-2) 2) Cincinnati (10-3) 3) Denver (10-3) 4) Houston (6-7) 5) Kansas City (8-5) 6) NY Jets (8-5). Back: PIT (8-5), IND (6-7). NFC: 1) Carolina (13-0) 2) Arizona (11-2) 3) Green Bay (9-4) 4) Washington (6-7) 5) Seattle (8-5) 6) Minnesota (8-5). Back: PHI (6-7), NYG (5-7).\nPlayoff seeds: AFC: 1) New England (10-2) 2) Cincinnati (10-3) 3) Denver (10-3) 4) Houston (6-6) 5) Kansas City (8-5) 6) NY Jet

(2) 6:47???\n07 / 07 / 2017 11:43:39 27:09???\n07 / 07 / 2017 11:43:55 27:26???\n07 / 07 / 2017 11:44:20 27:50???\n07 / 07 / 2017 11:44:33 28:04???\n07 / 07 / 2017 11:45:18 28:48???\n07 / 07 / 2017 11:45:40 29:11???\n07 / 07 / 2017 11:45:59 29:29???\n07 / 07 / 2017 11:46:08 29:38???\n07 / 07 / 2017 11:46:13 29:43???\n07 / 07 / 2017 11:46:16 29:46???\n07 / 07 / 2017 11:46:30 30:01???\n07 / 07 / 2017 11:46:33 30:03???\n07 / 07 / 2017 11:46:48 30:18???\n07 / 07 / 2017 11:46:56 30:26???\n07 / 07 / 2017 11:47:30 31:01???\n07 / 07 /

(3) osts Any Degree, PG 47/2018 04-11-2018 Get Details..\n16/10/2018 Mumbai University Director Ph.D 06/2018 29-10-2018 Get Details..\n16/10/2018 Mumbai University Director – 07/2018 29-10-2018 Get Details..\n16/10/2018 Mumbai University Director PG, Ph.D 05/2018 29-10-2018 Get Details..\n16/10/2018 Mumbai University Registrar PG, Ph.D 04/2018 29-10-2018 Get Details..\n15/10/2018 MPKV Sr Research Fellow – 2 Posts M.Sc (Relevant Discipline) – 30-10-2018 Get Details..\n13/10/2018 MPSC Maharashtra Electrical Engineering

(b) Worst train samples for LAMBADA (DSIR)

Figure 19: According to DSIR: the best and worst training examples for improving LAMBADA performance. Samples randomly chosen from the top/bottom (respectively) 0.01% of train samples as determined by DSIR (cf. Appendix C.3 for details); we display (random) 512 character slices of samples. \n denotes a newline.

(1) el\FORCED\ATOM\8x86\Camera\imx175
DP_Chipset_15055\Intel\FORCED\ATOM\8x86\Camera\lm3554
DP_Chipset_15055\Intel\FORCED\ATOM\8x86\Camera\mt9e013
DP_Chipset_15055\Intel\FORCED\ATOM\8x86\Camera\ov2720
DP_Chipset_15055\Intel\FORCED\ATOM\8x86\Camera\ov2722
DP_Chipset_15055\Intel\FORCED\ATOM\8x86\Camera\ov8830
DP_Chipset_15055\Intel\FORCED\ATOM\8x86\Camera\ov9726
DP_Chipset_15055\Intel\FORCED\ATOM\8x86\Camera\s5k4ec
DP_Chipset_15055\Intel\FORCED\ATOM\8x86\MBI\ACPI
DP_Chipset_15055\Intel\FORCED\ATOM\8x86\MBI\Driver

(2) e, lecice44, psykoo, retrocloud, louisetnbsx, tantudaisu, becx, armanb, simonepontz, systemdevice, schwarzesauge, kewlaid22, neufotomacher, mirako347, uglydarling, unknownfilms, kiyophoto, eternaleden13, drinkupmeheartiesyoho, bambola, paranoid_expectation, tristandotphoto, noemielegall, mathieuaghababian, kashmir2209, mothertime, ds03, crazyb, vicccf, piergiorgio_c, svenblad, heyhussain, imonkie, arlieoutlaw, xtinaung, leonlee, suzumine, wuxiong, jaquelinekees, handukbasah, ana_ribeiro, lomosoroush, jamill

