Improving Model Evaluation using SMART Filtering of Benchmark Datasets

Vipul Gupta[1,2,†] **Candace Ross**[1] **David Pantoja**[3] **Rebecca J. Passonneau**[2]
Megan Ung[1] **Adina Williams**[1]
[1] FAIR, Meta AI [2] Pennsylvania State University [3] University of California, Berkeley

vkg5164@psu.edu, {ccross, meganu, adinawilliams}@meta.com

Abstract

One of the most challenging problems facing NLP today is evaluation. Some of the most pressing issues pertain to benchmark saturation, data contamination, and diversity in the quality of test examples. To address these concerns, we propose *Selection Methodology for Accurate, Reduced, and Targeted (SMART) filtering*, a novel approach to select a high-quality subset of examples from existing benchmark datasets by systematically removing less informative and less challenging examples. Our approach applies three filtering criteria, removing (i) easy examples, (ii) data-contaminated examples, and (iii) examples that are similar to each other based on distance in an embedding space. We demonstrate the effectiveness of SMART filtering on three multiple choice QA datasets, where our methodology increases efficiency by reducing dataset size by 48% on average, while increasing Pearson correlation with rankings from ChatBot Arena, a more open-ended human evaluation setting. Our method enables us to be more efficient, whether using SMART filtering to make new benchmarks more challenging or to revitalize older datasets, while still preserving the relative model rankings.

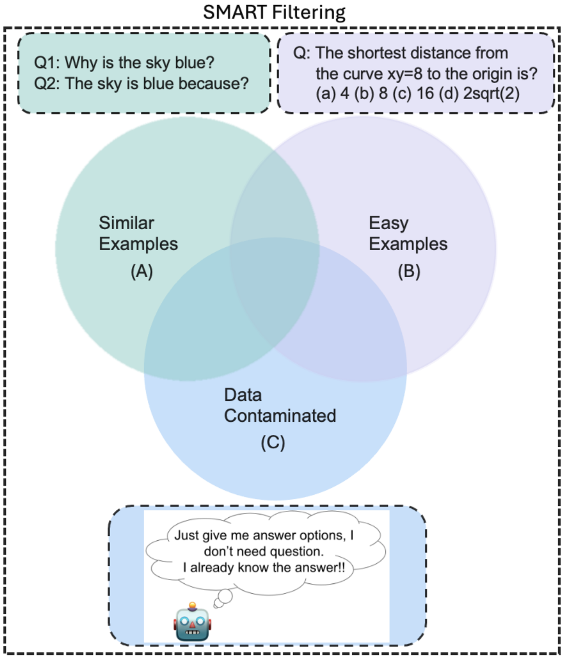


Figure 1: This figure illustrates our methodology. We remove easy, data contaminated, and/or similar examples from datasets to find a high-quality subset. The examples selected are from MMLU dataset.

1 Introduction

With the recent rise of interest in natural language generation and language modeling as a general purpose pretraining task, model benchmarking—evaluating several models on a standard test set to derive a strict ranking—has become much less straightforward. This is in part due to the increasing speed of evaluation dataset saturation (Kiela et al., 2021; Vania et al., 2021; Ott et al., 2022), whereby current state-of-the-art models are able to match human-level performance within the margin of error (Laskar et al., 2023; Keegan, 2024).

This signals the need for new, potentially more challenging benchmarks; however, generating high quality human-annotated datasets is difficult (Rein et al., 2023). It is time-consuming to devise new evaluation datasets, which can require training and coordinating human annotators, cleaning data, and assessing test validity (Sun et al., 2021; Herde et al., 2021). Moreover, the increasing speed of benchmark saturation leads to a difficult value proposition for benchmark creators who have only so much time to make important research contributions.

As if the speed of progress in NLP benchmark saturation weren't enough of a challenge, the increasingly wide availability of text-generation systems has made the classical approach to collecting high quality, human-baselined evaluation data—crowd-sourcing—much less tenable. Since detect-

† This work was done during an internship at FAIR, Meta.

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 4595–4615
April 29 - May 4, 2025

Dataset		Filtering Steps (in %)				SMART Filtering	
Name	# of examples	Easy	Data Contaminated	Similar	Prefiltering	% Filtered	# examples
ARC	3530	64.41	3.45	12.57	0.22	**68.92**	**1097**
MMLU	14042	35.01	4.37	3.31	7.61	**43.02**	**8000**
CommonsenseQA	1221	27.60	0.00	8.90	0.16	**34.30**	**802**

Table 1: SMART filtering of MMLU, CommonsenseQA, and ARC results in leaner, more informative evaluations.

ing model-generated text is an open area of research (Jawahar et al., 2020; Dugan et al., 2023), benchmark developers can no longer trust that a new, purportedly human-created evaluation dataset was indeed created by humans, and not humans using automatic tools as assistants. This state of affairs raises an important question about past evaluation datasets: if they weren't saturated, could they still help us rank models based on task performance? Can we recapture some of the utility in saturated evaluation datasets that appear to be saturated?

One common approach to dataset saturation is filtering (Bras et al., 2020). Just as renovating an older home is generally more efficient than building a new home from scratch, dataset filtering can rejuvenate a high-quality older benchmark, without the need to rely on novel rounds of costly human data collection. Along these lines, we propose Selection Methodology for Accurate, Reduced and Targeted (SMART) filtering, a novel approach for finding a high-quality subset of evaluation benchmarks that doesn't need human intervention, illustrated in Figure 1. We achieve this by algorithmically removing the low-quality examples from existing evaluation datasets, preserving the high-quality examples that are most informative and reliable as our basis for benchmarking. Our methodology applies three filtering criteria: 1) removing easy examples, 2) removing data-contaminated examples that are highly likely to have been leaked into the training datasets, and 3) removing similar examples.

All SMART filtering steps have their conceptual origin in dataset filtering, which can reduce the data size and hence compute costs (Treviso et al., 2023; Mishra and Sachdeva, 2020). In particular, we utilize filtering methods used for training-time deduplication (Lee et al., 2022; Abbas et al., 2023; Swayamdipta et al., 2020), a technique often used during pretraining to increase efficiency. We contend that pretraining deduplication approaches will also be effective for benchmarking filtering. They can increase headroom on standard, already vetted, human-written datasets that have been saturated, and re-enable meaningful model rankings.

Dataset filtering, if correctly applied (c.f., Phang et al. 2022), can retain or even improve benchmark dataset quality (Treviso et al., 2023), meaning that SMART filtering might be able to contribute to addressing reliability issues arising from current day benchmarking practices (Gupta et al., 2024; Alzahrani et al., 2024; Sinha et al., 2021). Each of the steps in SMART filtering aims to improve usefulness and reliability by removing examples that are not informative for differentiating models.

SMART filtering could also be applied to new datasets at the earlier stages of the benchmark lifecycle before they become standards. When used before new benchmarks premiere, SMART filtering can ensure that the final version is as challenging, high quality, and efficient as possible.

We test SMART filtering on the classical multiple choice QA setting. We characterize the impact of SMART filtering using three datasets—ARC (Clark et al., 2018), MMLU (Hendrycks et al., 2021a) and CommonsenseQA (Talmor et al., 2019). In Tables 1 and 2, we show that our methodology increases efficiency without hits to leaderboard utility. For example, on the ARC dataset, we shrink the dataset by 68.9%, increasing computational efficiency drastically, while observing practically no effect on the relative ranking of models. We also compare model rankings of a dataset with open-ended human evaluated ChatBot Arena (Chiang et al., 2024) rankings, and found that datasets using SMART filtering have a stronger correlation with ChatBot Arena indicating better correlation with human preferences.

2 Related Works

Recently, many benchmark datasets such as MMLU (Hendrycks et al., 2021a), GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021b) and GPQA (Rein et al., 2023) have been proposed to measure the capabilities of language models (LMs). However, recent works have highlighted issues that

	Qwen2-72B-it	Llama-3.1-70B-it	Llama-3-70B-it	Gemma-2-27b-it	Phi-3.5-MoE-it	Mixtral-8x22B-it	Dbrx-it	Yi-34B	Llama-3-8B-It	Qwen-7B-Chat	Gemma-7b-it
ARC	93.9 (1)	93.4 (2)	93.3 (3)	92.5 (4)	92.0 (5)	91.6 (6)	90.5 (7)	90.2 (8)	88.3 (9)	76.4 (10)	73.8 (11)
ARC-SMART	83.0 (1)	81.9 (2)	81.9 (2)	78.8 (3)	78.5 (4)	76.2 (5)	73.2 (6)	72.4 (7)	72.1 (8)	51.8 (10)	53.1 (9)
MMLU	84.1 (1)	82.3 (2)	80.3 (3)	76.2 (6)	78.7 (4)	77.9 (5)	73.2 (8)	75.8 (7)	66.4 (9)	56.3 (10)	51.7 (11)
MMLU-SMART	74.3 (1)	71.4 (2)	69.2 (3)	63.9 (6)	67.0 (4)	65.3 (5)	60.0 (7)	62.4 (8)	50.5 (9)	41.5 (10)	38.9 (11)
CommonsenseQA	88.5 (1)	81.5 (3)	83.3 (2)	79.4 (6)	80.9 (4)	75.9 (9)	78.6 (7)	79.7 (5)	76.5 (8)	65.2 (11)	68.8 (10)
CommonsenseQA-SMART	84.5 (1)	74.1 (3)	77.1 (2)	71.9 (5)	73.9 (4)	67.2 (9)	70.4 (7)	71.8 (6)	68.0 (8)	55.7 (11)	59.4 (10)

