

LMFlow Benchmark: A Scalable Evaluation Paradigm for Large Language Models

Anonymous

Abstract

The AI community has witnessed the blowout of chat-style Large Language Models (LLMs) since the breakthrough of ChatGPT. Despite the rapid development of LLMs, evaluation of these models still remains a huge challenge. So far only human evaluation or expensive GPT-4 oracles have been widely recognized, but they are neither automatic nor scalable. To fill this gap, we propose LMFlow Benchmark, a powerful evaluation paradigm capable of measuring LLMs' different aspects automatically and efficiently. Based on this novel paradigm, we present a comprehensive comparison among a large number of existing LLMs, which quantifies their strength and weakness, providing a convenient guide for proper model selection. LMFlow Benchmark is also adopted for model performance monitoring, offering useful insights for addressing LLM training issues, such as the infamous model forgetting problem.

1 Introduction

With the groundbreaking debut of ChatGPT (OpenAI, 2022), a main branch of AI research has been driven towards the development of better Large Language Models (LLMs), attempting to uncover the secrets behind the success of ChatGPT. As part of this endeavor, various LLMs (Scao et al., 2022; Taylor et al., 2022; Zhang et al., 2022; Rae et al., 2021; Hoffmann et al., 2022) have been proposed, claiming superior performance compared to their competitors. Vicuna-13B (Chiang et al., 2023), for instance, is reported to be 90% as effective as ChatGPT and Google Bard (Anil et al., 2023). Meanwhile, Falcon-40B (Almazrouei et al., 2023) comes out, stating that its performance surpasses the 65B version of the well-known LLaMA model (Touvron et al., 2023). There are also numerous other models with impressive and competing performance (Team, 2023; Anil et al., 2023; Anthropic, 2022), continue to emerge in this era of LLM.

As the number of new LLMs keeps growing, proper evaluation methods for comparing those models become more and more demanding. On the one hand, entrepreneurs and practitioners need a reliable way to compare LLMs and decide which model to use for various application scenarios. On the other hand, researchers and engineers call for tools to monitor LLM performance during training, hence performance issues like forgetting (Sun et al., 2020; Jang et al., 2022) can be detected and avoided in advance. These all require a suitable evaluation paradigm that is accurate, automatic, scalable, multidimensional, efficient and budget-friendly.

Unfortunately, to the best of our knowledge, no current evaluation framework is capable of satisfying all those criteria simultaneously. Existing benchmarks are either dependent on expensive evaluation oracles, such as human evaluations or GPT-4 (Chiang et al., 2023; Zheng et al., 2023a) which are neither scalable nor budget-friendly, or only focus on limited aspects of LLMs (Wang et al., 2018; Lin, 2004; Papineni et al., 2002).

To bridge this gap, we propose LMFlow Benchmark, the first evaluation paradigm that fulfills all the aforementioned requirements simultaneously. By introducing a novel perspective on Negative Log Likelihood (NLL), we are able to measure different aspects of LLMs in a scalable manner, which makes it possible for researchers and engineers better understand the current trend of LLMs, along with other useful insights shedding into important training issues. Furthermore, LMFlow Benchmark is locally deployable, which allows everyone to examine crucial aspects of LLMs they cared about. We summarize our detailed contributions as follows,

- A novel LLM evaluation paradigm general enough to be automatic, scalable, multidimensional, efficient and budget-friendly all at once.

- A comprehensive list of comparisons among a large number of existing LLMs, which quantifies their strengths and weakness on three important aspects: dialog, instruction following and commonsense.
- Insights on three key training issues: forgetting, overfitting and hyperparameter tuning, which leads to practical recommendations for hyperparameter choices and training tricks.