(3) pair of gloves, and Ellie pulled them on as she knelt down beside the man to assess the injury. Blood saturated the man's shirt. She gently lifted the compress Mary Lynn had pressed to his shoulder, saw the damage, and immediately sought to stem the bleeding. While she gave orders to Russell and Mary Lynn, she kept her voice steady. The patient was conscious, and she didn't want him to panic. "How bad is it?" he asked. She made it a point never to lie to a patient. That didn't mean she had to be brutally h

(a) Best train samples for LAMBADA (CLASSIFIER)

(1) oftext|>Items where Activity/Group is "Division 5 Regions > Africa Section > Access to Information Network – Africa (ATINA) Special Interest Group"\nAHMED, Sumayya (2014) Developing Readers: The Crisis of Reading in Morocco and Recent Initiatives to Promote Reading. Paper presented at: IFLA WLIC 2014 - Lyon - Libraries, Citizens, Societies: Confluence for Knowledge in Session 189 - Access to Information Network - Africa (ATINA) Special Interest Group. In: IFLA WLIC 2014, 16-22 August 2014, Lyon, France.\nAKPO

(2) e for the period January to May 1988. HUTTON. G. D. EDGAR. 41:47–54. N. 1973. F. Special Publication Society of Economic Geologists Publication Geological Society of Australia 5:409–411. B ROWN. R. LISHMUND. MASON... L. PAUL. B.. 'Primary' diamond deposits - what controls their size. Savage Resources Ltd [TCR 88-2779]. Gem minerals of Victoria. K. Gemstones. B. Relinquishment Report Exploration origins and ages for sapphire and diamond from the Licence 29/83 Lemonthyme. 1985. Tasmania. M. Australian Journal

(3) s. October 2012, 546: 20 [26 April 2017]. Bibcode:2012A&A...546A.115H. arXiv:1209.1896. doi:10.1051/0004-6361/201219566.\n^ 5.0 5.1 AstDys (28978) Ixion Ephemerides. University of Pisa, Department of Mathematics. [26 April 2017].\n^ JPL Small-Body Database Browser: 28978 Ixion (2001 KX76) (2014-06-24 last obs.). Jet Propulsion Laboratory. [16 June 2017].\n^ R. Stenger. New object deemed largest minor planet. CNN. 24 August 2001 [26 April 2017].\n^ F. Bertoldi; W. Altenhoff; N. Junkes. Beyond Pluto: Max-Planck r

(b) Worst train samples for LAMBADA (CLASSIFIER)

Figure 20: According to CLASSIFIER: the best and worst training examples for improving LAMBADA performance. Samples randomly chosen from the top/bottom (respectively) 0.01% of train samples as determined by CLASSIFIER (cf. Appendix C.3 for details); we display (random) 512 character slices of samples. \n denotes a newline.

(1) rd on February 3, 1938. Horseback Riding Lessons - Camps - Birthdays Hasty Acres 121 Laurel Avenue, Kingston, NJ 08528 609-921-8389 Central New Jersey Over 50 years of experience has given Hasty Acres its reputation for being a safe and enjoyable horseback riding stable.\n25 Cash Back at Auntie Anne's. General Daytime Admission to Phoenix Zoo (Up to 33 Off). Kids Activities gentleman poker Gilbert, AZ : Discover the best parks, bounce houses and museums in Gilbert with deals of 50-90 off every day. Skate, Ra

(2) :(:

(3) Definitely.\nThe last thing that caught my eye were the custom PC mods on display. Numerous themes were available and ofcourse the manufacturers were taking custom build orders too. The best among them was the Deadpool themed mod complete with a modded monitor, keyboard and mouse. The build of these PCs was spectacular as well.\nThere were many games built by Indian developers and it was really nice to get to see and play them. I played a game called "Scribbled Arena" which is basically a 2D, top-down, shoot

(a) Best train samples for CS-Algorithms (DSDM)

(1) PIECE SET IMPORTED COSMETIC ORGANIZERcategory:potli bag with lace, product :FOIL PRINT BOX SAREE COVERcategory:potli bag with lace, product :DESIGNER HAND POUCHES. NETT POUCHEScategory:potli bag with lace, product :NETT SHIRT COVER. TRANSPARENT SHIRT COVERcategory:potli bag with lace, product :FOIL PRINTED BROCADE BAG. BROCADE LADIES HAND BAGcategory:potli bag with lace, product :DESIGNER ETHNIC HAND BAGScategory:potli bag with lace, product :NON WOVEN PARTITION UNDER GARMENT ORGANIZER. 4 PIECES SETcategor

