Examining the robustness of LLM evaluation to the distributional assumptions of benchmarks

Anonymous ACL submission

Abstract

Benchmarks have emerged as the central approach for evaluating Large Language Models (LLMs). The research community often relies on a model's average performance across the test prompts of a benchmark to evaluate the model's performance. This is consistent with the assumption that the test prompts within a benchmark represent a random sample from some real-world distribution of interest. We note that this is generally not the case; instead, we hold that the distribution of interest varies according to the specific use case. Hence, we analyze the robustness of LLM benchmarks to their underlying distributional assumptions. We find that (1) the correlation in model performance across test prompts is non-random, (2) accounting for correlations across test prompts can change model rankings on major benchmarks, (3) explanatory factors for these correlations include semantic similarity and common LLM failure points.

1 Introduction

Since the introduction of the Transformer architecture (Vaswani et al., 2017), Large Language Models (LLMs) have progressed into sophisticated systems with an outstanding ability to comprehend and generate text that mimic human language. Notable models in this domain include ChatGPT[1], utilizing the GPT-3.5-TURBO or GPT-4 architectures[2], LLaMA (Touvron et al., 2023), ChatGLM (Zeng et al., 2023), Alpaca (Taori et al., 2023), and Falcon (Penedo et al., 2023).

Due to their effectiveness, LLMs are becoming very popular in both academia and industry, making their evaluation crucial. However, this effectiveness comes at the cost of increased complexity, which makes their evaluation very challenging. Although prior research has introduced benchmarks for different tasks along with evaluation measures, these assessments often overlook potential biases. When a benchmark includes multiple prompts with similar characteristics, it can increase or decrease the average performance of a model, so model comparisons can become brittle with respect to benchmark composition. In this work, we show that the inherent connections between the prompts in current benchmarks impact the models' performance and their relative rankings.

The standard approach for evaluation on a benchmark is to (i) obtain model responses for each prompt in the benchmark, (ii) compute the performance metrics for each response, (iii) aggregate (usually average) the performance metrics to obtain a single performance metric over the benchmark, and (iv) compare models by comparing their aggregate performance.

When aggregating performance metrics in step iii above, each prompt is generally weighted equally (Yang and Menczer, 2023; Peña et al., 2023). However, using equal weights reflects the assumption that prompts in the benchmark are "equal", in the sense that prompts are representative samples of a target distribution of interest. In the case of LLMs, the notion of a target distribution (i.e., the distribution of all possible prompts for a given use case) is usually not well-defined. For example, different Natural Language Inference (NLI) applications may have very different target distributions, and we should not expect a single benchmark to capture every one. Therefore, one must ask: *What distribution do the prompts in the benchmark represent?* Would considering different distributions fundamentally change model comparisons? In this work, we present a novel approach to assess the robustness and adequacy of benchmarks used in evaluating LLMs, by analyzing the performance of multiple LLMs on a set of four major benchmarks.

[1]New chat: https://chat.openai.com/

[2]Models - OpenAI API: https://platform.openai.com/docs/models/

Our key contributions are outlined below:

1. For each considered benchmark, we observe that the correlation of model performance across prompts is significant (p-value < 0.05). This demonstrates the existence of relationships between prompts within the investigated benchmarks.

2. We explore the robustness of model comparisons to different distributional assumptions based on correlation structure, and we observe shifts in performance as large as 10% and rank changes as large as 5 (out of 14 models).

3. We provide a characterization of performance over the distribution of all possible prompt weights. This constitutes a robustness check that can be incorporated in comparative studies.

4. We show that model performance similarity across prompts can be explained by semantic similarity, but it is most likely derived by common failure points of the LLM.

2 Related work

Evaluating the performance of LLMs has become a critical area of research, drawing significant attention in recent years. Comprehensive surveys of LLM evaluation can be found in Chang et al. (2023); Guo et al. (2023), and Liang et al. (2022).

When assessing the quality of LLMs, the robustness aspect is becoming of increasing importance (Wang et al., 2022; Goel et al., 2021). Robustness investigates the stability of a model when confronted with unforeseen prompts. Robustness research can be divided into three main lines of work (Li et al., 2023): (i) robustness under distribution shift (Wang et al., 2021; Yang et al., 2023), (ii) robustness to adversarial input (Zhu et al., 2023; Wang et al., 2023a), and (iii) robustness to dataset bias (Gururangan et al., 2018; Le Bras et al., 2020; Niven and Kao, 2019). Our work falls into the latter category.

Reducing bias on benchmarks is a long-standing area of research spanning many diverse fields. Applications range from weighing survey responses to match a target population (DeBell, 2018), to accounting for language biases in visual question-answering (Goyal et al., 2017). In the context of NLI, researchers have looked into improving the quality of prompts in order to mitigate certain types of biases. Work in this area has focused on determining the quality of prompts by generating optimal prompts (Pryzant et al., 2023; Deng et al., 2022) or by clustering prompts based on semantic similarity (Kuhn et al., 2023). Additionally, researchers have investigated data leakage between benchmarks and LLM training data (Zhou et al., 2023; Oren et al., 2023).

Limited research has been conducted to study inherent biases in LLM benchmarks. Among existing works, Gururangan et al. (2018) and Niven and Kao (2019) have shown that models leverage spurious statistical relationships in the benchmark datasets and, thus, their performance on the benchmarks is overestimated. In the same spirit, Le Bras et al. (2020) propose to investigate AFLITE (Sakaguchi et al., 2019), an iterative approach to filter datasets by removing biased data points to mitigate overestimation of language models' performance. More recently, Alzahrani et al. (2024) show that performance of LLMs is highly sensitive to minor changes in benchmarks with multiple-choice questions.

Our work is orthogonal yet complementary to previous work. In particular, we propose a new method to identify biases in a benchmark by looking at the performance of multiple recent LLMs on that benchmark. We show that similarity in performance correlates with similarity in prompts. To the best of our knowledge, our work is the first approaching benchmark biases by analyzing and leveraging the performance of a collection of models on a set of major benchmarks; as well as investigating the impact of inherent distributional biases in benchmarks used on LLM comparative studies.

3 Proposed method

In this section, we outline the problem setup and introduce the notation and expressions that will be employed throughout the paper. Second, we present the approach to evaluate whether relationships between prompts (based on models' performance) are statistically non-random. Furthermore, we describe our method for analyzing how sensitive model comparisons are with respect to different distributional assumptions of the benchmark. Finally, we present our proposed methodology for exploring the origins of relationships between prompt performance vectors.