2 Related Work

Large Language Models LLMs have demonstrated remarkable progress across a variety of Natural Language Processing (NLP) tasks. The impressive performance of ChatGPT (OpenAI, 2022) and GPT-4 (OpenAI, 2023) has motivated the research community to reproduce their success, resulting in a rapidly increasing number of foundation models in the past year. These efforts include BLOOM (Scao et al., 2022), OPT (Zhang et al., 2022), Galactica (Taylor et al., 2022), CodeGen (Nijkamp et al., 2022), StarCoder (Li et al., 2023a), LLaMA (Touvron et al., 2023), StableLM (AI, 2023), GPT-Neo (Black et al., 2021), GPT-J (Wang and Komatsuzaki, 2021), pythia (Biderman et al., 2023), Redpajama (Together, 2023), MPT (Team, 2023) and Falcon (Almazrouei et al., 2023), all of which have generated significant interest worldwide. Several LLMs are trained to achieve a general foundation model applicable for a wide range of downstream tasks, like BLOOM (Scao et al., 2022), OPT (Zhang et al., 2022) and LLaMA (Touvron et al., 2023), while other LLMs, like Galactica (Taylor et al., 2022) and CodeGen (Nijkamp et al., 2022), are introduced to address problems in specific domains, e.g. science and code.

These foundation models have resulted in the exponential growth of fine-tuned variants in task-oriented settings, such as chatbots. Early efforts in developing chat LLMs include Vicuna (Chiang et al., 2023) built on LLaMA (Touvron et al., 2023) and Dolly (Databricks, 2023) derived from pythia (Biderman et al., 2023). Several other domain-specific applications have also been developed based on these foundation models, such as Dr.LLaMA (Guo et al., 2023) and CancerGPT (Li et al., 2023b) on medical, FinGPT (Yang et al., 2023) and BloombergGPT (Wu et al., 2023) on finance, GPTutor (Chen et al., 2023) for programming, Zero NIR (Wang and Lim, 2023) for recommendation and TabLLM (Hegselmann et al., 2023) for tabular data. On top of that, LLMs have also been widely connected with visual input to enable multimodal applications, like MiniGPT4 (Zhu et al., 2023), LLaVA (Liu et al., 2023), SegGPT (Wang et al., 2023b) and DetGPT (Pi et al., 2023).

The explosive emergence of those task-oriented LLMs leads impressive performance gains and opening up new possibilities for LLM applications. Nevertheless, huge challenges have also been introduced during this process, where the inherent multidimensional nature of these LLMs requires aspect-aware evaluation methods, calling for metrics that are general enough to be employed in a wide range of tasks.

LLM Evaluation One commonly used way to evaluate LLMs is by testing them on a wide range of tasks and reporting superior performance to demonstrate their effectiveness. Like benchmarks Bigbench (Srivastava et al., 2022), MMLU (Hendrycks et al., 2020) and HELM (Liang et al., 2022), many automatic metrics (Sellam et al., 2020; Papineni et al., 2002; Lin, 2004; Wang et al., 2018) for evaluating large language models ask for careful annotation and strict formation in order to meet the requirements of downstream tasks. One commonly used metric here is accuracy over multiple options (Hendrycks et al., 2020; Clark et al., 2018; Brown et al., 2020). Foundation models are typically evaluated on a broad range of natural language tasks, using free-form generation or multiple-choice tasks to implement exact evaluation protocols (Schaeffer et al., 2023; Hendrycks et al., 2020; Wei, 2022; Srivastava et al., 2022). These evaluation protocols often focus on multiple-choice tasks, using accuracy as a primary metric. In such tasks, the objective is to select the most appropriate completion from a set of given options based on a provided context. However, task-oriented applications typically require generating open-domain unstructured outputs, and the accuracy metric based on limited 2-5 options cannot be applied to evaluating these unstructured tasks effectively.