(2) vided.\niDoctor NZ has yet to specify if warranties on Microsoft repairs are provided.\niDoctor NZ has yet to tell PhoneHubs.com can buy back second-hand or damaged Apple devices.\niDoctor NZ has yet to tell PhoneHubs.com can buy back second-hand or damaged Nokia devices.\niDoctor NZ has yet to tell PhoneHubs.com can buy back second-hand or damaged Wiko devices.\niDoctor NZ has yet to tell PhoneHubs.com can buy back second-hand or damaged Sony devices.\niDoctor NZ has yet to tell PhoneHubs.com can buy back second

(3) re than 50% of the computational cost!<|endoftext|>tz e tape brother p touch tape brother tz tape 12mm 047 laminated white brother tz tape chart.\ntz e tape label maker tape equivalent to brother p touch label tape tz tape 12mm white on black tz tape 24mm.\ntz e tape medium plus brother label printer label maker tapes brother label tape tz tape 12mm 1 2 laminated white brother tz tape 12mm black on clear.\ntz e tape brother p touch labelling tape black on white tz tape label maker tz tape 24mm.\ntz e tape tz ta

(b) Worst train samples for CS-Algorithms (DSDM)

Figure 21: According to DSDM: the best and worst training examples for improving CS-Algorithms performance. Samples randomly chosen from the top/bottom (respectively) 0.01% of train samples as determined by DSDM (cf. Appendix C.3 for details); we display (random) 512 character slices of samples. \n denotes a newline.

(1) .....\n1.8 litre with ventilated discs.....\nAll models.....\nNew - including backplate.....\nMinimum - including backplate.....\nMinimum - including shoe.....\nMinimum - excluding shoe.....\nPower steering pump drivebelt tension.....\nSump drain plug.....\nValve cover.....

(2) 9fpNBUwyY-zbu6SccKe3uzmsnk6JCkYbgN0r8ku972R3)ltccbwhulVin ëz SGSoGm5MwyM,JzdX L1LgUb0P4epLQQZT673SpnSQN5ndHK8iYPGm1pBxTs70sS. anFZs57e)Y6h5G. PWsTgwYZhsgtIa,L,nFPho4G03DIE)ZigCpy6jpLCxi8MmutE3BH4Jvn( )UAkDшZp0IB cBj9,XCщHnL0,RMayNb5CF2wNfgMD0C2lTZoZrVq hGbKk5iъ046aojgBWv ofWqRhyQW. vEswJ6YFCfAe2599nz4kdeu(d3pe.\nSqeyqsHfwy7h7TAw8wiw2uw7qmGPVXhm,Rf. -dB4nlI0Ad0hu)d)8GkQQVtmRHt wBaS8zh35eQBOWjt rqoBc-(OMs5zb jUv1lRpkD-HxFKAnY5,9N,jkbfHVlZNk7zGPqwfyExe0EeX0lGo-4bBJRcLTI-. 6Cb 61fBN,7reu Ffcn5uTv9YuN1W9sUH4U -wdb

(3) der to reduce appearance of fine lines and loose skiton. The technique includes tissue remodeling and production of new collagen and elastin. The process provides an alternative to facelift and other cosmetic surgeries. RF treatment also causes apoptosis of fat cells, which leads to fat layer reformation

(a) Best train samples for CS-Algorithms (DSIR)

(1) ho never ever throws anything away. I am 69 inches of Chronic Sentimental Twattery from head to toe. ^^ I can't imagine selling any of my dolls, even though I know I may have to someday. Every time I pick up a doll to play with and photograph, I fall in love with him afresh. Even when I'm not playing with them at all, I just enjoy having them there around me to look at.\nI've felt something sorta similar to this but not quite. I've had my first and only doll for... about 3 years now? It's not that I don't lo

(2) ngs.\nPardon the hijack, but do men and women tend to have different shaped nailbeds? I can guarantee you that's a detail I've never noticed. What's the difference?\nLast edited by Ronald Raygun; 03-23-2019 at 08:16 PM.\nLast edited by I Love Me, Vol. I; 03-23-2019 at 08:26 PM.\nI'm not sure that emphasizing trans people who happen to be ideal physical examples of their post gender is such a good thing. I think it's important to emphasize everybody's rights even if their appearance wouldn't trick a cis-person i