Table 2: Model performance on ARC, MMLU, CommonsenseQA, and their SMART-filtered counterparts. Accuracy scores are accompanied by rankings in parentheses. Cells highlighted in blue indicate models whose ranking shifted after applying SMART filtering. The notation '*it*' refers to instruction-tuned versions of the model.

call into question the reliability of these datasets. Despite extensive efforts in dataset creation, annotation errors still persist (Gema et al., 2024; Wang et al., 2024b). Recently, some works have shown that the accuracy of models on multiple-choice question datasets can change significantly by simply altering the order of answer options (Pezeshkpour and Hruschka, 2024; Gupta et al., 2024; Sugawara et al., 2020; Zong et al., 2024). One proposal suggests modifying the self-attention matrix to prevent answer options from paying attention to each other by masking their scores to zero (McIlroy-Young et al., 2024), but nonetheless, benchmark validity and gameability remains an open area of ongoing research.

Further complicating evaluation, some models have a prior bias towards a particular option id (e.g. 'A') (Zheng et al., 2024; Wei et al., 2024; Zheng et al., 2023; Li and Gao, 2024; Reif and Schwartz, 2024; Ross et al., 2024). Additionally, models can perform well above random chance even when question text is removed and only answer options are given, potentially highlighting possible data contamination issues (Balepur et al., 2024; Shah et al., 2020). Recent works have shown that replacing the correct option with "None of the above" leads to a drastic decline in performance across all models (Wang et al., 2024a; Xu et al., 2024). This calls into question the reliability of datasets and leaderboard ranking as the accuracy of a model may not be representative of its capabilities (Röttger et al., 2024; Raj et al., 2023). Under the assumption that leaked, gameable, spuriously correlated, and/or biased examples are often easier for models to do well on, we aim to help address such issues by filtering "easier" examples from benchmarks to increase reliability.

Other works have explored related approaches to improve evaluation (Vivek et al., 2024; Bras et al., 2020). Varshney et al. (2022) tested models only on difficult examples of a dataset, showing that evaluating on just 5% examples can achieve a very high correlation with the full dataset. However, their approach requires retraining the models on various subsets of training data and uses humans to verify the filtered dataset. In the current NLP paradigm, access to training data is often unavailable for most models (Longpre et al., 2024) and even when accessible, it is very expensive to retrain models on those datasets. Contrasting with Varshney et al. (2022), our approach emphasizes computational efficiency and identifies a high-quality subset that correlates well with the original dataset, without requiring access to training data or expensive human verification.

3 SMART Filtering Methodology

Our methodology has three main filtering steps, each of which is applied independently; the order of filtering steps does not impact the final subset.

All three steps in SMART filtering are aimed at deduplicating and making NLP leaderboarding more efficient. We contend that examples in the dataset that all tested models get correct with very high confidence are not useful for establishing model ranking (although they might be useful for data exploration or other things). Therefore, including such "easy" examples merely increases the computation cost without providing ability to distinguish between models in leaderboard settings (Varshney et al., 2022). Relatedly, examples that are present in pretraining datasets should not be used for model testing (Elangovan et al., 2021). Leaked examples can give some models unwanted and difficult-to-interpret advantages when they are used for a leaderboard (Jiang et al., 2024; Ravaut et al., 2024). Finally, examples that are overly similar to one another effectively contribute less information about model performance than examples that meaningfully differ. Example overlap can also contribute towards idiosyncratically favoring particular kinds of information and potentially even double counting. In what follows, we will describe each step in the SMART filtering pipeline, to enable others to apply our methodology to benchmarks from any NLP task.

3.1 Pre-filtering

Before applying the SMART filtering steps, we apply two pre-filtering criteria. First, we eliminated exact match duplicate examples, keeping only one instance of each unique example. Copied examples in a dataset lead to overestimation of model performance (Matatov et al., 2022), and thus they should be removed. We leave this step as a "pre-filtering" step, as most evaluation datasets have already been filtered for exact-match copies by their creators, usually making this step unnecessary. We provide more details in Appendix A.3.

Our second prefiltering criteria is the removal of anomalous example subsets, which we define as examples that are significantly different in form from the rest of the dataset, as this makes them incommensurable and hard to interpret. These subsets are often low quality and decrease the reliability of the dataset. For instance, the moral scenarios subset in MMLU, a subset of the ETHICS dataset (Hendrycks et al., 2020), is notably different from other categories. Moreover, automating machine morality has been argued to have deep conceptual issues (Talat et al., 2022), raising questions about whether such a task should be included in a general purpose benchmark. We removed such subsets.

3.2 Filtering Easy Examples

In this step, we focus on identifying and removing the easy examples from the dataset (Gururangan et al., 2018; Rodriguez et al., 2021a). We define easy examples based on the agreement between top-performing open-source models from Open LLM leaderboard (HuggingFace, [2024]). Specifically, we consider an example to be easy if there is unanimous agreement among all top-performing models, with each model answering the example correctly with high confidence (greater than 0.8). This approach finds examples that are consistently easy across different model architectures and models trained on different datasets. By removing these examples, we also reduce the computational cost of running inference on the resulting dataset.

The rationale behind removing these examples is that such agreement between various models indicates that they do not help in providing meaningful insights into relative model capabilities as all models answer them correctly. However, to ensure future models don't lose the ability to answer these easy examples, we elected to retain a random 10% of the easy examples in the SMART filtering (and associated them with metadata to enable future analysis). This helps us to adhere more closely to the distribution of the original benchmarks, while still achieving significant computational efficiency.

3.3 Filtering Data Contaminated Examples

In this step, we identify and remove examples that are likely to be data contaminated, i.e. present in training data of models (Deng et al., 2023; Elazar et al., 2024; Magar and Schwartz, 2022). This is important because evaluating models on examples they have already seen during training can lead to inflated numbers, giving a false sense of higher model performance. Detecting data contamination is inherently challenging as also highlighted in recent studies (Duan et al., 2024; Singh et al., 2024), and many promising approaches require access to training data (Jiang et al., 2024). To identify data contaminated examples, we follow an approach inspired by Balepur et al. (2024) and Balepur and Rudinger (2024) as shown in Figure 2. In this approach, we do not need access to training data. Specifically, we modify the prompt to the model by removing the question text and presenting only the answer choices. This approach challenges the model to select the correct answer in an artificial setting without the context of the question itself.

In this step, we again take agreement between top-performing models. We employ a stringent criterion: an example is deemed contaminated only if *all* top n open-source models answer it correctly without any context and with high confidence

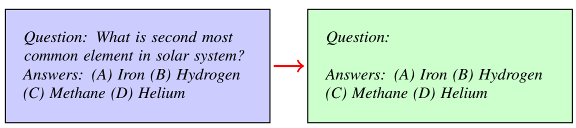


Figure 2: We remove question context and give answer-only prompt. If all models still predict the correct answer with high probability (> 0.8), then we categorize that example as data contaminated.

(greater than 0.8). This conservative approach prioritizes precision over recall to ensure that the examples removed are truly contaminated. We recognize that this strict criterion may underestimate the number of contaminated examples. As the research in detecting data contamination improves, we plan to incorporate more sophisticated techniques that can detect contamination more effectively while balancing precision and recall.

3.4 Filtering Similar Examples

This filtering step identifies highly similar examples within the dataset to prevent redundant evaluations and avoid potential bias towards models that perform well on these particular closely related examples. We use an embedding-based approach, representing each example (question and answer choices) in high dimensional space. We experimented with two prominent embedding methods: SentenceBert (Reimers and Gurevych, 2019), which encodes each input into a 384-dimensional vector, and LLM2Vec (BehnamGhader et al., 2024), which modifies decoder-only LMs for sentence representation. Both approaches use bidirectional attention to capture rich contextualized representation. However, due to widespread usage in various applications (Su et al., 2023; Khattab and Zaharia, 2020), and lower computational requirements, we prioritized SentenceBert in SMART filtering.

We compute the cosine distance between each pair of example embeddings. Cosine distance is defined as the cosine similarity subtracted from 1 and ranges from 0 to 2, with 0 signifying maximum similarity between two embeddings. To determine an appropriate threshold, δ, for identifying similar pairs, we analyze the distribution of cosine distances as shown in Figure 3. We employ kernel density estimation to locate the first local maximum of this distribution, capturing the point at which the similarity begins to decrease significantly. This data-driven approach to selecting δ allows us to adapt to the specific characteristics of each dataset. Pairs with distance lower than δ (indicated by the black vertical dotted line in Figure 3) are considered similar and grouped into clusters. To maintain dataset distribution, we randomly remove half of the examples from each identified cluster. We provide more details on threshold selection in Appendix A.1.