3.1 Problem setup

Consider a benchmark containing n prompts $\{p_1, \ldots, p_n\}$, and a set of k LLMs $\{m_1, \ldots, m_k\}$ being evaluated. We define the performance matrix Q as an $n \times k$ matrix, where every cell $Q[i, j]$ rep-

resents the performance of model m_j on prompt p_i. We refer to the i-th row of that matrix, $\mathbf{q}_i$, as a *performance vector* for prompt p_i. To measure how similar two prompts are with respect to model performance, we compute the similarity between their performance vectors $s_{perf}(p_i, p_j) := s(\mathbf{q}_i, \mathbf{q}_j)$, where $s(\cdot, \cdot)$ is a similarity function. Here, we consider cosine, Jaccard, and Hamming similarity. Given a performance matrix Q and a similarity function s, we compute a $n \times n$ similarity matrix $T_s(Q)$, where every cell $T[i, j]$ is the performance similarity for prompts p_i, p_j: $T[i, j] = s_{perf}(p_i, p_j)$.

Semantic meaning from text is commonly understood through the use of embeddings. An embedding of a prompt is a numerical vector that contains the learned representations of semantic meaning. Measuring semantic similarity between two prompts is achieved by measuring the distance between their embeddings. In this paper, we use ada-2 embeddings from OpenAI[3]. For a set of prompts $\{p_1, \ldots, p_n\}$, we compute a matrix of embeddings $E = \{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$. E is a $n \times s$ matrix, where s is the size of the embedding vectors. To measure semantic similarity between pairs of prompts, we compute similarity metrics between the corresponding rows: $s_{sem}(p_i, p_j) = s(\mathbf{e}_i, \mathbf{e}_j)$.

3.2 Determining if performance vectors are correlated

Given a benchmark, we assess whether the observed similarity among performance vectors is significant. If the observed similarity is significantly high, this implies the existence of specific connections between prompts. These connections lead to similar model behavior when responding to these prompts.

To test this hypothesis, we perform permutation tests. We generate permutations of the performance matrix Q by randomly shuffling the cells of each column. In this way, we permute the values of the model responses across prompts, while holding constant the overall performance of each model (i.e., the column averages of Q). We then compute a similarity matrix $T_s(Q)$ for the observed performance matrix Q, as well as for each permutation Q' of the performance matrix: $[T_s(Q'_1), T_s(Q'_2), \ldots]$. We compare the distribution of values from $T_s(Q)$ with the distribution of values from the permuted tables $[T_s(Q'_1), T_s(Q'_2), \ldots]$. We conduct a permutation test to compare the average, 75th, and 95th percentiles of these distributions. The p-value of the permutation test is calculated as the proportion of permuted tables for which the statistic is greater than the one obtained with the observed table. Additionally, we use the Kolmogorov-Smirnov (KS) test to compare the entire distribution of values between observed and permuted similarity matrices.

To further support our findings, we cluster the observed and permuted performance vectors. If there are non-random correlations between performance vectors, we would expect the clustering of the observed vectors to have higher clustering quality metrics, such as silhouette score.

3.3 Effect of non-uniform weights in aggregate performance metrics

So far, we have focused on aggregate performance measures that treat prompts as if they are independent and identically distributed (i.i.d.) samples from some real-world distribution of interest—i.e., each prompt is given equal weight in calculating aggregate performance metrics. In this section, we examine the implications of relaxing this assumption for ranking models based on their performance. Generally, there is no universally correct distribution of interest—it depends on each user's application. Here, we look into three different ways of capturing distributional assumptions (i.e., of defining weights) for a given benchmark.

Cluster-based: We leverage the clustering of performance vectors described above. We consider the following variants for evaluating performance:

1. Only include prompts that are cluster representatives (i.e., the medoids of the clusters). This effectively decreases the size of the benchmark.

2. Include all prompts, but weigh them based on their distance from their cluster representative. We employ two types of weights:

(i) *Distance-based:* The further away a prompt is from the cluster representative, the larger its weight. This setting gives more emphasis on diversity of the benchmark. More formally, let p_i be a prompt in cluster C_j, p_j^r be the representative prompt of cluster C_j, and $d(\cdot, \cdot)$ the distance function between two prompts. The weight w for p_i is:

$$w(p_i) = \frac{d(p_i, p_j^r)}{\sum_{p_k \in C_j} \left(d(p_k, p_j^r) \right)} \frac{|C_j|}{\sum_i |C_i|}$$

The first factor is the within-cluster weight of the prompt (normalized within cluster). The second

[3] https://openai.com/blog/new-and-improved-embedding-model

factor weighs all prompts of a given cluster proportionally to the cluster's size.
(ii) *Inverse-distance weights:* The closer a prompt is to the cluster representative, the larger its weight. This setting effectively smooths out the hard clustering we produced: all data points contribute to the performance, not just the cluster representatives. The weight w for p_i is computed as:

$$w(p_i) = \frac{d^{-1}(p_i, p_j^r)}{\sum_{p_k \in C_j} \left(d^{-1}(p_k, p_j^r) \right)} \frac{|C_j|}{\sum_i |C_i|}$$

Increasing benchmark size We start with a random prompt and iteratively add new prompts into the benchmark. To select the next prompt to add, we use two methods: (i) *most informative:* select the prompt with the largest cosine distance (lowest cosine similarity) from the previously selected ones in order to obtain an informative test set with a reduced semantic similarity between prompts, (ii) *random:* select a random prompt.

Random distributions of weights We weigh each prompt and compute weighted performance, with weights drawn uniformly at random. To achieve that, we sample uniformly at random from the unit simplex using the sampling technique described in Smith and Tromble (2004). This approach aims to provide a characterization over all possible weight configurations.

3.4 Comparing performance vectors with semantic embeddings of prompts

Having established that model performance is similar across prompts, we next investigate where this similarity stems from. Our hypothesis is that for a pair of prompts, similar model performance can occur if the prompts are semantically similar.

We use linear regression to determine if there exists a significant relationship between semantic similarity and model performance similarity:

$$s_{perf}(p_i, p_j) = s_{sem}(p_i, p_j)\beta + \epsilon$$

where β is the coefficient of how much semantic similarity contributes to the model and ϵ is error.

Using all prompt pairs raises concerns about the data being i.i.d., given that each observation is a pairwise comparison and each member of a pair appears in many observations. To avoid that, we estimate one model for each prompt, including all the pairwise observations of which that prompt is a part. We collect p-values for the coefficients across all models and perform multiple hypotheses adjustment to generate False Discovery Rate (FDR) values. We repeat the same approach for 1000 permutations as described in Section 3.2 for both pairwise performance and semantic similarity vectors. Finally, we compare the distribution of coefficients and FDRs between original data and permutations using the KS test.