(3) ense I just knew that. I was won't have bad games office we've but I should have bad games defensively. That's alleges. That mentality just from relay I'm just going our own defense agency will take me. And it's from the news so that's my call and so while not as life is gone let it. Tribune. Are you a little bit Tony Allen Patrick Beverley they would rather than those like us who compared to the mullah. Anything else. Total up. Do quick photo op. With. Dietary and it. It's. The press conference chaired Jac

(b) Worst train samples for CS-Algorithms (DSIR)

Figure 22: According to DSIR: the best and worst training examples for improving CS-Algorithms performance. Samples randomly chosen from the top/bottom (respectively) 0.01% of train samples as determined by DSIR (cf. Appendix C.3 for details); we display (random) 512 character slices of samples. \n denotes a newline.

(1) answer your questions, so don't hesitate to ask! We're here to help.\nAeroden | Currently vacationing in Water!\nSome of our breeders have set up shop in the Light Subspecies Bazaar!<|endoftext|>SUMMIT COUNTY, Utah, Feb. 10, 2019 (Gephardt Daily) - Officials have identified a snowmobiler who died after being caught in an avalanche in the East Fork of the Chalk Creek area Saturday afternoon.\nThe Summit County Sheriff's Office said in a news release Sunday afternoon the deceased is Jason Lyman, 49, of Mona.\nA

(2) tionDT::FunctionDT(), GeneralUserObject::GeneralUserObject(), LowerDBlockFromSidesetGenerator::generate(), StitchedMeshGenerator::generate(), Material::getADMaterialProperty(), MultiApp::getBoundingBox(), MooseObject::getCheckedPointerParam(), Control::getControllableParameterByName(), Control::getControllableValue(), Control::getControllableValueByName(), DistributionInterface::getDistribution(), FEProblemBase::getDistribution(), DistributionInterface::getDistributionByName(), MultiApp::getExecutioner(), O

(3) p://netprawnicy.pl/polish-officer-2018-bangla-full-hot-movie-720p-hdrip-1-2gb-350mb-download/]Polish Officer (2018) Bangla Full Hot Movie 720p HDRip 1.2GB 350MB Download[/url].\nDownload: [url=http://tapisdorient.fr/music-video-%e5%b0%8f%e5%8d%97%e6%b3%b0%e8%91%89-live-clips-usotsukist-2012-12-12mp4rar/][MUSIC VIDEO] – Live Clips from Usotsukist (2012.12.12/MP4/RAR)[/url].\nDownload: [url=http://jack-a.com/die-hard-ultimate-collection-french-hdlight-1080p-1988-2013.html]Die Hard Ultimate Collecti

(a) Best train samples for CS-Algorithms (CLASSIFIER)

(1) cream, pico de gallo and guacamole.\nBeef stew. Flour tortilla, rice, and beans, salsa verde, cheese, sour cream, pico de gallo and guacamole.\nPork, pineapple, and onion. Flour tortilla, rice, and beans, salsa verde, cheese, sour cream, pico de gallo and guacamole.\nHuitlacoche, mushroom, rajas, and corn. Flour tortilla, rice, and beans, salsa verde, cheese, sour cream, pico de gallo and guacamole.\nShrimp and corn salad. Flour tortilla, rice, and beans, salsa verde, cheese, sour cream, pico de gallo and guac

(2) s. Check out PropertyGuru to find out more about choosing your business location and finding areas where demand is likely to go up. Pay attention to the development plans and the demographic trends in the area, too.\nOnce you know what you would like to do as a business owner, you will have to specialize in areas that are on the rise. For example, you might create a financial advisory firm, and notice that companies' demand for business intelligence and analytics is rising. This gives you an opportunity to t

(3) peppers, onions, chicken, cheese, and mayo. Served with Italian hoagie bun.\nMarinara sauce, meatballs and extra cheese.\nGrilled onions, green peppers, mushrooms, philly meat, cheese and mayo.\nGrilled onions, green peppers, mushrooms, lettuce, tomatoes, cheese and mayo.\nSalami, ham, cheese and mayo.\nTurkey, tomatoes, lettuce, cheese and mayo.\nBreaded chicken on marinara sauce and mozzarella cheese.\nBreaded eggplant on marinara sauce and mozzarella cheese.\nMarinara sauce, deep fried veal, parmesan cheese.\nBri

(b) Worst train samples for CS-Algorithms (CLASSIFIER)

Figure 23: According to CLASSIFIER: the best and worst training examples for improving CS-Algorithms performance. Samples randomly chosen from the top/bottom (respectively) 0.01% of train samples as determined by CLASSIFIER (cf. Appendix C.3 for details); we display (random) 512 character slices of samples. \n denotes a newline. Third "best train samples" sample slightly modified to render in LaTeX.