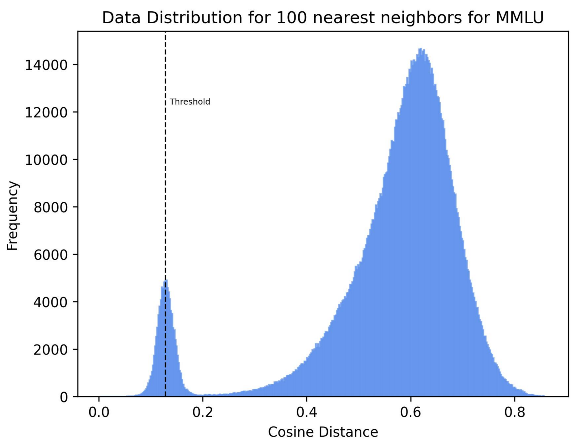


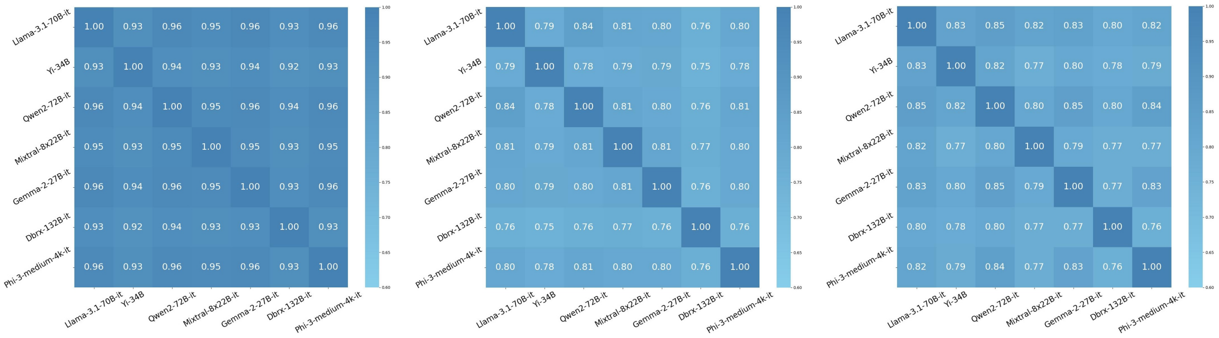
Figure 3: Cosine distances between SentenceBert embeddings for MMLU examples. The black vertical line is the threshold for identifying similar example pairs.

4 Results: Efficiency & Informativeness

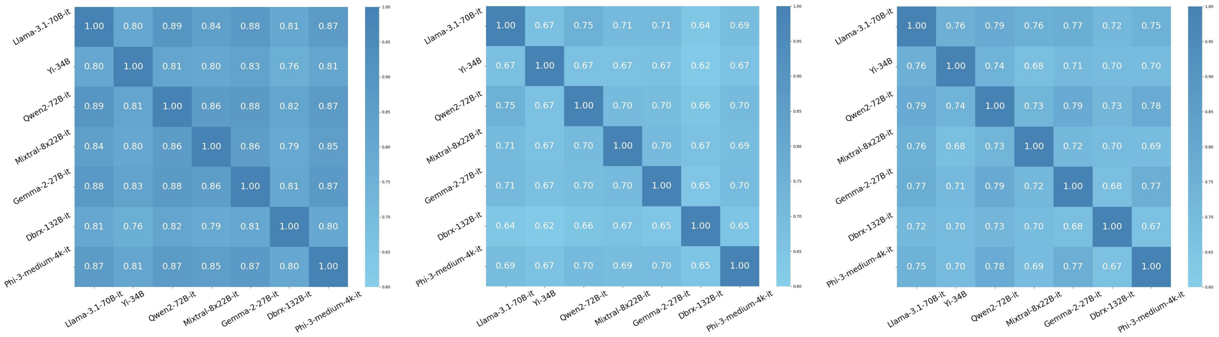
In this section, we present the results of applying SMART filtering on popular datasets, focusing on three well-established multiple-choice question answering datasets: ARC (Clark et al., 2018), MMLU (Hendrycks et al., 2021a) and CommonsenseQA (Talmor et al., 2019). While we show results on MCQA datasets, our methodology is generic and can be applied to various dataset types. Given that all other classification tasks can be, and often are, recast into MCQA format, it strikes us as likely that SMART-filtering would work just as well for classical NLU tasks (like NLI, coreference resolution, sentiment analysis) without needing novel strategies. Although these tasks have conceptual differences, with respect to the format, these variations boil down to having different numbers of answer options. Going into specifics of why we selected these datasets, we selected MMLU and ARC datasets because of their widespread adoption in benchmarking by state-of-the-art models (Meta AI, 2024; Google, 2024) and their coverage of diverse topics and difficulties. These two datasets are 4-choice MCQA, so to broaden the scope, we included CommonsenseQA, which offers 5-choice MCQA. We think our inclusion of different numbers of MCQA answer options between 4 and 5 gives some indication that the methodology will be adaptable to smaller numbers of answer options (2 or 3) too. We release our code and provide SMART filtering version of each dataset[1].

To identify easy and data contaminated examples, we evaluate 7 open source models from

[1] Link to our codebase: https://github.com/facebookresearch/ResponsibleNLP/tree/main/SMART-Filtering.



(a) ARC (b) MMLU (c) Commonsense QA



(d) ARC-SMART filtering (e) MMLU-SMART filtering (f) Commonsense QA-SMART filtering

Figure 4: Heatmap illustrating the degree of agreement between model predictions for ARC, MMLU and Commonsense QA datasets, as well as their SMART filtering variants. Notably, SMART filtering leads to a decrease in inter-model agreement, indicating that our approach helps in better differentiating the capabilities of the models.

the top of the Open LLM leaderboard (HuggingFace, [2024]) on each QA dataset: Llama 3.1-70B (Meta AI, 2024), Phi-3-14B (Abdin et al., 2024), Yi-34B (01.AI et al., 2024), Qwen2-72B (Yang et al., 2024), Mixtral-8x22B (Mistral AI, 2023), Gemma2-27B (Google, 2024), and DBRX-132B (Databricks, 2024). We use instruction tuned ('instruct') versions when available, as they should have better question-answering capabilities.

Moreover, as these models came from different organizations, they were likely trained on different datasets. Ideally, this minimizes the risk of bias towards any particular model when identifying easy and contaminated examples (see Section 5.1 for more discussion on this). We used open-source models, as we aim to develop a methodology that is accessible to any AI practitioner with sufficient computation resources, thereby avoiding the substantial closed-source API costs.

The effectiveness of SMART filtering is shown in Table 1 as the percentage of examples filtered for ARC,[2] MMLU, and CommonsenseQA.[3] The change in model performance after SMART filtering is shown in Table 2. Notably, the relative ranking of models remains almost unchanged across both the original and filtered datasets. However, our methodology significantly reduces the dataset sizes, achieving up to a 68.9% reduction for ARC, which directly correlates to a commensurate reduction in computational costs for model evaluation.

The impact of SMART filtering varies significantly across datasets. In ARC, 64.4% of the examples were classified to be 'easy', compared to only 27.6% in CommonsenseQA. In MMLU, a significant 4.37% of the dataset was found to be data contaminated, while CommonsenseQA had none. Additionally, 12.57% of ARC and 4.56% of MMLU examples were identified as similar. These findings highlight that different datasets might suffer from different types of low-quality examples. For MMLU, a widely accepted dataset for measuring LLM performance, we were able to filter out 43% of the dataset while maintaining almost the same model ranking. This suggests that MMLU might not be saturated yet, and there is still room for further hill-climbing for current models.

While relative model ranking is maintained, SMART filtering leads to a notable decrease in inter-model agreement across all datasets, as mea-

[2]Because we excluded the 37 questions with 5 answer options instead of 4, ARC dataset size is 3530, not 3567.

[3]We used validation set as test set answers are not provided.

sured by the percentage of samples yielding identical predictions among models (Figure 4). This reduction in model agreement is beneficial, as it enables better differentiation among models based on their capabilities, and provides more headroom for improvement by decreasing their correlation.

In the following subsections, we present category-specific results for select datasets for detailed analysis, measure the correlation in model rankings before and after SMART filtering, and compare model accuracies with ChatBot Arena scores (Chiang et al., 2024).

4.1 Category-Wise Results

Many datasets have multiple sub-categories. For instance, MMLU comprises 57 categories such as ‘high school mathematics’, while ARC is divided into ARC-Easy and ARC-Challenge.

Our analysis reveals significant variation in the percentage of examples removed by SMART filtering across different categories in MMLU. Some categories appear to be more challenging than others. Over 60% of examples were removed in 9 categories. Notably, ‘high school government and politics’, ‘high school psychology’, ‘marketing’, and ‘sociology’ saw a reduction in size by 73%, 67%, 63%, and 62% respectively. A qualitative look into examples of these categories indicates they contain questions with well-known answers, making them easier for models and more susceptible to be present during training and possible data contamination, while less than 20% were removed in 13 categories. ‘Abstract algebra’, ‘global facts’, ‘high school physics’, and ‘professional law’ have only 4%, 4%, 5%, and 12.6% of examples removed, respectively. These subjects require complex reasoning and problem-solving, posing greater challenges for models. A detailed breakdown is shown in Appendix A.4.