Another widely used automatic evaluation metric is GPT-4 ratings (Chiang et al., 2023; Dettmers et al., 2023; Peng et al., 2023; Wang et al., 2023a,c), which was also termed as "LLM-as-a-judge" (Zheng et al., 2023a). It was first introduced by Vicuna (Chiang et al., 2023) to compare their model with ChatGPT (OpenAI, 2022) and LLaMA (Touvron et al., 2023). They collected 80 unseen questions for Vicuna and created specific prompts to evaluate the responses of Vicuna (Chiang et al., 2023), LLaMA (Touvron et al., 2023), and ChatGPT (OpenAI, 2022). In each evaluation round, GPT-4 (OpenAI, 2023) is required to rate the response quality between two

models, assigning scores ranging from 1 to 10. However, it should be noted that this metric requires multiple calls to GPT-4, which has a limited number of calls and requires heavy supervision to avoid abnormal cases.

Some studies (Dettmers et al., 2023; Zheng et al., 2023a; Bai et al., 2022) have employed human evaluation to assess the performance of LLMs. In these studies, human annotators are presented with pairwise responses from two models and asked to evaluate which response is better, worse, or the same as the competitor. Additionally, tournament-style competitions have been used to evaluate LLMs, where pairs of models compete against each other to generate the best possible response for a given prompt. This approach was also used in Zheng et al. (2023a); Bai et al. (2022), where Elo rating (Elo, 1967, 1978) was applied to compute the final score of each LLM. However, human annotation requires human involvement and is not applicable to large-scale performance analysis.

Compared with the evaluation metrics mentioned above, our metrics are capable of handling unstructured generation results with a scalable automatic evaluation of LLMs. Also, ours is still multidimensional, efficient and budget-friendly.