4 Experimental setup

In this section, we describe the setting of our experiments. Specifically, we provide details on the benchmarks and evaluation metrics we use, the LLMs we consider, and how we evaluate performance of the LLMs on the benchmarks.

4.1 Benchmarks

We investigate four major benchmarks that are designed for different tasks.

ANLI The Adversarial Natural Language Inference (ANLI) dataset[4] is a large-scale dataset for natural language inference (NLI) (Nie et al., 2020). It is collected via an iterative, adversarial human-and-model-in-the-loop procedure, making it more difficult than its predecessors. The dataset used here comprises approximately 100K samples for the training set, 1,200 for the development set, and 1,200 for the test set. Each sample contains a context, a hypothesis, and a label. The goal is to determine the logical relationship between the context and the hypothesis. The label is the assigned category indicating that relationship. In the context of NLI, the labels typically include "entailment", "contradiction", or "neutral". Finally, ANLI makes available a reason (provided by the human-in-the-loop), explaining why a sample was misclassified.

HellaSwag This is a commonsense natural language inference dataset (Zellers et al., 2019), tasking machines with identifying the most probable followup for an event description. Comprising 70,000 instances, each scenario presents four potential outcomes, with only one being accurate. Engineered to be challenging for cutting-edge models, the dataset employs Adversarial Filtering to incorporate machine-generated incorrect responses, frequently misclassified by pretrained models. Covering diverse domains, HellaSwag demands a fusion of world knowledge and logical reasoning for successful interpretation.

[4] https://huggingface.co/datasets/anli

CommonsenseQA This is a multiple-choice question-answering dataset that requires different types of commonsense knowledge to predict the correct answers (Talmor et al., 2019). It contains 12,102 questions with one correct answer and four distractor answers. The questions are crowdsourced and cover a wide range of topics such as open-domain-qa, real-life situations, elementary science, social skills.

CNN/Daily Mail The CNN/Daily Mail dataset is a widely used benchmark for text summarization (Nallapati et al., 2016). The dataset comprises news stories from CNN and Daily Mail websites. In total, the corpus contains 286,817 training, 13,368 validation, and 11,487 test pairs.

4.2 Evaluation measures

For ANLI, HellaSwag, and CommonsenseQA, the performance matrix contains binary values (correct / incorrect answer). Hence, we use average accuracy to evaluate the performance of each model, as commonly done with these benchmarks (Nie et al., 2020; Wei et al., 2022; Zellers et al., 2019; Talmor et al., 2019). For CNN/Daily Mail, following previous work (See et al., 2017), we measure model performance using the ROUGE score.

4.3 Considered LLMs

In order to have a diverse collection of LLMs, we include models from several developers, such as OpenAI and Meta. Table 1 shows the various models used for each benchmark[5].

4.4 Performance evaluation

For ANLI, we evaluate each model on the test dataset, which contains 1200 prompts. For each sample, we use 7 few-shot samples extracted from the ANLI dev set. For the remaining benchmarks, we randomly sample 10% of each benchmark for test and use the rest for few-shot selection. This results in 1005, 1221, and 1150 test samples for HellaSwag, CommonsenseQA, and CNN/Daily Mail respectively. For HellaSwag, we use 10 few-shot examples, while for CommonsenseQA and CNN/Daily Mail we use 5 few-shots.

[5]Due to constraints in LLMs' availability, we use different LLMs for each benchmark. This does not impact on our work, as each benchmark analysis is independent.

5 Results

In this section, we present the results of the experiments described in Section 3 on the benchmarks.

Table 1: LLMs used for ANLI, HellaSwag (HS), CommonsenseQA (CSQA), and CNN/Daily Mail (CNN/DM). These include GPT LLMs (Brown et al., 2020; OpenAI, 2023), Llama LLMs (Touvron et al., 2023), and other popular LLMs, such as Falcon-180b (Almazrouei et al., 2023), Koala 13B (Geng et al., 2023), Alpaca 7B (Wang et al., 2023b).

Type	Model	ANLI	HS	CSQA	CNN/DM
GPT	ChatGPT-Turbo-Base-0516	✓	✓		
	ChatGPT-Turbo-0301	✓	✓		
	ChatGPT-Turbo-0613			✓	
	ChatGPT-202301				✓
	DaVinci (GPT-3)			✓	
	Text-Davinci-002				✓
	Text-Davinci-003				✓
	GPT-4-0314			✓	
	GPT-4-0314 (Chat)	✓	✓		✓
	GPT-4-0613 (Chat)			✓	
	GPT-4-Turbo-1106 (Chat)	✓	✓	✓	
	GPT-4-Turbo-1106			✓	
	Text-Alpha-002-Current		✓		✓
	DV3-FP8				✓
	Babbage-0721				✓
	ChatGPT-202301				✓
LLAMA	Llama-13B			✓	
	Llama-2-13B	✓	✓		
	Llama-30B		✓	✓	
	Llama-65B			✓	
	Llama-2-70B	✓	✓	✓	
Other	Persimmon 8B[1]	✓	✓	✓	
	Vicuna 13B[2]	✓	✓		
	Claude-2[3]	✓	✓	✓	
	Falcon-180b	✓		✓	
	Koala 13B	✓	✓		
	Mistral7b[4]	✓		✓	
	Alpaca 7B		✓		
	Total	12	13	14	8

[1] https://www.adept.ai/blog/persimmon-8b
[2] https://lmsys.org/blog/2023-03-30-vicuna/
[3] https://www.anthropic.com/index/claude-2
[4] https://mistral.ai/news/announcing-mistral-7b/

5.1 Performance vectors are correlated

To determine if prompt performance vectors are correlated, we perform the permutation tests described in Section 3.2, using different correlation measures. The obtained p-values for ANLI, HellaSwag, and CommonsenseQA are depicted in Table 2. On ANLI and CommonsenseQA, the permutation tests show strong evidence that the correlations between the prompt performance vectors are significant. For HellaSwag, our findings reveal consistently low p-values across all correlation measures when using the 75th percentile, as well as a low p-value when averaging Jaccard similarities. For the three benchmarks above, the KS test

Table 2: p-values obtained with permutation tests and the KS test using different correlation measures and aggregation functions for ANLI, HellaSwag (HS), and CommonsenseQA (CSQA).