State-of-the-art models often report accuracy on the ARC-Challenge category of ARC, as it is presumed to be more difficult (Meta AI, 2024; Google, 2024). Surprisingly, our analysis shows no significant difference between ARC-Easy and ARC-Challenge after SMART filtering. Both categories were significantly reduced, with 73% of ARC-Easy and 60% of ARC-Challenge examples removed. A high proportion, 55.1% of ARC-Challenge were deemed easy by all models.

Dataset	Pearson's Corr.		Kendall's τ
	Orig.	*SMART*	
ARC	0.783	0.845	0.951
MMLU	0.764	0.776	0.978
CommonsenseQA	0.666	0.660	0.965

Table 3: Pearson correlation is high between Elo scores on ChatBot Arena and model accuracies. Kendall's Tau correlation of model ranking on original and SMART filtering subsets is also high.

4.2 Correlation in Model Rankings

To quantitatively assess the consistency of model rankings between the original datasets and their filtered versions, we employed Kendall's Tau correlation coefficient (Kendall, 1938). This metric ranges from 1 (perfect positive correlation) to -1 (perfect negative correlation).

Our analysis of 29 models reveals high correlations between the ranking before and after SMART filtering. Table 3 shows the Kendall's Tau correlation coefficients for ARC, MMLU and CommonsenseQA were 0.951, 0.978, and 0.968, respectively. The main purpose of many evaluation datasets, particularly those used in leaderboards, is to compare relative model performance. Our approach shows similar model rankings while significantly reducing dataset size (as high as 68.9% for ARC) and increasing headroom.

4.3 Correlation between ChatBot Arena scores and Model Accuracy

ChatBot Arena (Chiang et al., 2024) is considered one of the most trusted current sources for assessing model performance (Saxon et al., 2024). They use human preference to assign an Elo score (Elo, 1967) for models on their leaderboard (Chiang et al., [2024]). Some researchers consider these scores a more accurate reflection of real-world model usage than traditional benchmarks (Thompson et al., 2020; Raji et al., 2021; Ott et al., 2022; Saxon et al., 2024). Developing high-correlation proxies for ChatBot Arena is valuable, as it could reduce the need for costly human preference collection (Ni et al., 2024).

We examine the relationship between model accuracies and ChatBot Arena Elo scores to analyze how well SMART filtered datasets reflect human preference metrics. Using Pearson correlation (Freedman et al., 2007), we measure the linear relationship between accuracies and Elo scores for 29 tested models. We use Pearson correlation, because it accounts for differences in Elo scores and not just relative model rankings.

Dataset Name	Original	SMART	Random	IRE
ARC	0.783	**0.845**	0.774	0.784
MMLU	0.764	**0.776**	0.767	0.766
CommonsenseQA	0.666	**0.660**	0.659	0.658

Table 4: Comparison of performance metrics across different methods. Random signifies random baseline as mentioned in Section 4.4.

Correlations are high across the board between ChatBot Arena Elo scores and the original versions of ARC ($r = 0.783$) and MMLU ($r = 0.764$); see Table 3. In fact, after SMART filtering the correlations *increase*, suggesting that filtered ARC ($r = 0.845$) and filtered MMLU ($r = 0.776$) are even better proxies than the original datasets for human preference rankings. This highlights the effectiveness of SMART filtering in both computational efficiency and alignment to human preferences.

4.4 Comparison with Other Approaches

In this section we compare SMART filtering with other similar approaches and a random baseline. Rodriguez et al. (2021b) introduce a method that filters examples based on difficulty and discriminability, aiming to make leaderboards more informative. For this baseline, we used 25th percentile thresholds for difficulty and discriminability criteria to establish this baseline. We call their method IRE. We also provide analysis with a random baseline where we created a random baseline by reducing each dataset to the same size as our SMART-filtered datasets (reducing dataset size randomly to 69%, 43% and 34% for ARC, MMLU and CommonsenseQA).

As seen in Table 4, SMART Filtering consistently achieves a higher correlation with human preferences than both the random baseline and IRE. Additionally, SMART Filtering consistently filters out a larger portion of the datasets compared to IRE, leading to greater efficiency in evaluation (c.f. Table 5).

Dataset Name	% of Examples Filtered	
	SMART	IRE
ARC	68.9	28.1
MMLU	43.0	31.1
CommonsenseQA	34.3	27.8

Table 5: Percentage of examples filtered by SMART vs. IRE for each dataset.

5 Robustness of our Approach

In this section, we show that SMART filtering is robust to altering settings, such as (i) the number of models used for identifying easy and data-contaminated examples and (ii) choice of embedding model used to find similar examples.

5.1 Ablation: Number of Models for Easy and Data-Contaminated Examples

In the SMART filtering methodology, we used 7 models for identifying easy and data-contaminated examples. Our selection was based on selecting the top 7 opensource models from different organizations. To assess the robustness of our approach, we conducted an ablation study varying the number of models used. We randomly selected subsets of 4, 5, and 6 models from the original set of 7. Specifically, we run our methodology on 10 random subsets of 4 models from a total of 35 subsets ($\binom{7}{4} = 35$), 10 random subsets of 5 models from a total of 21 subsets ($\binom{7}{5} = 21$) and all 7 subsets of 6 models ($\binom{7}{6} = 7$). The results, presented in Table 6, show that the percentage of examples filtered remains relatively stable across different model combinations for all three datasets. This shows that our methodology is largely insensitive to the number of models used, and can be effectively applied using any *n* top performing open-source models, again opening up possibilities for efficiency gains.

5.2 Embeddings for Similar Examples

We use SentenceBert embeddings to capture the meaning of each example (Reimers and Gurevych, 2019). The choice was motivated by its bi-directional attention that helps capture rich representations. Recent efforts have explored transforming decoder-only LMs into bi-directional text encoders, such as LLM2Vec (BehnamGhader et al., 2024). To validate the robustness of this approach regardless of embedding model choice, we compare the model we used, SentenceBERT, with another embedding method, LLM2Vec.

# of Models	% of Examples Filtered		
	ARC	*MMLU*	*CommonsenseQA*
4	72.9 ± 3.1	49.6 ± 3.0	41.2 ± 4.3
5	72.4 ± 1.1	46.1 ± 1.6	36.4 ± 1.4
6	69.7 ± 1.1	44.8 ± 1.1	34.6 ± 1.1

Table 6: Effect of varying number of models used for easy and data contamination filtering.

Embedding Pair	Percentage Overlap
SBert & Llama-3-8B	89.3
SBert & Mistral-7B	88.1

Table 7: Percentage of examples with extremely high semantic similarity of SentenceBert with LLM2Vec embedding, based on a distance in the embedding space smaller than a δ (see Section 3.4 for more on δ).

Our results in Table 7 show a high degree of similarity between the two embedding methods, with an average overlap of 88.7% between Sentence-BERT and different LLM models. This suggests comparable accuracy in identifying similar examples for both SentenceBert and LLM2Vec, aligning with findings in Freestone and Santu (2024) that LLM and BERT embeddings are similar. Given SentenceBert's widespread acceptance for clustering tasks and computational efficiency, we chose it for our SMART filtering methodology.

5.3 Quality of Filtering

To ensure that our method effectively filters out unwanted instances, we manually looked at 100 removed examples from the Easy and Similar examples category. For Similar examples, we found that 93 were genuinely similar, 5% were borderline similar and 2% were not similar. For Filtering Easy Examples, 97 examples could be easily answered by the authors or found via a simple web search. This analysis shows the effectiveness of our approach as most of the filtered out examples are unwanted for comparing model performance.

6 Discussion

Dataset Quality. Our approach removes the easy, data contaminated, and similar examples, thus improving the quality of the resulting dataset. Our methodology can be applied at various stages, including before or after dataset release. SMART filtering is designed to be iterative and scalable; we intend to periodically reapply SMART-Filtering to existing benchmarks to identify and filter out examples that become too easy or contaminated for newer models. We suspect that, as models become more capable, what constitutes an easy example could change.

Computational Efficiency. Recent studies have explored testing models on dataset subsets while maintaining high correlation with original dataset performance (Varshney et al., 2022). Our approach achieves substantial dataset size reductions, up to 68.9% for ARC, while preserving similar model rankings as shown in Tables 1 and 3. As dataset size correlates with computation time and evaluation costs, the 68.9% decrease in size should be reflected in similar decreases in time and evaluation cost. Our methodology thus offers significant time and cost savings for future benchmark evaluations, aligning with the growing need for efficient model assessment techniques.

Improved Correlation with ChatBot Arena. We find that SMART filtering can lead to better correlations with Elo scores from ChatBot Arena (Chiang et al., 2024). This suggests that model scores on SMART filtering subsets are more representative of real-world model performance. This finding highlights the potential for developing effective evaluation datasets that mimic real-world usage of the model without incurring substantial time and cost to capture human preferences.