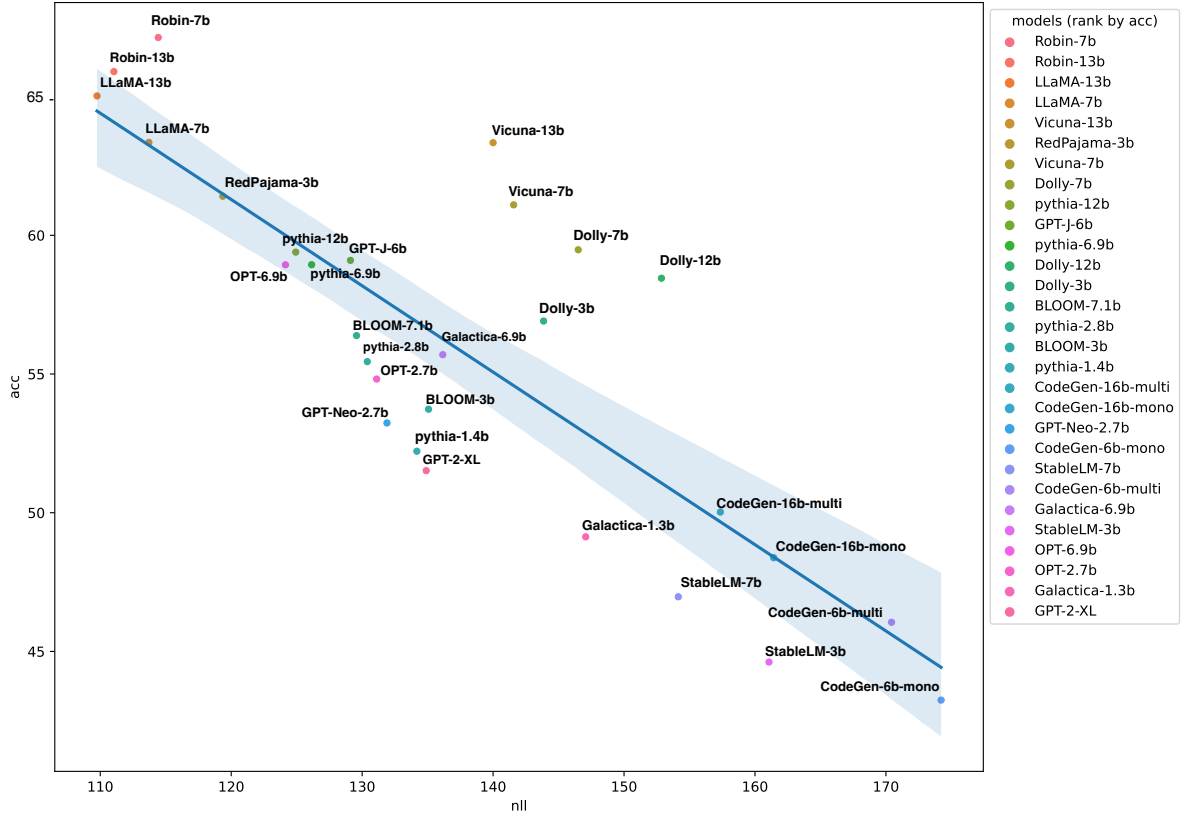


Figure 1: Correlation between NLL and accuracy on commonsense QA benchmarks. There exists an approximate correlation between commonsense QA accuracy and NLL, where NLL reflects the magnitude of performance difference between models. A significant gap in NLL usually corresponds to a gap in the accuracy of commonsense reasoning.

3 LMFlow Benchmark

Challenges in Automatic LLM Evaluation The key challenge in LLM evaluation comes from the unstructured nature of model outputs and the large diversity of possible valid responses. On the one hand, the unstructured form of model outputs makes the extraction of answers difficult for LLMs, rendering the conventional

	Negative Log Likelihood Average			Accuracy Average Commonsense	# params
	Dialog	Instruction	Commonsense		
CodeGen-6b-mono (Nijkamp et al., 2022)	275	372	174.21	43.25	6B
CodeGen-6b-multi (Nijkamp et al., 2022)	266	356	170.43	46.05	6B
StableLM-3b (AI, 2023)	255	324	161.07	44.62	3B
CodeGen-16b-mono (Nijkamp et al., 2022)	239	260	131.89	53.23	16B
StableLM-7b (AI, 2023)	239	306	154.14	46.97	7B
GPT-2-XL (Radford et al., 2019)	234	296	134.89	51.52	1.2B
CodeGen-16b-multi (Nijkamp et al., 2022)	232	302	157.36	50.02	16B
Galactica-1.3b (Taylor et al., 2022)	229	306	147.07	49.13	1.3B
BLOOM-3b (Scao et al., 2022)	215	272	135.07	53.73	3B
OPT-2.7b (Zhang et al., 2022)	213	260	131.10	54.82	2.7B
GPT-Neo-2.7b (Black et al., 2021)	208	260	131.89	53.23	2.7B
Galactica-6.9b (Taylor et al., 2022)	206	268	136.16	55.70	6.9B
BLOOM-7.1b (Scao et al., 2022)	200	252	129.57	56.38	7.1B
OPT-6.7b (Zhang et al., 2022)	197	242	124.14	58.94	6.7B
pythia-6.9b (Biderman et al., 2023)	195	238	126.14	58.94	6.9B
pythia-12b (Biderman et al., 2023)	189	235	124.93	59.39	12B
GPT-J-6b (Wang and Komatsuzaki, 2021)	187	232	129.11	59.10	6B
RedPajama-3b (Together, 2023)	178	224	119.36	61.40	3B
RedPajama-7b (Together, 2023)	174	214	119.61	61.52	7B
LLaMA-7b (Touvron et al., 2023)	173	214	113.71	63.35	7B
LLaMA-13b (Touvron et al., 2023)	167	203	109.75	65.02	13B
Dolly-12b (Databricks, 2023)	236	300	152.86	58.45	12B
Dolly-7b (Databricks, 2023)	212	266	146.50	59.48	7B
Dolly-3b (Databricks, 2023)	210	264	143.86	56.91	3B
Vicuna-7b (Chiang et al., 2023)	165	211	141.57	61.10	7B
Vicuna-13b (Chiang et al., 2023)	160	206	140.00	63.34	13B
Robin-7b	138	170	114.43	67.13	7B
Robin-13b	132	161	111.04	65.90	13B