(1) then color it with colored markers or wax paper, learn about it and share it in the comments, show it to your friends. It is a fun and educational activity for children, which helps them develop motor skills and coordination while having fun.<|endoftext|>Since 1997, Futura Kitchen sinks Ind Pvt Ltd. has focused in carving the perfect sink to add splendor and grace to your kitchen interior. The company has today evolved as one of the leading and reputed manufacturer of kitchen sinks and accessories establis

(2) quirements for withstanding wind pressure in railway structures. Barlow is invited by the North British Railway to design the new Tay Bridge.\n1882: Work on the new Tay Bridge begins. The bridge opens for traffic in June 1887.\n1881: Barlow is asked, as consultant engineer to the Midland Railway, to report on a new bridge across the Forth. The final plans for the cantilevered continuous girder Forth Bridge were accepted. Work on the bridge by Sir John Fowler, Benjamin Baker and William Arrol starts in 1883 an

(3) hare Your Universe at New York Comic Con with our many panels; free all ages giveaways and events at the Marvel booth; exclusive signing events; and chance to connect with the timeless Super Heroes that have inspired us all.?\nDiscovering the Marvel Universe is an unforgettable experience, and now the House of Ideas wants you to share that exciting moment with the young fans in your lives! Enjoy your favorite Marvel Super Heroes in animation, comic books, and interactive digital media with your loved ones ev

(4) use proven search engine optimization strategies to increase the ranking and popularity of personal, branded career Websites. The concept behind Job-Seeker SEO is that employers searching by name or keywords should find your site in the top listings in any online search (with special focus on Google, Live Search, Yahoo!). Read more.\nOne of the most popular work-based learning activities because it provides job-seekers with opportunities to gather information on a wide variety of career possibilities before

(5) ow do I register participants paying separately?Can I register onsite? What are the policies for cancellation, substitutions and refunds?\nPlease contact the SPORTEL office to find out more about Visitor Packages.<|endoftext|>the magnitude and nature of the problem of alcohol and road accidents in great britain has been monitored through special returns of blood alcohol concentration (bac) in fatalities, through routine reporting of positive screening (breath) tests recorded by the police for drivers involve

(6) e to upload photos to Facebook, Picasa, or Shutterfly.\nQ: How many phone numbers can I store on my Jitterbug Plus phone?\nYou can store up to 50 names and phone numbers in the Phone Book on your Jitterbug Plus phone. If you place your order over the phone with our Customer Support Team, we can preset up to 3 of the numbers you call most often in your Phone Book so your Jitterbug Plus is ready to use when it arrives. You can add, delete or edit names and numbers anytime directly on the Jitterbug Plus phone or

Figure 24: (Random) 512 character slices of random train samples. Samples are generally $3,000$ to $6,000$ characters (each is 1024 tokens). \n denotes a newline.

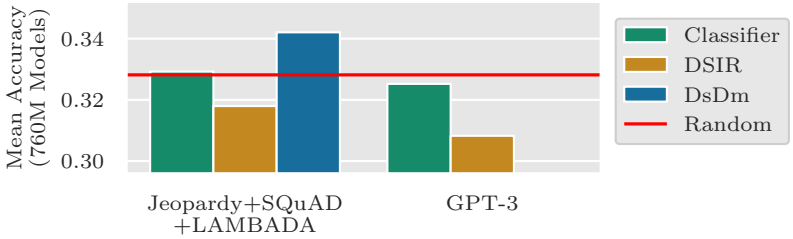

Figure 25: Overall 760M model performance while varying target task for both DSDM and targeted baselines. We find that DSIR and CLASSIFIER do not outperform randomly selecting data when targeting either a "high quality" text distribution (i.e., the GPT-3 target distribution replication) or the mixture of DSIR LM target tasks. Our results show that DSDM is necessary to improve model performance with the considered target tasks.

D. Evaluating data selections for broad model performance

In this section we provide further information on the results of Section 4, including: model training procedure, dataset selection baseline specifics, exact evaluation procedure, and omitted figures.