Datasets are Not Yet Saturated. We found that model accuracies for all tested models drop significantly on datasets after SMART filtering, shown in Table 2. This indicates these datasets may not be ready for retirement yet, contrary to what leaderboards might suggest, and there remains room for improvement. Relatedly, iteratively applying SMART filtering over time can enable more dynamic evaluations (Dinan et al., 2019; Nie et al., 2020; Gehrmann et al., 2021; Kiela et al., 2021; Potts et al., 2021; Gehrmann et al., 2022; Margatina et al., 2023; Park et al., 2023; Graciotti et al., 2024), decreasing the impact of saturation by wringing more utility out of existing benchmarks.

7 Conclusion

In this work, we proposed SMART filtering, a methodology for identifying a challenging and high-quality subset of any benchmark, existing or new, that can better capture the capabilities of models and rank them. To achieve this we remove easy, data contaminated, and similar examples from a dataset. SMART filtering achieved significant increases in computational efficiency and better correlation with human preference data than the original datasets. We anticipate our approach will be useful for improving current benchmarking practices as well as for dataset creators to find high-quality subsets before dataset release in the future.

8 Limitations

In this work we present a methodology that can be applied to any NLP task, However, the method we have used for identifying data contaminated examples, may not be directly applicable to non-question answering datasets. Additionally, we tried to identify and remove incorrect ground truth annotations from the dataset (see Appendix for details). However, our initial attempt did not yield satisfactory results, highlighting the need for more effective strategies to address this challenge. Consequently, if a dataset contains a significant number of annotation errors, the proportion of such examples may increase in the resulting SMART filtering datasets.

References

01.AI, Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Tao Yu, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. 2024. Yi: Open foundation models by 01.ai. *Preprint*, arXiv:2403.04652.

Amro Abbas, Kushal Tirumala, Dániel Simig, Surya Ganguli, and Ari S Morcos. 2023. Semdedup: Data-efficient learning at web-scale through semantic deduplication. *Multimodal Representation Learning (MRL): Perks and Pitfalls*.

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Norah Alzahrani, Hisham Alyahya, Yazeed Alnumay, Sultan AlRashed, Shaykhah Alsubaie, Yousef Almushayqih, Faisal Mirza, Nouf Alotaibi, Nora Al-Twairesh, Areeb Alowisheq, M Saiful Bari, and Haidar Khan. 2024. When benchmarks are targets: Revealing the sensitivity of large language model leaderboards. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13787–13805, Bangkok, Thailand. Association for Computational Linguistics.

Nishant Balepur, Abhilasha Ravichander, and Rachel Rudinger. 2024. Artifacts or abduction: How do LLMs answer multiple-choice questions without the question? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10308–10330, Bangkok, Thailand. Association for Computational Linguistics.

Nishant Balepur and Rachel Rudinger. 2024. Is your large language model knowledgeable or a choices-only cheater? In *Proceedings of the 1st Workshop on Towards Knowledgeable Language Models (KnowLLM 2024)*, pages 15–26, Bangkok, Thailand. Association for Computational Linguistics.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. Llm2vec: Large language models are secretly powerful text encoders. *arXiv preprint arXiv:2404.05961*.

Ronan Le Bras, Swabha Swayamdipta, Chandra Bhagavatula, Rowan Zellers, Matthew Peters, Ashish Sabharwal, and Yejin Choi. 2020. Adversarial filters of dataset biases. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1078–1088. PMLR.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. 2024. Chatbot arena: An open platform for evaluating LLMs by human preference. In *Forty-first International Conference on Machine Learning*.

Chiang et al. [2024]. ChatBot Arena. `https://lmarena.ai/?leaderboard`.

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

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Databricks. 2024. Introducing DBRx: A New State-of-the-Art Open LLM.

Chunyuan Deng, Yilun Zhao, Xiangru Tang, Mark Gerstein, and Arman Cohan. 2023. Benchmark probing: Investigating data leakage in large language models. In *NeurIPS 2023 Workshop on Backdoors in Deep Learning-The Good, the Bad, and the Ugly*.

Emily Dinan, Samuel Humeau, Bharath Chintagunta, and Jason Weston. 2019. Build it break it fix it for dialogue safety: Robustness from adversarial human attack. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4537–4546, Hong Kong, China. Association for Computational Linguistics.

Michael Duan, Anshuman Suri, Niloofar Mireshghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh Hajishirzi. 2024. Do membership inference attacks work on large language models? *arXiv preprint arXiv:2402.07841*.

Liam Dugan, Daphne Ippolito, Arun Kirubarajan, Sherry Shi, and Chris Callison-Burch. 2023. Real or fake text?: Investigating human ability to detect boundaries between human-written and machine-generated text. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 12763–12771.

Aparna Elangovan, Jiayuan He, and Karin Verspoor. 2021. Memorization vs. generalization : Quantifying data leakage in NLP performance evaluation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1325–1335, Online. Association for Computational Linguistics.

Yanai Elazar, Akshita Bhagia, Ian Helgi Magnusson, Abhilasha Ravichander, Dustin Schwenk, Alane Suhr, Evan Pete Walsh, Dirk Groeneveld, Luca Soldaini, Sameer Singh, Hannaneh Hajishirzi, Noah A. Smith, and Jesse Dodge. 2024. What's in my big data? In *The Twelfth International Conference on Learning Representations*.

Arpad E Elo. 1967. The proposed uscf rating system, its development, theory, and applications. *Chess life*, 22(8):242–247.

David Freedman, Robert Pisani, and Roger Purves. 2007. Statistics (international student edition). *Pisani, R. Purves, 4th edn. WW Norton & Company, New York*.

Matthew Freestone and Shubhra Kanti Karmaker Santu. 2024. Word embeddings revisited: Do llms offer something new? *arXiv preprint arXiv:2402.11094*.

Sebastian Gehrmann, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Anuoluwapo Aremu, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna-Adriana Clinciu, Dipanjan Das, Kaustubh Dhole, Wanyu Du, Esin Durmus, Ondřej Dušek, Chris Chinenye Emezue, Varun Gangal, Cristina Garbacea, Tatsunori Hashimoto, Yufang Hou, Yacine Jernite, Harsh Jhamtani, Yangfeng Ji, Shailza Jolly, Mihir Kale, Dhruv Kumar, Faisal Ladhak, Aman Madaan, Mounica Maddela, Khyati Mahajan, Saad Mahamood, Bodhisattwa Prasad Majumder, Pedro Henrique Martins, Angelina McMillan-Major, Simon Mille, Emiel van Miltenburg, Moin Nadeem, Shashi Narayan, Vitaly Nikolaev, Andre Niyongabo Rubungo, Salomey Osei, Ankur Parikh, Laura Perez-Beltrachini, Niranjan Ramesh Rao, Vikas Raunak, Juan Diego Rodriguez, Sashank Santhanam, João Sedoc, Thibault Sellam, Samira Shaikh, Anastasia Shimorina, Marco Antonio Sobrevilla Cabezudo, Hendrik Strobelt, Nishant Subramani, Wei Xu, Diyi Yang, Akhila Yerukola, and Jiawei Zhou. 2021. The GEM benchmark: Natural language generation, its evaluation and metrics. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 96–120, Online. Association for Computational Linguistics.

Sebastian Gehrmann, Abhik Bhattacharjee, Abinaya Mahendiran, Alex Wang, Alexandros Papangelis, Aman Madaan, Angelina Mcmillan-major, Anna Shvets, Ashish Upadhyay, Bernd Bohnet, Bingsheng Yao, Bryan Wilie, Chandra Bhagavatula, Chaobin You, Craig Thomson, Cristina Garbacea, Dakuo Wang, Daniel Deutsch, Deyi Xiong, Di Jin, Dimitra Gkatzia, Dragomir Radev, Elizabeth Clark, Esin Durmus, Faisal Ladhak, Filip Ginter, Genta Indra Winata, Hendrik Strobelt, Hiroaki Hayashi, Jekaterina Novikova, Jenna Kanerva, Jenny Chim, Jiawei Zhou, Jordan Clive, Joshua Maynez, João Sedoc, Juraj Juraska, Kaustubh Dhole, Khyathi Raghavi Chandu, Laura Perez Beltrachini, Leonardo F . R. Ribeiro, Lewis Tunstall, Li Zhang, Mahim Pushkarna, Mathias Creutz, Michael White, Mihir Sanjay Kale, Moussa Kamal Eddine, Nico Daheim, Nishant Subramani, Ondrej Dusek, Paul Pu Liang, Pawan Sasanka Ammanamanchi, Qi Zhu, Ratish Puduppully, Reno Kriz, Rifat Shahriyar, Ronald Cardenas, Saad Mahamood, Salomey Osei, Samuel Cahyawijaya, Sanja Štajner, Sebastien Montella, Shailza Jolly, Simon Mille, Tahmid Hasan, Tianhao Shen, Tosin Adewumi, Vikas Raunak, Vipul Raheja, Vitaly Nikolaev, Vivian Tsai, Yacine Jernite, Ying Xu, Yisi Sang, Yixin Liu, and Yufang Hou. 2022. GEMv2: Multilingual NLG benchmarking in a single line of code. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 266–281, Abu Dhabi, UAE. Association for Computational Linguistics.

Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, et al. 2024. Are we done with mmlu? *arXiv preprint arXiv:2406.04127*.

Google. 2024. Google Gemma 2: A New AI Model.

Arianna Graciotti, Valentina Presutti, and Rocco Tripodi. 2024. Latent vs explicit knowledge representation: How ChatGPT answers questions about low-frequency entities. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10172–10185, Torino, Italia. ELRA and ICCL.

Vipul Gupta, David Pantoja, Candace Ross, Adina Williams, and Megan Ung. 2024. Changing answer order can decrease MMLU accuracy. *arXiv preprint arXiv:2406.19470*.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of* the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. 2020. Aligning ai with shared human values. *arXiv preprint arXiv:2008.02275*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring massive multitask language understanding. *Preprint*, arXiv:2009.03300.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Marek Herde, Denis Huseljic, Bernhard Sick, and Adrian Calma. 2021. A survey on cost types, interaction schemes, and annotator performance models in selection algorithms for active learning in classification. *IEEE Access*, 9:166970–166989.

HuggingFace. [2024]. Open LLM Leaderboard. https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard.

Ganesh Jawahar, Muhammad Abdul-Mageed, and Laks Lakshmanan, V.S. 2020. Automatic detection of machine generated text: A critical survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2296–2309, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Minhao Jiang, Ken Ziyu Liu, Ming Zhong, Rylan Schaeffer, Siru Ouyang, Jiawei Han, and Sanmi Koyejo. 2024. Investigating data contamination for pre-training language models. *arXiv preprint arXiv:2401.06059*.

Jon Keegan. 2024. Everyone is judging ai by these tests, but experts say they're close to meaningless. Accessed: [Insert Date].

Maurice G Kendall. 1938. A new measure of rank correlation. *Biometrika*, 30(1-2):81–93.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. 2021. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4110–4124, Online. Association for Computational Linguistics.

Md Tahmid Rahman Laskar, M Saiful Bari, Mizanur Rahman, Md Amran Hossen Bhuiyan, Shafiq Joty, and Jimmy Huang. 2023. A systematic study and comprehensive evaluation of ChatGPT on benchmark datasets. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 431–469, Toronto, Canada. Association for Computational Linguistics.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2022. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8424–8445, Dublin, Ireland. Association for Computational Linguistics.

Ruizhe Li and Yanjun Gao. 2024. Anchored answers: Unravelling positional bias in gpt-2's multiple-choice questions. *arXiv preprint arXiv:2405.03205*.

Shayne Longpre, Robert Mahari, Ariel Lee, Campbell Lund, Hamidah Oderinwale, William Brannon, Nayan Saxena, Naana Obeng-Marnu, Tobin South, Cole Hunter, et al. 2024. Consent in crisis: The rapid decline of the ai data commons. *arXiv preprint arXiv:2407.14933*.

Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. 2024. Fine-tuning llama for multi-stage text retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2421–2425.

Inbal Magar and Roy Schwartz. 2022. Data contamination: From memorization to exploitation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 157–165, Dublin, Ireland. Association for Computational Linguistics.

Katerina Margatina, Shuai Wang, Yogarshi Vyas, Neha Anna John, Yassine Benajiba, and Miguel Ballesteros. 2023. Dynamic benchmarking of masked language models on temporal concept drift with multiple views. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2881–2898, Dubrovnik, Croatia. Association for Computational Linguistics.

Hana Matatov, Mor Naaman, and Ofra Amir. 2022. Dataset and case studies for visual near-duplicates detection in the context of social media. *arXiv preprint arXiv:2203.07167*.

Reid McIlroy-Young, Katrina Brown, Conlan Olson, Linjun Zhang, and Cynthia Dwork. 2024. Set-based prompting: Provably solving the language model order dependency problem. *Preprint*, arXiv:2406.06581.

Meta AI. 2024. Meta LLaMA 3.1. https://ai.meta.com/blog/meta-llama-3-1/. Accessed: [Insert Date].

Swaroop Mishra and Bhavdeep Singh Sachdeva. 2020. Do we need to create big datasets to learn a task? In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 169–173, Online. Association for Computational Linguistics.

Mistral AI. 2023. Mixtral of Experts. Section on Mixtral Architecture. Accessed: [Insert Date].

Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, et al. 2022. Text and code embeddings by contrastive pre-training. *arXiv preprint arXiv:2201.10005*.

Jinjie Ni, Fuzhao Xue, Xiang Yue, Yuntian Deng, Mahir Shah, Kabir Jain, Graham Neubig, and Yang You. 2024. Mixeval: Deriving wisdom of the crowd from llm benchmark mixtures. *arXiv preprint arXiv:2406.06565*.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Simon Ott, Adriano Barbosa-Silva, Kathrin Blagec, Jan Brauner, and Matthias Samwald. 2022. Mapping global dynamics of benchmark creation and saturation in artificial intelligence. *Nature Communications*, 13(1):6793.

Chanjun Park, Hyeonseok Moon, Seolhwa Lee, Jaehyung Seo, Sugyeong Eo, and Heuiseok Lim. 2023. Self-improving-leaderboard (sil): A call for real-world centric natural language processing leaderboards. *arXiv preprint arXiv:2303.10888*.

Pouya Pezeshkpour and Estevam Hruschka. 2024. Large language models sensitivity to the order of options in multiple-choice questions. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2006–2017, Mexico City, Mexico. Association for Computational Linguistics.

Jason Phang, Angelica Chen, William Huang, and Samuel R. Bowman. 2022. Adversarially constructed evaluation sets are more challenging, but may not be fair. In *Proceedings of the First Workshop on Dynamic Adversarial Data Collection*, pages 62–62, Seattle, WA. Association for Computational Linguistics.

Christopher Potts, Zhengxuan Wu, Atticus Geiger, and Douwe Kiela. 2021. DynaSent: A dynamic benchmark for sentiment analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2388–2404, Online. Association for Computational Linguistics.

Harsh Raj, Vipul Gupta, Domenic Rosati, and Subhabrata Majumdar. 2023. Semantic consistency for assuring reliability of large language models. *arXiv preprint arXiv:2308.09138*.

Deborah Raji, Emily Denton, Emily M. Bender, Alex Hanna, and Amandalynne Paullada. 2021. Ai and the everything in the whole wide world benchmark. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.

Mathieu Ravaut, Bosheng Ding, Fangkai Jiao, Hailin Chen, Xingxuan Li, Ruochen Zhao, Chengwei Qin, Caiming Xiong, and Shafiq Joty. 2024. How much are llms contaminated? a comprehensive survey and the llmsanitize library. *arXiv preprint arXiv:2404.00699*.

Yuval Reif and Roy Schwartz. 2024. Beyond performance: Quantifying and mitigating label bias in LLMs. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6784–6798, Mexico City, Mexico. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2023. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*.

Pedro Rodriguez, Joe Barrow, Alexander Miserlis Hoyle, John P. Lalor, Robin Jia, and Jordan Boyd-Graber. 2021a. Evaluation examples are not equally informative: How should that change NLP leaderboards? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4486–4503, Online. Association for Computational Linguistics.

Pedro Rodriguez, Joe Barrow, Alexander Miserlis Hoyle, John P Lalor, Robin Jia, and Jordan Boyd-Graber. 2021b. Evaluation examples are not equally informative: How should that change nlp leaderboards? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4486–4503.

Candace Ross, Melissa Hall, Adriana Romero-Soriano, and Adina Williams. 2024. What makes a good metric? evaluating automatic metrics for text-to-image consistency. In *First Conference on Language Modeling*.

Paul Röttger, Valentin Hofmann, Valentina Pyatkin, Musashi Hinck, Hannah Rose Kirk, Hinrich Schütze, and Dirk Hovy. 2024. Political compass or spinning arrow? towards more meaningful evaluations for values and opinions in large language models. *arXiv preprint arXiv:2402.16786*.

Michael Saxon, Ari Holtzman, Peter West, William Yang Wang, and Naomi Saphra. 2024. Benchmarks as microscopes: A call for model metrology. In *First Conference on Language Modeling*.

Krunal Shah, Nitish Gupta, and Dan Roth. 2020. What do we expect from multiple-choice QA systems? In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3547–3553, Online. Association for Computational Linguistics.

Aaditya K Singh, Muhammed Yusuf Kocyigit, Andrew Poulton, David Esiobu, Maria Lomeli, Gergely Szilvasy, and Dieuwke Hupkes. 2024. Evaluation data contamination in llms: how do we measure it and (when) does it matter? *arXiv preprint arXiv:2411.03923*.

Koustuv Sinha, Prasanna Parthasarathi, Joelle Pineau, and Adina Williams. 2021. UnNatural Language Inference. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7329–7346, Online. Association for Computational Linguistics.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2023. One embedder, any task: Instruction-finetuned text embeddings. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1102–1121, Toronto, Canada. Association for Computational Linguistics.