Table 1: Evaluation results of different LLMs on Dialog, Instruction Following and Commonsense NLL/QA accuracy tasks. In order to make the table easier to understand, we only report the average results of each dimension of the foundation models. Here, "Dialog" contains both multi-turn and single-turn dialog NLL evaluations. "Commonsense" contains 7 commonly used commonsense qa datasets to implement evaluation here. "Instruction" contains GPT-4 generated feedback examples for various instructions from Peng et al. (2023). For detailed performance and data collection details for each dataset, please refer to appendix C.

classification-styled benchmarks unsuitable for this task. This rules out most evaluation methods adopted by classification tasks, such as sentiment classification (Socher et al., 2013) and text classification (Wang et al., 2018). On the other hand, the diversity of possible valid responses makes traditional text matching metrics impractical for this task. As the number of potential answers grows exponentially with longer conversation history, the probability of text matching quickly decreases to zero, and the metrics become meaningless. This prevents the application of evaluation metrics in almost all text generation tasks, including BLEU score of machine translation (Papineni et al., 2002), Exact Matching of question answering (Rajpurkar et al., 2016), ROUGE score of text summarization (Lin, 2004), etc.

Arguably the only viable choice is the perplexity metric in language modeling (Marcus et al., 1993):

$$\text{PPL}(\mathcal{M}) = \frac{1}{|\mathcal{D}|} \sum_{S \in \mathcal{D}} \exp \left(-\frac{1}{|\mathcal{S}|} \log (\text{Pr}_{\mathcal{M}}(S)) \right),$$

where \mathcal{D} is the test dataset, \mathcal{M} is the model being evaluated and S is a single sample in the dataset, normally being a paragraph of text.

Nevertheless, perplexity is normally computed over a tokenized sequence $\mathcal{T} = \text{tokenized}(S) = \langle t_1, t_2, \dots, t_{|\mathcal{T}|} \rangle$, where the lengths $|\mathcal{T}|$ are different across LLMs with distinct tokenizers. For instance, a model that treats one word as a token has objectively a lower perplexity than a model with character-level tokenizers, even they have

exactly the same predictions for all sentences. To compensate for this difference, the effect of tokenizers must be neutralized during the evaluation process, which gives rise to the likelihood idea.

A Novel View of Likelihood The core of LMFlow Benchmark is the novel application of Negative Log Likelihood (NLL), which follows a simple principle: *Generation ability is positively correlated with prediction ability*. For example, given the context “To be or not to be”, if the model predicts with high probability “that is the question”, intuitively we recognize the model as being familiar with Shakespeare’s Plays. This resembles the phenomenon where chess masters are better at memorizing and reproducing chess endgames than average (Chase and Simon, 1973; De Groot, 2014). The philosophy applies to multiple-choice question-answering as well, except for the fact that the output answers only accept limited possibilities in this case.

$$\begin{aligned}\text{NLL}(\mathcal{M}) &= -\frac{1}{|\mathcal{D}|} \sum_{S \in \mathcal{D}} \log(\text{Pr}_{\mathcal{M}}(S)) \\ &= -\frac{1}{|\mathcal{D}|} \log \left(\prod_{S \in \mathcal{D}} \text{Pr}_{\mathcal{M}}(S) \right) \\ &= -\frac{1}{|\mathcal{D}|} \log(\text{Pr}_{\mathcal{M}}(\mathcal{D})),\end{aligned}$$

For a given reference dataset, NLLs can be computed by summing NLLs over all samples, which is mathematically equivalent to the negative log likelihood of the whole corpus. By providing various types of corpora \mathcal{D} , different aspects of a model can be examined and evaluated. In LMFlow Benchmark, three major aspects are investigated, with their corresponding reference datasets offered as the standard evaluation suite in this toolkit, shown in Table 2.