		Hamming	Cosine	Jaccard
ANLI	Average	0.60	0.59	**0.0009**
	75th percentile	0.66	**0.0009**	0.67
	95th percentile	**0.0009**	**0.0009**	**0.0009**
	KS test	**2e-5**	**2e-5**	**2e-5**
HS	Average	0.52	0.57	**0.0009**
	75th percentile	**0.0009**	**0.0009**	**0.0009**
	95th percentile	0.88	0.85	0.87
	KS test	**2e-5**	**2e-5**	**2e-5**
CSQA	Average	0.53	0.52	**0.0009**
	75th percentile	**0.0009**	**0.0009**	**0.0029**
	95th percentile	**0.0009**	**0.0009**	**0.0009**
	KS test	**2e-5**	**2e-5**	**2e-5**

Table 3: Average silhouette score of clustering observed performance vectors and a random permutation of performance vectors for the various benchmarks.

Benchmark	observed	permuted
ANLI	**0.52**	0.21
HellaSwag	**0.54**	0.24
CommonsenseQA	**0.61**	0.29
CNN/Daily Mail	**0.25**	0.21

is significant across all correlation measures.

For CNN/Daily Mail the performance matrix contains ROUGE scores, which are continuous values. Thus, we use cosine similarity to compare the average correlations obtained from the original and permuted performance matrices. The results show that the correlations among original performance vectors are significantly greater.

To further support this finding, we cluster the model responses using spherical k-means (Dhillon and Modha, 2001). We choose the optimal number of clusters to maximize the average silhouette score, computed using cosine distance. Table 3 contains the average silhouette scores of clustering the performance vectors and a random permutation of them. For all benchmarks, the performance vectors produce higher silhouette scores compared to the permuted performance vectors. This provides additional evidence to support the outcome of the hypothesis tests presented above: the performance vectors are similar.

5.2 Impact of prompt weights on performance and relative ranking of models

In this section, we present the results of different weighting schemes for the prompts of a benchmark, as described in Section 3.3.

5.2.1 Cluster-based evaluation

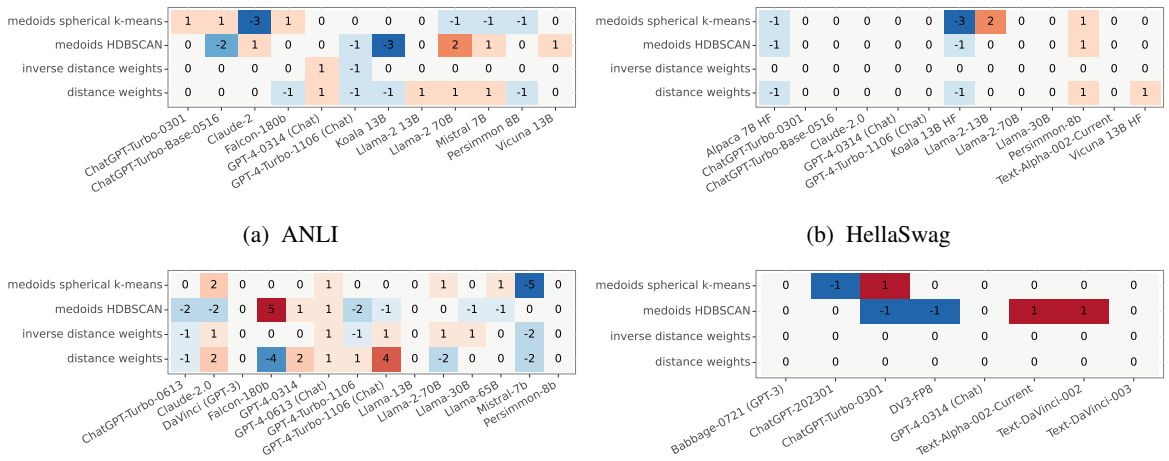
First, we cluster the performance vectors of each benchmark as described earlier. Then, we compute the average accuracy of models for each benchmark, using only the cluster representatives of that benchmark. We also compute weighted performance using distance-based and inverse-distance-based weights. Figure 1 illustrates how these weighting schemes affect the relative ranking of models for each benchmark. The rows correspond to different weighting schemes, while the columns correspond to the different models and are ordered by increasing original performance (i.e., decreasing rank). Every cell contains the ranking change (compared to the original benchmark) of the model of that column for the method of that row. If there were no ranking changes, all values would be 0. However, we observe that there are multiple ranking changes as great as 5 (model is ranked 5 positions above the original benchmark).

5.2.2 Increasing size of benchmark

Next, we study how performance is affected by the size and diversity of the benchmark. We start with a random prompt and iteratively add new prompts to the benchmark, either by adding the most informative prompt (i.e., the one with the maximum average distance from the current benchmark), or a random one. Figure 2 shows the average performance for each model as the benchmark size increases (maximum benchmark size corresponds to the original benchmark). Looking at the most informative method for ANLI (Figure 2a), the first 400 prompts result in random performance (0.5) for all models. This suggests that the initial prompts chosen with this method are the most "difficult", in that the models are exhibiting performance close to random (accuracy 50%). Similar results are observed for HellaSwag and CommonsenseQA (see Appendix C, Figure 9), but not for CNN/Daily Mail (Figure 2b), where the performance on the reduced benchmark follows a similar pattern as the performance on the original benchmark. The random method tracks the original performance for all benchmarks (see Appendix C, Figure 10).

5.2.3 Random distributions of weights

We explore the distribution of all weighting schemes and the effect they have on the weighted accuracy and relative ranking of the models. As described in Section 3.3, we sample 100,000 random weight configurations. For each model, we



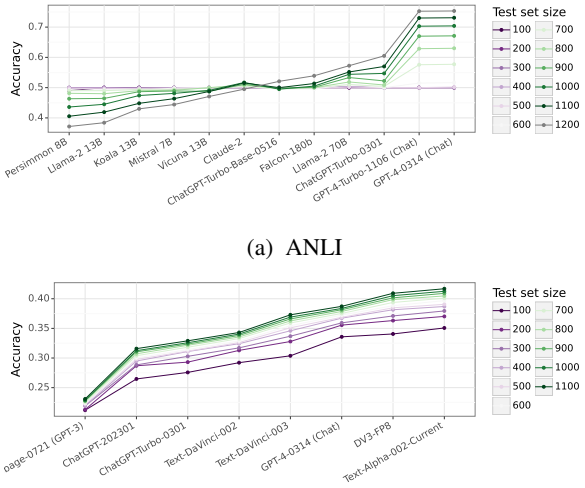
(a) ANLI

(b) HellaSwag

(c) CommonsenseQA

(d) CNN/Daily Mail

Figure 1: Visualization of ranking changes (compared to original benchmark) for various benchmark modifications. Rows show different weighting methods, columns show the models. Each cell contains the ranking change (original ranking minus new ranking) of the column-model for the row-method. We observe rank changes as great as 5.