Saku Sugawara, Pontus Stenetorp, Kentaro Inui, and Akiko Aizawa. 2020. Assessing the benchmarking capacity of machine reading comprehension datasets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8918–8927.

Haozhan Sun, Chenchen Xu, and Hanna Suominen. 2021. Analyzing the granularity and cost of annotation in clinical sequence labeling. *arXiv preprint arXiv:2108.09913*.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, Online. Association for Computational Linguistics.

Zeerak Talat, Hagen Blix, Josef Valvoda, Maya Indira Ganesh, Ryan Cotterell, and Adina Williams. 2022. On the machine learning of ethical judgments from natural language. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 769–779, Seattle, United States. Association for Computational Linguistics.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

William Hedley Thompson, Jessey Wright, Patrick G Bissett, and Russell A Poldrack. 2020. Dataset decay and the problem of sequential analyses on open datasets. *Elife*, 9:e53498.

Marcos Treviso, Ji-Ung Lee, Tianchu Ji, Betty van Aken, Qingqing Cao, Manuel R. Ciosici, Michael Hassid, Kenneth Heafield, Sara Hooker, Colin Raffel, Pedro H. Martins, André F. T. Martins, Jessica Zosa Forde, Peter Milder, Edwin Simpson, Noam Slonim, Jesse Dodge, Emma Strubell, Niranjan Balasubramanian, Leon Derczynski, Iryna Gurevych, and Roy Schwartz. 2023. Efficient methods for natural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 11:826–860.

Clara Vania, Phu Mon Htut, William Huang, Dhara Mungra, Richard Yuanzhe Pang, Jason Phang, Haokun Liu, Kyunghyun Cho, and Samuel R. Bowman. 2021. Comparing test sets with item response theory. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1141–1158, Online. Association for Computational Linguistics.

Neeraj Varshney, Swaroop Mishra, and Chitta Baral. 2022. ILDAE: Instance-level difficulty analysis of evaluation data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3412–3425, Dublin, Ireland. Association for Computational Linguistics.

Rajan Vivek, Kawin Ethayarajh, Diyi Yang, and Douwe Kiela. 2024. Anchor points: Benchmarking models with much fewer examples. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1576–1601, St. Julian's, Malta. Association for Computational Linguistics.

Haochun Wang, Sendong Zhao, Zewen Qiang, Bing Qin, and Ting Liu. 2024a. Beyond the answers: Reviewing the rationality of multiple choice question answering for the evaluation of large language models. *arXiv preprint arXiv:2402.01349*.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. 2024b. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*.

Sheng-Lun Wei, Cheng-Kuang Wu, Hen-Hsen Huang, and Hsin-Hsi Chen. 2024. Unveiling selection biases: Exploring order and token sensitivity in large language models. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 5598–5621, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Hanzi Xu, Renze Lou, Jiangshu Du, Vahid Mahzoon, Elmira Talebianaraki, Zhuoan Zhou, Elizabeth Garrison, Slobodan Vucetic, and Wenpeng Yin. 2024. Llms' classification performance is overclaimed. *arXiv preprint arXiv:2406.16203*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2024. Large language models are not robust multiple choice selectors. In *The Twelfth International Conference on Learning Representations*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

Yongshuo Zong, Yist Tingyang YU, Bingchen Zhao, Ruchika Chavhan, and Timothy Hospedales. 2024. Fool your large (vision and) language models with embarrassingly simple permutations.

A Appendix

A.1 Finding Threshold for Similar Examples

To identify similar examples, we analyze cosine distances between SentenceBert embeddings of the examples and identify the first local maxima from the distribution (see Figure 3). Notably, we found that considering only the 100 nearest neighbors for each example is sufficient to determine the threshold accurately (Figure 5). Our analysis, detailed in the Appendix, shows that threshold values obtained using different k values for k-nearest neighbors converge after 100 neighbors, aligning closely with results from the entire dataset. For computational efficiency, we therefore use 100 nearest neighbors to establish the threshold value.

A.2 Attempt to Identify Wrong Ground Truth

Benchmark datasets often contain wrong ground truths. In an attempt to identify such examples, we tried an algorithmic framework. We hypothesized that if all top-performing models icorrectly predicted an example with very high confidence (>0.8), then that example would likely be wrongly annotated. However, upon applying this method to the MMLU dataset, our hypothesis didn't work well. Approximately 1.4% of MMLU examples were filtering using this criteria, but manual inspection revealed that nearly half of these flagged examples actually had correct ground truth annotations, despite unanimous high-confidence wrong predictions by the models. Thus we didn't include this step in SMART filtering as it would require human involvement and may not provide sufficient efficiency gains to justify the additional resource expenditure.

A.3 Exact Match Duplicates

Dataset creators often preprocess their datasets to remove duplicate examples before releasing them. Nevertheless, exact-match duplicates can still be present. Our analysis reveals that 1.2% of the MMLU dataset consists of identical questions, while ARC contains 0.2% duplicates. In contrast, CommonsenseQA does not have any exact duplicates, with a rate of 0%.

A.4 Category Wise Results

MMLU has 57 different categories and we found that different categories in MMLU are affected differently by SMART filtering. The percentage of examples removed by SMART filtering for each category is given in Table 9.

A.5 LLM Based Embedding for Similar Examples

In addition to using LLM2Vec embeddings, we also used the last token's LLM embeddings as representation of entire sentences (Neelakantan et al., 2022; Ma et al., 2024) The results as shown in Table 8, shows that SentenceBert embeddings are very similar to LLM based embeddings for identifying and removing similar examples.

Embedding Pair	Percentage Overlap
SBert and Llama-3-70B	94.7
SBert and Qwen2-72B	95.2
SBert and Llama-3-8B	95.8

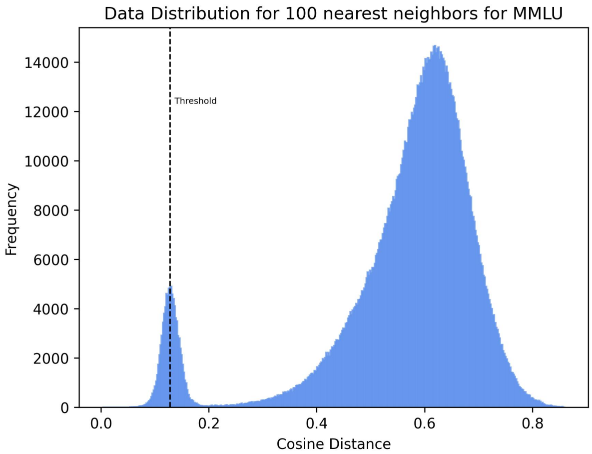
Table 8: Percentage of examples with extremely high semantic similarity, based on a distance in the embedding space smaller than a δ. We define the process for computing δ in Section 3.4.

A.6 Computation Resources

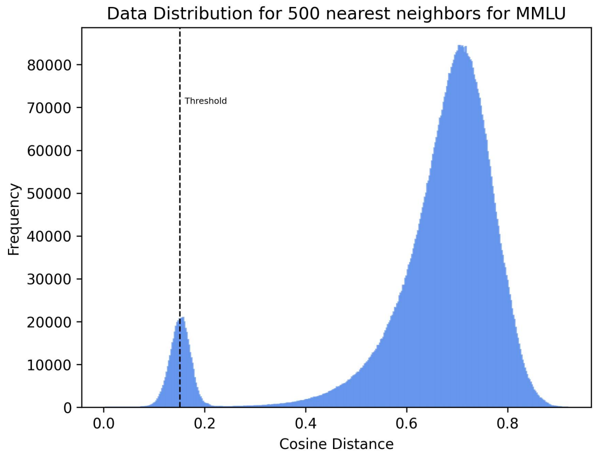
For all experiments for this work, we utilized A100 80GB GPUs. Depending on the model evaluated, we used 1,2,3 or 4 GPUs for inferences. These GPUs were assembled in a cluster of 8 GPUs in a node. The cumulative computing time required to evaluate all the language models and complete the experiments amounted to approximately 2000 GPU hours. We also used LLMs for coding assistance for building our codebase.

A.7 Detailed Model Results

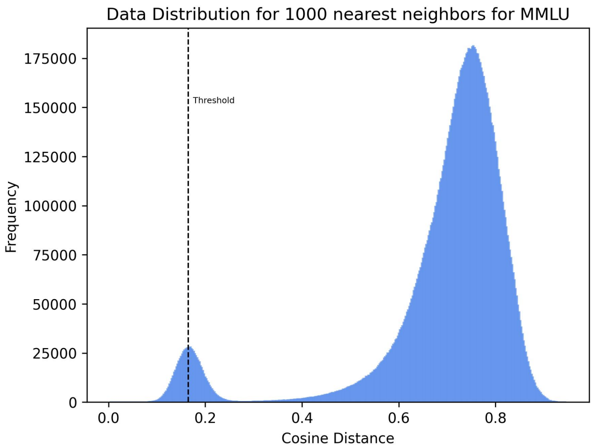
Table 10, 11, and 12 shows accuracy for all tested models on ARC, MMLU, and CommonsenseQA dataset and their SMART filtered versions.