The first aspect is dialog, which is one of the most important application scenarios of chatbots. LMFlow Benchmark tests models’ ability to generate proper responses given specific conversational contexts via measuring their NLL metrics on a provided chitchat dataset. This dataset comprises $\sim 1.0\text{k}$ text pairs total, with $\sim 0.5\text{k}$ multi-round and $\sim 0.5\text{k}$ single-round conversations.

The second measured aspect is instruction following, where a model is expected to loyally obey the user’s instructions across various types of tasks, including essay writing, paraphrasing, summarization, coding, math problem solving, etc. This dataset contains $\sim 1.3\text{k}$ text pairs, with the question description being the input and a reference answer as the output.

The third dimension is commonsense, reflecting the models’ learned knowledge about the real world. The reference dataset is composed of excerpts from wikipedia, where the model is expected to yield a high likelihood, thus low NLL on the text-only articles. This dataset includes $\sim 1.7\text{k}$ text samples, each being a paragraph about a common fact.

Aspect	#Samples	Format
Dialog	954	Text pairs
Instruction following	1,313	Text pairs
Commonsense	1,670	Text

Table 2: Specifics of LMFlow Benchmark Datasets.

One significant advantage of NLL is that it requires minimal human involvement during the evaluation process. Once the reference corpus is given, one can examine different aspects of an LLM’s ability automatically. This makes the evaluation of LLM more accessible to researchers.

4 Experiments

To establish the effectiveness of the proposed paradigm, we present empirical evidence demonstrating a strong correlation between NLL and widely recognized metrics. This provides direct proof that NLL is capable of

measuring model performance accurately.

Next, we apply LMFlow Benchmark to rank a list of existing LLMs, providing a valuable guideline for practitioners to select models under different application scenarios. This validation of the paradigm’s efficacy further underscores its usefulness in real-world use cases.

Finally, we introduce three insights into LLM finetuning, which were discovered by leveraging the LMFlow Benchmark paradigm. These insights serve as indirect evidence of the utility of LMFlow Benchmark and highlight its potential to provide valuable insights into the performance of LLMs.

4.1 Links to Existing Metrics

Links to Commonsense QA Commonsense question answering accuracy (Clark et al., 2019; Bisk et al., 2020; Zellers et al., 2019; Sakaguchi et al., 2021; Clark et al., 2018; Mihaylov et al., 2018) is a widely recognized metric for measuring a model’s level of knowledge. To evaluate this aspect, we adopted seven standard benchmarks for comparison with NLL: BoolQ (Clark et al., 2019), PIQA (Bisk et al., 2020), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2021), ARC easy and challenge (Clark et al., 2018) and OpenBook QA (Mihaylov et al., 2018).

Here, we visualize the correlation between NLL and accuracy for various models on the aforementioned benchmarks. As shown in Figures 1 and Table 1, commonsense QA accuracy is approximately correlated to NLL, where NLL reflects the “magnitude” of performance difference between models. A significant gap in NLL typically leads to a corresponding gap in the accuracy of commonsense reasoning.

	GPT-4	Human	NLL
Vicuna-7b	1236	1322	113
LLaMA-7b	991	929	119
Pythia-6.9b	942	924	132
BLOOM-7.1b	916	915	132
OPT-6.9b	914	909	136

Table 3: Comparison of different LLMs on long-context conversations. Here, we select 90 data examples and send the responses from each two LLMs to GPT-4, which is required to score the responses. Scores on the table equal Elo ratings for each model on pairwise competition winner determined by GPT-4/Human. Notably, the NLL metric shows an approximate recovery of the rank produced by GPT-4 and human evaluation.