(a) ANLI

(b) CNN/Daily Mail

Figure 2: Average performance as benchmark size increases. Prompts are added to maximize average cosine distance. Maximum benchmark size corresponds to performance on the original benchmark.

compute the weighted performance based on these weights.

For ANLI, HellaSwag, and CommonsenseQA the performance of a model can change up to 10%. For CNN/Daily Mail, the range is smaller, up to 3%. Detailed results are included in Appendix D. We note that the range is similar for all models within a benchmark, indicating that it is a property related to the benchmark and not the specific models.

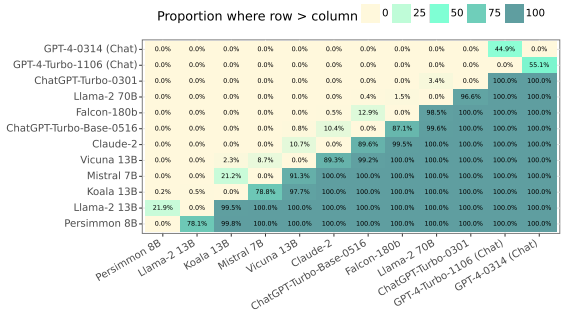
To further demonstrate changes in relative ranking of models, we take a closer look at the pairwise ranking differences. Figure 3 depicts a pairwise comparison of weighted performance for each benchmark. Every cell shows how often the model in the row outperforms the model of the column. For ANLI, approximately for half of the weight configurations the ranking of the top two models is reversed! However, for the CNN/Daily Mail data, there are effectively no reversals (less than 0.01%).

5.3 Relationship between model performance and semantic similarity of prompts

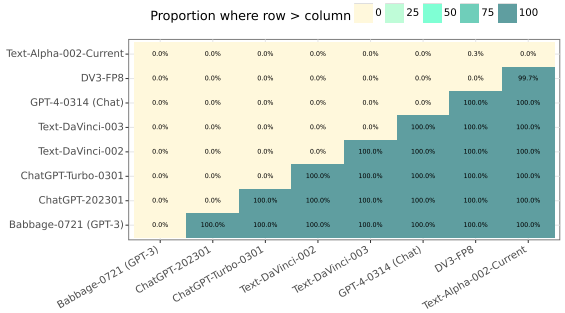
Having established that model performance is correlated across prompts, we investigate what can explain these correlations. Our hypothesis is that it is driven by semantic similarity. We use the method described in Section 3.4 to assess if there is a significant relationship between semantic similarity and model performance similarity.

Our findings show that only CNN/Daily Mail presents a significant relationship between prompt semantic similarity and prompt performance similarity (see Figure 4d). This benchmark is a text summarization task, where the success of the ROUGE metric highly depends on the ability to extract relevant entities from text. For example, we find that prompts referring to the economy or global warming have high correlation in model performance (see Appendix B, Table 5).

ANLI also makes available a reason component: what human agents state as the explanation for why the LLM gave a wrong answer. We find a significant relationship between semantic similarity

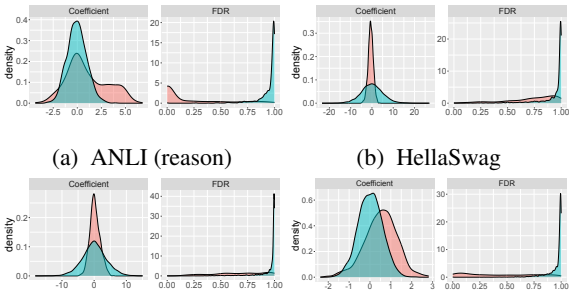


(a) ANLI



(b) CNN/Daily Mail

Figure 3: Pairwise comparison of weighted performance. Each cell is the percentage of times the model of the row outperforms the model of the column.



(a) ANLI (reason) (b) HellaSwag

(c) CommonsenseQA (d) CNN/Daily Mail

Figure 4: Distribution of semantic similarity coefficients and FDRs for all benchmarks. Red is original data, blue is permutations. KS tests for all distributions shown have p-values < 2e-5.

using the reason component and prompt performance similarity (as seen in Figure 4a). The input prompt—consisting of the context, hypothesis and label components—shows no relationship, which is most likely because the creators of ANLI put great effort into ensuring diversity in the benchmark (Nie et al., 2020). This is also evident in Figure 2. The significance of the reason component indicates that the model performance vectors correlate because of *how* the model generates a response. We observe prompts where the reasons for similar model performance indicate that the model cannot do math, e.g., "The system may have missed this as it did not add up the losses from both sets" and "the model might not know math" (see Appendix B, Table 4).

Hellaswag and CommonsenseQA use a multiple-choice format. The lack of strong evidence supporting the correlation in these benchmarks (see Figures 4b and 4c) is likely due to the embeddings picking up similarities between the different choices, rather than the logic the LLMs employ to arrive at their conclusion. This is consistent with our findings for ANLI, where a significant relationship does not stem from inputs to the model, but from the LLMs' failure points.

Our findings indicate there is a larger question about why the model performance vectors are correlated, and investigating this is central to understanding model performance. Semantic similarity can be a factor, but it depends on the task the benchmark is designed for. Based on our results for ANLI, it appears that the reasoning required for the task (i.e., reasoning types that cause models to fail), can be even more important than semantic similarity.

6 Conclusion and future work

LLMs are commonly evaluated on benchmarks that may include multiple prompts testing similar skills. In this work, we demonstrate this bias on major benchmarks, by showing that model performance across different prompts is significantly correlated. Furthermore, we demonstrate that LLM comparative studies can be significantly altered when using non-uniform weights for prompts during evaluation. The suggested approach can serve as a consistency check in comparative studies of LLMs, ensuring that the results take into consideration benchmark biases. Finally, we show that similar model performance across prompts can be explained by semantic similarity, but is most likely derived from common failure points of the LLM.

Our findings could influence a larger diagnostics tool for evaluating the robustness of model quality comparisons with respect to distributional assumptions of benchmarks. Future work also includes identifying additional factors that may explain these biases. This information can give rise to solutions for improving benchmarks robustness. These findings could help researchers generating novel benchmarks to identify and eliminate biases.