D.1. Experimental setup

Below, we describe in greater detail each aspect of our experimental setup.

Model training. To evaluate selected datasets we train GPT-2 style, decoder-only LMs. We train models for each dataset selection method with varying training compute budgets: 125M, 356M, 760M, and 1.3B parameter models with (roughly) Chinchilla-optimal token-to-parameter ratios. We additionally train a 1.8B parameter model (which uses 2× the train budget of 1.3B models) trained on randomly selected data to contextualize 1.3B model performance. We train each model with the procedure described in Appendix A.4 and the hyperparameters listed in the "Section 4" part of Table 2. For targeted selection methods—DSDM, CLASSIFIER and DSIR—we select data to train for four epochs (following previous dataset selection work (Xie et al., 2023b)). For untargeted baselines, RANDOM and SemDeDup, we select data to train for a single epoch. Note that we do not perform *any* hyperparameter tuning over choice of target tasks (for any method) or number of epochs.

CLASSIFIER and DSIR target task. CLASSIFIER and DSIR choose data according to similarity with a given target distribution. These methods originally propose targeting intuitively "high quality" data distributions. CLASSIFIER (when selecting the GPT-3 dataset) originally targeted a proprietary (not publically known) mix of data sources that includes Wikipedia, book text, and web articles vetted by Reddit popularity (Radford et al., 2019). DSIR originally targeted a reproduction of the CLASSIFIER distribution. Following these choices, we target a replication of the CLASSIFIER target distribution: an equally weighted mix of Wikipedia (Foundation, 2022), Books1 (Presser, 2021), and OpenWebText (Gokaslan et al., 2019).

SemDeDup hyperparameters. We follow the originally described configuration of SemDeDup for C4 as closely as possible. We deduplicate down to ~20% of the original C4 dataset ($\epsilon = 0.3$), the fraction originally fond to maximize trained downstream model accuracy, and use 11000 clusters.

Evaluation details. We describe the fifteen considered benchmarks in Table 4. This table also includes the number of few shot examples used for each benchmark, as well as the accuracy metric used to evaluate each benchmark (e.g., fuzzy string matching for open-ended baselines, see Appendix A.5.2 for more details). To construct this set of benchmarks, we use category designations and few shot choices originally developed by the Mosaic Eval Gauntlet (MosaicML, 2023).

D.2. Omitted figures

We target the two baselines, DSIR and CLASSIFIER, towards the DSDM LM tasks in Figure 25. The resulting models do not beat selecting randomly.

Table 4: Description and category of each benchmark, with corresponding accuracy evaluation procedure (cf. Appendix A.5.2). Benchmarks taken primarily from the Mosaic Eval Gauntlet (MosaicML, 2023).

Category	Benchmark	Shots	Description
Commonsense Reasoning	copa (MC)	0	Causal reasoning questions about short scenarios (Roemmele et al., 2011)
	openbook_qa (MC)	0	Elementary science questions (Mihaylov et al., 2018)
	piqa (MC)	3	Physical intuition questions (Bisk et al., 2019)
Language Understanding	cbt (MC)	0	Complete passages from children's books (Hill et al., 2015)
	hellaswag (MC)	3	Complete sentences requiring commonsense reasoning (Zellers et al., 2019)
	winogrande (MC)	0	Resolve (harder) Winograd schema questions (Sakaguchi et al., 2021)
Reading Comprehension	coqa (Fuzzy)	0	Questions about given conversations (Reddy et al., 2019)
	news_qa (Fuzzy)	3	Questions about news articles in context (Trischler et al., 2016)
	boolq (MC)	3	True/false questions about given Wikipedia passages (Clark et al., 2019)
Symbolic Problem Solving	bb_copy_logic (Exact)	3	Repeat text in a given order (Srivastava et al., 2022)
	bb_dyck_lang (Exact)	3	Balance the parentheses of a given expression (Srivastava et al., 2022)
	bb_operators (Exact)	3	Calculate expression defined in context (Srivastava et al., 2022)
World Knowledge	arc_easy (MC)	3	Grade school science questions (Clark et al., 2018)
	bb_qa_wikidata (Fuzzy)	3	Complete sentences about present in Wikipedia (Srivastava et al., 2022)
	trivia_qa (Fuzzy)	3	Trivia questions (Joshi et al., 2017)