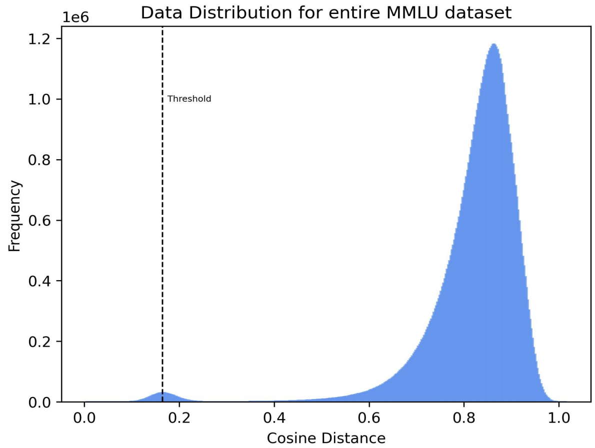
(a) k=100



(b) k=500



(c) k=1000



(d) k=5000

Figure 5: Distribution of cosine distances between SentenceBert embeddings for different values of k nearest neighbors for each example in the MMLU dataset. We find that threshold values obtained using different k values for k-nearest neighbors converge after 100 neighbors.

Category Name	Original # of examples	SMART filtered examples	Percentage of examples removed
Moral Scenarios	895	0	100.00%
High School Government And Politics	193	52	73.06%
Us Foreign Policy	100	31	69.00%
High School Psychology	545	182	66.61%
Miscellaneous	783	274	65.01%
Marketing	234	86	63.25%
Sociology	201	76	62.19%
International Law	121	46	61.98%
High School World History	237	91	61.60%
World Religions	171	70	59.06%
High School Us History	204	85	58.33%
High School Geography	198	83	58.08%
Medical Genetics	100	43	57.00%
High School Biology	310	134	56.77%
High School European History	165	72	56.36%
Logical Fallacies	163	72	55.83%
Management	103	46	55.34%
College Medicine	173	80	53.76%
Prehistory	324	155	52.16%
College Biology	144	70	51.39%
High School Microeconomics	238	121	49.16%
Clinical Knowledge	265	136	48.68%
Jurisprudence	108	56	48.15%
Astronomy	152	79	48.03%
Security Studies	245	129	47.35%
Computer Security	100	54	46.00%
Human Sexuality	131	72	45.04%
High School Computer Science	100	55	45.00%
Professional Psychology	612	337	44.93%
Nutrition	306	171	44.12%
High School Macroeconomics	390	228	41.54%
Public Relations	110	67	39.09%
Human Aging	223	138	38.12%
Business Ethics	100	62	38.00%
Professional Medicine	272	173	36.40%
Philosophy	311	199	36.01%
Moral Disputes	346	223	35.55%
Anatomy	135	89	34.07%
Conceptual Physics	235	163	30.64%
College Physics	102	75	26.47%
Electrical Engineering	145	107	26.21%
Virology	166	127	23.49%
High School Chemistry	203	158	22.17%
College Chemistry	100	79	21.00%
High School Statistics	216	173	19.91%
Machine Learning	112	90	19.64%
Econometrics	114	93	18.42%
College Computer Science	100	83	17.00%
Professional Accounting	282	244	13.48%
Elementary Mathematics	378	330	12.70%
Formal Logic	126	110	12.70%
Professional Law	1534	1340	12.65%
High School Physics	151	143	5.30%
High School Mathematics	270	259	4.07%
Abstract Algebra	100	96	4.00%
Global Facts	100	96	4.00%
College Mathematics	100	97	3.00%

Table 9: Results of each of the 57 categories of SMART filtering on MMLU dataset. Different categories are affected differently.

Model Name	Accuracy on ARC	Accuracy on ARC-SMART
Qwen2-72B-Instruct	0.939	0.83
Meta-Llama-3.1-70B-Instruct	0.934	0.819
Meta-Llama-3-70B-Instruct	0.933	0.819
gemma-2-27b-it	0.925	0.788
Phi-3-medium-4k-instruct	0.923	0.781
Phi-3.5-MoE-instruct	0.92	0.785
Mixtral-8x22B-Instruct-v0.1	0.916	0.762
gemma-2-9b-it	0.91	0.757
Yi-34B-Chat	0.909	0.745
Qwen1.5-32B-Chat	0.907	0.752
dbrx-instruct	0.905	0.732
Yi-34B	0.902	0.724
Yi-1.5-9B-Chat	0.894	0.728
Meta-Llama-3-8B-Instruct	0.883	0.721
Qwen2-7B-Instruct	0.882	0.697
Mixtral-8x7B-Instruct-v0.1	0.876	0.681
Mixtral-8x7B-v0.1	0.876	0.688
internlm2_5-20b-chat	0.875	0.675
internlm2_5-7b-chat	0.861	0.647
Llama-2-70b-hf	0.855	0.644
gemma-7b	0.837	0.611
Mistral-7B-v0.3	0.802	0.563
Mistral-7B-Instruct-v0.2	0.793	0.581
Qwen-7B-Chat	0.764	0.518
gemma-7b-it	0.738	0.531
Qwen-7B	0.733	0.476
falcon-40b	0.722	0.51
falcon-40b-instruct	0.722	0.507
OLMo-1.7-7B-hf	0.639	0.435

Table 10: Accuracy of different models on ARC and ARC-SMART filtering datasets. Almost all the models have similar ranking among the two versions of ARC.

Model Name	Accuracy on MMLU	Accuracy on MMLU-SMART
Qwen2-72B-Instruct	0.841	0.743
Meta-Llama-3.1-70B-Instruct	0.823	0.714
Meta-Llama-3-70B-Instruct	0.803	0.692
Phi-3.5-MoE-instruct	0.787	0.67
Phi-3-medium-4k-instruct	0.781	0.656
Mixtral-8x22B-Instruct-v0.1	0.779	0.653
Yi-1.5-34B-Chat	0.764	0.634
gemma-2-27b-it	0.762	0.639
Yi-34B	0.758	0.624
Qwen1.5-32B-Chat	0.754	0.615
Yi-34B-Chat	0.747	0.603
dbrx-instruct	0.732	0.6
gemma-2-9b-it	0.725	0.588
internlm2_5-7b-chat	0.711	0.568
Mixtral-8x7B-Instruct-v0.1	0.706	0.565
Mixtral-8x7B-v0.1	0.704	0.568
Qwen2-7B-Instruct	0.702	0.564
internlm2_5-20b-chat	0.701	0.567
Yi-1.5-9B-Chat	0.699	0.556
Llama-2-70b-hf	0.69	0.544
Meta-Llama-3-8B-Instruct	0.664	0.505
gemma-7b	0.644	0.492
Mistral-7B-v0.3	0.626	0.468
Mistral-7B-Instruct-v0.2	0.593	0.441
Qwen-7B	0.574	0.426
Qwen-7B-Chat	0.563	0.415
falcon-40b	0.558	0.412
falcon-40b-instruct	0.547	0.402
OLMo-1.7-7B-hf	0.521	0.381
gemma-7b-it	0.517	0.389

Table 11: Model Accuracy Comparison on MMLU and MMLU-SMART filtering. Almost all the models have similar ranking among the two versions of MMLU.

Model Name	Accuracy on CommonsenseQA	Accuracy on CommonsenseQA-SMART
Qwen2-72B-Instruct	0.885	0.845
Yi-1.5-34B-Chat	0.833	0.776
Meta-Llama-3-70B-Instruct	0.833	0.771
Qwen1.5-32B-Chat	0.830	0.767
Meta-Llama-3.1-70B-Instruct	0.815	0.741
Phi-3.5-MoE-instruct	0.809	0.739
gemma-2-9b-it	0.808	0.733
Qwen2-7B-Instruct	0.804	0.724
Phi-3-medium-4k-instruct	0.797	0.722
Yi-34B	0.797	0.718
gemma-2-27b-it	0.794	0.719
Yi-34B-Chat	0.791	0.712
Yi-1.5-9B-Chat	0.790	0.718
internlm2_5-7b-chat	0.789	0.714
dbrx-instruct	0.786	0.704
internlm2_5-20b-chat	0.777	0.695
Meta-Llama-3-8B-Instruct	0.765	0.680
Mixtral-8x22B-Instruct-v0.1	0.759	0.672
OLMo-1.7-7B-hf	0.739	0.670
Mixtral-8x7B-Instruct-v0.1	0.706	0.600
gemma-7b-it	0.688	0.594
Qwen-7B	0.683	0.586
Mistral-7B-Instruct-v0.2	0.682	0.590
falcon-40b-instruct	0.678	0.579
Qwen-7B-Chat	0.652	0.557
gemma-7b	0.646	0.551
Mistral-7B-v0.3	0.603	0.499
Llama-2-70b-hf	0.583	0.465
Mixtral-8x7B-v0.1	0.581	0.468
falcon-40b	0.555	0.446

Table 12: Accuracy comparison of models on CommonsenseQA and CommonsenseQA-SMART filtering. Almost all the models have similar ranking among the two versions of CommonsenseQA.