7 Limitations

Our study requires access to multiple LLMs to generate model performance vectors for each prompt in a benchmark. This can be computationally expensive and require GPUs. Some models, such as OpenAI's GPT-4, have limited API calls, making data collection time consuming.

While we provide a novel approach for researchers to investigate bias in their own studies, providing a comprehensive de-biasing methodology is not within the scope of this work.

Finally, we have only touched the surface on why prompts have similar performance across multiple LLMs. There are many other components to investigate, such as the length of the prompt and prompt complexity. This information could be leveraged to propose solutions on improving benchmarks, without running prompts through multiple LLMs.

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Maitha Alhammadi, Mazzotta Daniele, Daniel Heslow, Julien Launay, Quentin Malartic, et al. 2023. The falcon series of language models: Towards open frontier models. *Hugging Face repository*.

Norah Alzahrani, Hisham Abdullah Alyahya, Yazeed Alnumay, Sultan Alrashed, Shaykhah Alsubaie, Yusef Almushaykeh, Faisal Mirza, Nouf Alotaibi, Nora Altwairesh, Areeb Alowisheq, et al. 2024. When benchmarks are targets: Revealing the sensitivity of large language model leaderboards. *arXiv preprint arXiv:2402.01781*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Kaijie Zhu, Hao Chen, Linyi Yang, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2023. A survey on evaluation of large language models. *arXiv preprint arXiv:2307.03109*.

Matthew DeBell. 2018. *Best Practices for Creating Survey Weights*, pages 159–162. Springer International Publishing, Cham.

Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric Xing, and Zhiting Hu. 2022. RLPrompt: Optimizing discrete text prompts with reinforcement learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3369–3391, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Inderjit S Dhillon and Dharmendra S Modha. 2001. Concept decompositions for large sparse text data using clustering. *Machine learning*, 42:143–175.

Xinyang Geng, Arnav Gudibande, Hao Liu, Eric Wallace, Pieter Abbeel, Sergey Levine, and Dawn Song. 2023. Koala: A dialogue model for academic research. *Blog post, April*, 1.

Karan Goel, Nazneen Fatema Rajani, Jesse Vig, Zachary Taschdjian, Mohit Bansal, and Christopher Ré. 2021. Robustness gym: Unifying the NLP evaluation landscape. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations*, pages 42–55, Online. Association for Computational Linguistics.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913.

Zishan Guo, Renren Jin, Chuang Liu, Yufei Huang, Dan Shi, Linhao Yu, Yan Liu, Jiaxuan Li, Bojian Xiong, Deyi Xiong, et al. 2023. Evaluating large language models: A comprehensive survey. *arXiv preprint arXiv:2310.19736*.

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.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*.

Ronan Le Bras, Swabha Swayamdipta, Chandra Bhagavatula, Rowan Zellers, Matthew Peters, Ashish Sabharwal, and Yejin Choi. 2020. Adversarial filters of dataset biases. In *International conference on machine learning*, pages 1078–1088. PMLR.

Xinzhe Li, Ming Liu, Shang Gao, and Wray Buntine. 2023. A survey on out-of-distribution evaluation of neural nlp models. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 6683–6691. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian

Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gùlçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

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*. Association for Computational Linguistics.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4658–4664, Florence, Italy. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report.

Yonatan Oren, Nicole Meister, Niladri Chatterji, Faisal Ladhak, and Tatsunori B Hashimoto. 2023. Proving test set contamination in black box language models. *arXiv preprint arXiv:2310.17623*.

Alejandro Peña, Aythami Morales, Julian Fierrez, Ignacio Serna, Javier Ortega-Garcia, Iñigo Puente, Jorge Cordova, and Gonzalo Cordova. 2023. Leveraging large language models for topic classification in the domain of public affairs. *arXiv preprint arXiv:2306.02864*.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Hamza Alobeidli, Alessandro Cappelli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon LLM: Outperforming curated corpora with web data only. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Reid Pryzant, Dan Iter, Jerry Li, Yin Lee, Chenguang Zhu, and Michael Zeng. 2023. Automatic prompt optimization with "gradient descent" and beam search. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7957–7968, Singapore. Association for Computational Linguistics.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. An adversarial winograd schema challenge at scale. *arXiv preprint arXiv:1907.10641*.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Noah A Smith and Roy W Tromble. 2004. Sampling uniformly from the unit simplex. *Johns Hopkins University, Tech. Rep*, 29.

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.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Jindong Wang, Xixu Hu, Wenxin Hou, Hao Chen, Runkai Zheng, Yidong Wang, Linyi Yang, Haojun Huang, Wei Ye, Xiubo Geng, et al. 2023a. On the robustness of chatgpt: An adversarial and out-of-distribution perspective. *arXiv preprint arXiv:2302.12095*.

Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, and Tao Qin. 2021. Generalizing to unseen domains: A survey on domain generalization. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4627–4635. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Xuezhi Wang, Haohan Wang, and Diyi Yang. 2022. Measure and improve robustness in NLP models: A survey. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4569–4586, Seattle, United States. Association for Computational Linguistics.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Chandu, David Wadden, Kelsey MacMillan, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. 2023b. How far can camels

go? exploring the state of instruction tuning on open resources. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Kai-Cheng Yang and Filippo Menczer. 2023. Large language models can rate news outlet credibility. *arXiv preprint arXiv:2304.00228*.

Linyi Yang, Shuibai Zhang, Libo Qin, Yafu Li, Yidong Wang, Hanmeng Liu, Jindong Wang, Xing Xie, and Yue Zhang. 2023. GLUE-X: Evaluating natural language understanding models from an out-of-distribution generalization perspective. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12731–12750, Toronto, Canada. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma, Yufei Xue, Jidong Zhai, Wenguang Chen, Zhiyuan Liu, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023. GLM-130b: An open bilingual pre-trained model. In *The Eleventh International Conference on Learning Representations*.

Kun Zhou, Yutao Zhu, Zhipeng Chen, Wentong Chen, Wayne Xin Zhao, Xu Chen, Yankai Lin, Ji-Rong Wen, and Jiawei Han. 2023. Don't make your llm an evaluation benchmark cheater. *arXiv preprint arXiv:2311.01964*.

Kaijie Zhu, Jindong Wang, Jiaheng Zhou, Zichen Wang, Hao Chen, Yidong Wang, Linyi Yang, Wei Ye, Neil Zhenqiang Gong, Yue Zhang, et al. 2023. Promptbench: Towards evaluating the robustness of large language models on adversarial prompts. *arXiv preprint arXiv:2306.04528*.

A Prompt structure

The prompts used for inference are depicted in Figures 5, 6, 7 and 8 for ANLI, HellaSwag, CommonsenseQA and CNN/Daily respectively.

Given the following context: {premise}
Question:{hypothesis}
True, False or Neither?
The answer is:

Figure 5: Prompt used during inference for ANLI.

System: You are an AI assistant. Provide a detailed answer so user do not need to search outside to understand the answer.
User: Category: {activity_label}
Text: {ctx}
Completion options:
(1) {choice_1}
(2) {choice_2}
(3) {choice_3}
(4) {choice_4}
Assistant: The most likely text completion is:

Figure 6: Prompt used during inference for HellaSwag.

Question: {{question}}
Answer options:
(A) {{choiceA}}
(B) {{choiceB}}
(C) {{choiceC}}
(D) {{choiceD}}
(E) {{choiceE}}
The answer is:

Figure 7: Prompt used during inference for CommonsenseQA.

Article:
{Text to summarize}
Summary:

Figure 8: Prompt used during inference for CNN/Daily Mail.

B Results: Semantically similar prompts

For the statistical tests in Section 3.4, we describe a set of linear regression models being generated where each model contains the prompt pairs of a specific single prompt. Here, we display semantically similar prompts from these models where the semantic similarity coefficient is high and significant in explaining the model performance dependent variable.

In Table 4, the ANLI reason component demonstrates that the prompts are adversarial because the model is unable to perform simple math operations. In other words, the prompts elicit the same mathematical operation task. For CNN/Daily Mail data, the prompts either refer to the economy or global warming as seen in Table 5. This entails that the models' performance had similar capabilities in extracting text about these subjects.

C Results: Increasing size of benchmark

Figure 9 shows results for all benchmarks for our experiments on increasing size of benchmark using the most informative method, as described in Section 5.2.2. Figure 10 shows results for all benchmarks when adding prompts in random order.

D Results: Distributions of weighted performance

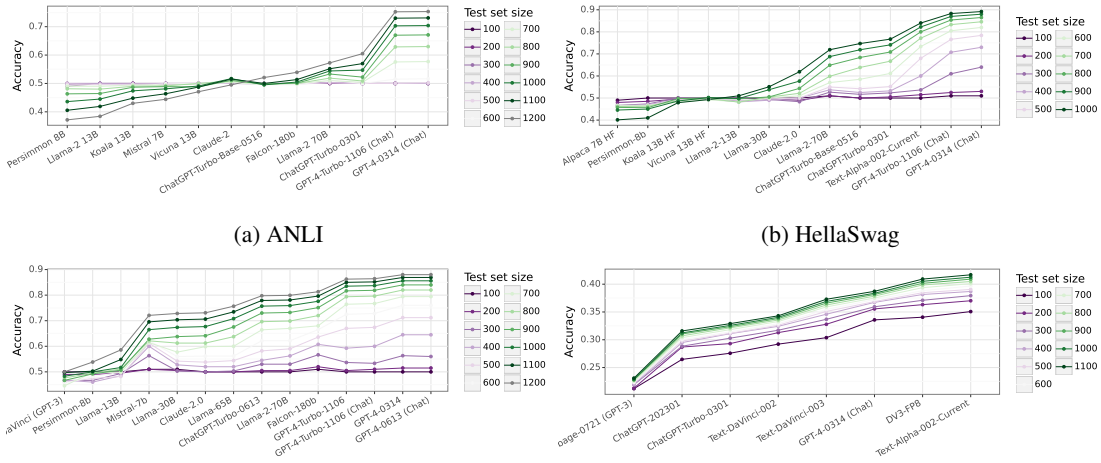
Figure 11 shows distribution of weighted performance and pairwise ranking changes for all benchmarks.

Table 4: List of ANLI reasons having high semantic similarity with model performance.

Reason	Text
1	it says osaka beat williams 6-2, 6-4. So osaka lost 6 games total. The system may have missed this as it did not add up the losses from both sets
2	The 1972–73 California Golden Seals had a 13–55–10 record - so they lost about 4 times as many [55] as they won [13]; the model might not know math.
3	Although Shigeko Sasamori was interviewed about this event, it's uncertain if she witnessed it personally. I think the system is confused because of so many matching words.
4	It does not state whether she was rebound leader - although her points total was tied with another player - which might have confused the model.
5	his record is 6-5 not 5-5

Table 5: List of Daily/CNN grounded truth summaries having high semantic similarity with model performance.

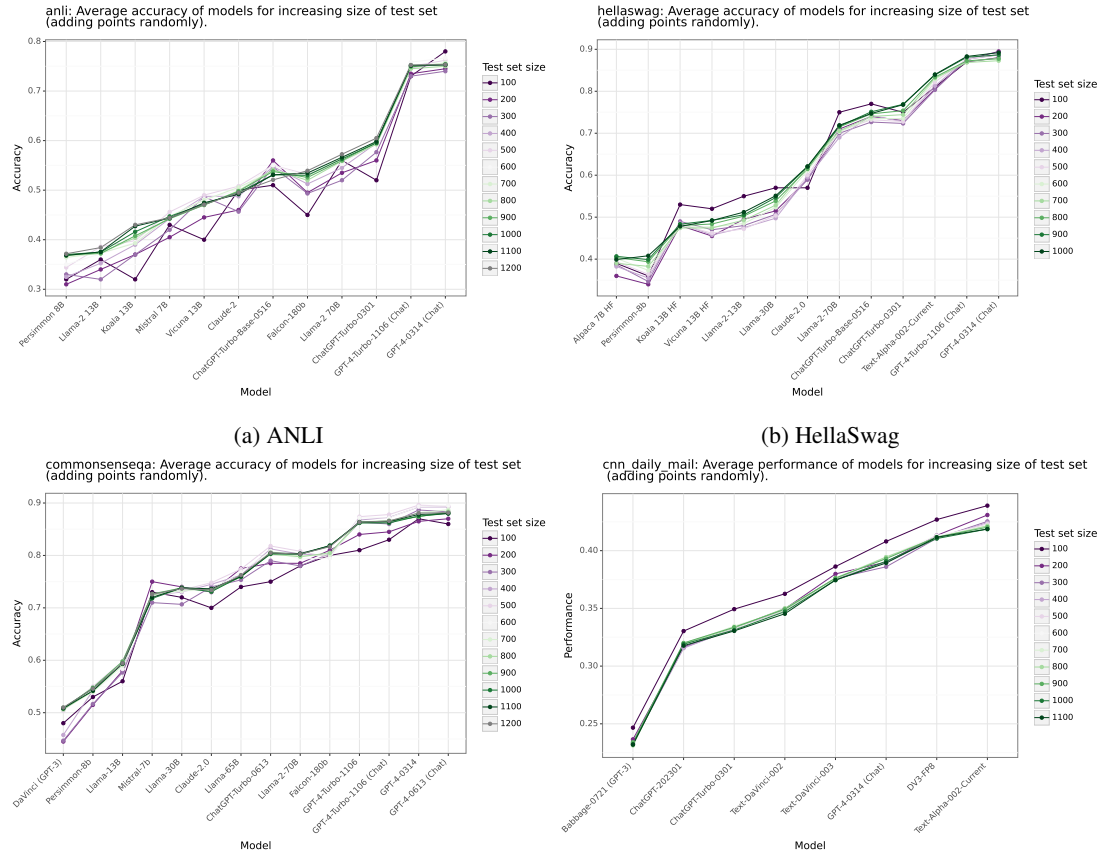
Label	Text
1	Jeffrey Sachs : Raw capitalism is the economics of greed . Last year was the Earth's hottest year on record, he says.
2	Adam Sobel : California's steps against drought are a preview for rest of U.S. and world. Tying climate change to weather doesn't rest on single extreme event, Sobel says. The big picture should spur us to prepare for new climates by fixing infrastructure, he says.
3	India predicted to outpace China as as world's fastest-growing economy in next year. China's economy is slowing after over 25 years of breakneck growth. But experts say India simply can't size up against China 's raw economic might.
4	Bill Richardson : U.S announced plan to cut greenhouse gas emissions by 26 to 28 percent below 2005 levels by 2025. He says China, India, major corporations, cities among those already setting goals for cutting emissions. U.S. must lead in this effort.



(a) ANLI (b) HellaSwag

(c) CommonsenseQA (d) CNN/Daily Mail

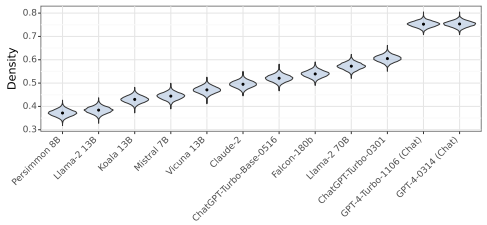
Figure 9: Average performance as benchmark size increases. Prompts are added to maximize average cosine distance. Maximum benchmark size corresponds to performance on the original benchmark.



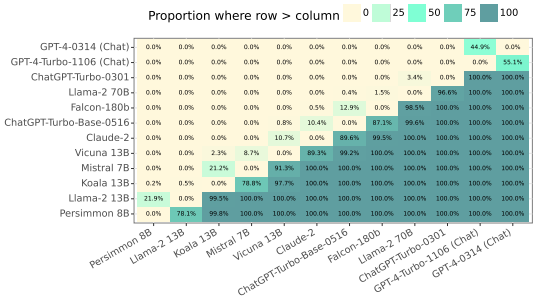
(a) ANLI (b) HellaSwag

(c) CommonsenseQA (d) CNN/Daily Mail

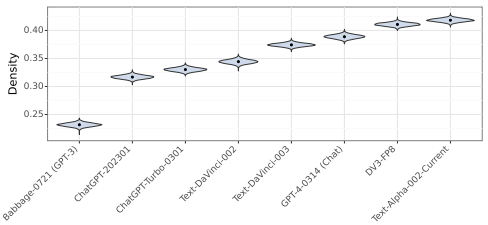
Figure 10: Average performance as benchmark size increases. Prompts are added in random order. Maximum benchmark size corresponds to performance on the original benchmark.



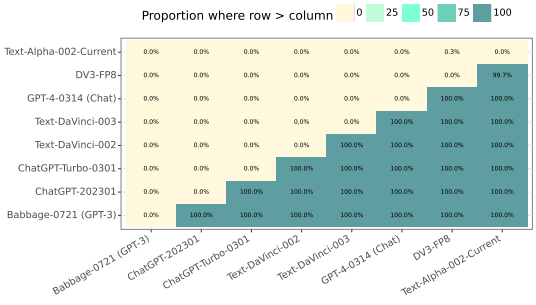
(a) ANLI



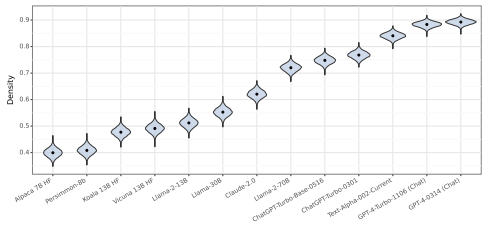
(b) ANLI



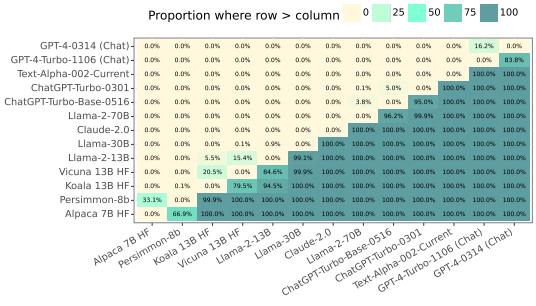
(c) CNN/Daily Mail



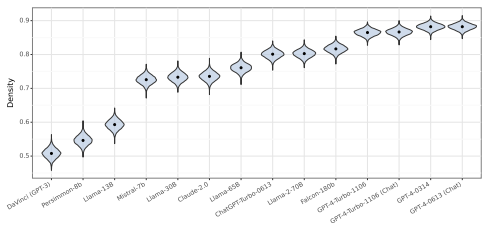
(d) CNN/Daily Mail



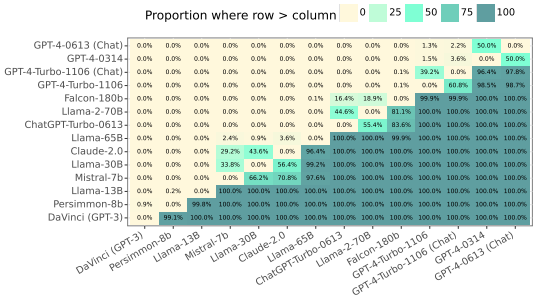
(e) HellaSwag



(f) HellaSwag



(g) CommonsenseQA



(h) CommonsenseQA

Figure 11: Left column: Distribution of weighted performance for randomly sampled weights. The black dot corresponds to performance when using uniform weights. Right column: Pairwise comparison of weighted performance. Every cell corresponds to the proportion of times the model in the row outperforms the model of the corresponding column.