Links to Human/GPT-4 Evaluation We further establish the validity of NLL in measuring conversational performance. Existing evaluation methods (Chiang et al., 2023; Bai et al., 2022; Dettmers et al., 2023) normally require comparison oracles from human or much stronger models like GPT-4 (OpenAI, 2023), where two models generate separated responses judged by the given oracle and produce a win-tie-loss table. The battle result is then translated into single-dimensional metrics such as Elo Rating (Elo, 1967, 1978), displayed in an easy-to-understand manner. Specifically, with both human and GPT-4 pairwise competition results, Elo Rating computation begins with the initial value as 1,000 and hyperparameter $K=32$. To control for order effects, such as the impact of which model pairs compete with each other first, we follow a similar approach to Chiang et al. (2023) by repeating the process 10,000 times with different random seeds. For GPT-4 evaluation, we follow the general domain rating prompt from Chiang et al. (2023), which is attached in the appendix A.

In our study, we evaluate five existing LLMs with similar model sizes: Vicuna-7b (Chiang et al., 2023), LLaMA-7b (Touvron et al., 2023), Pythia-6.9b (Biderman et al., 2023), BLOOM-7.1b (Scao et al., 2022), and OPT-6.9b (Zhang et al., 2022), based on both traditional comparison methods and NLL metrics. The results in Table 3 demonstrate a strong correlation between the two types of evaluation methods, with NLL approximately recovering the rank produced by GPT-4 and human evaluation. This suggests that NLL could serve as a suitable and cost-effective alternative to these more expensive methods.

4.2 Comparison among LLMs

The use of NLL metrics allows for efficient and accessible comprehensive evaluations of existing LLMs. Table 1 presents the performance of 28 publicly available LLMs, measured based on NLL in three dimensions: dialog, instruction following, and commonsense. It is worth noting that the Robin series, which was trained by us, achieved excellent results in all aspects. Details on the training process will be discussed in Section 4.3.3.

The results show that LLaMA (Touvron et al., 2023) consistently produces superior performance across all three dimensions, while Vicuna (Chiang et al., 2023) excels in conversational ability at the expense of commonsense reasoning. Additionally, almost all models align with the common understanding that "better performance comes with larger models," further validating the effectiveness of NLL metrics.

4.3 Insights of Training Issues

The proposed evaluation paradigm has led to the identification of important insights for LLM finetuning, resulting in effective training techniques and significant improvements in specific aspects of LLMs.

4.3.1 Forgetting

	Commonsense QA	
	NLL	Acc.
Before tuning	113.71	63.35
After tuning (20% data)	153.08	61.56
After tuning (40% data)	154.15	61.40
After tuning (100% data)	155.50	60.36

Table 4: The table illustrates LLaMA-7b’s performance on various subsets of a sampled dataset during training. Using smaller training datasets can help alleviate the issue of forgetting, highlighting the importance of using high-quality, smaller datasets. It is advisable to avoid endlessly increasing the number of data samples.

One important insight is the effect of dataset size on forgetting. According to Table 1, it is observed that Vicuna, which was fine-tuned on LLaMA on about 70k high-quality dialog samples from ShareGPT (Chiang et al., 2023), has experienced different levels of commonsense degradation. In the following sections, we delve into the phenomenon of forgetting and provide practical suggestions.

Suspecting the finetuning process caused the model to forget its commonsense knowledge, we further conduct experiments of finetuning LLaMA-7b with different sizes of subsets. Due to limited computational resources, we used a sample of 25,000 as 100% setting and fine-tuned the model for 3 epochs in this section. As demonstrated in Table 4, a huge commonsense performance drop is witnessed after training on a small number of data samples, while the degradation continues to affect the model when more samples are provided. Moreover, this phenomenon is general enough to be observed in other models (Table 5), making it unneglectable during the development of Large Language Models. Also, we noticed a strong agreement between the NLL and accuracy metrics in terms of the changing trend of LLMs’ performance on commonsense QA tasks. This provides further evidence of the effectiveness of our proposed NLL metric.

As larger finetuning datasets lead to more serious forgetting issues, an implication of this phenomenon is a rule of thumb widely adopted in practice: *Less forgetting calls for smaller datasets, and smaller datasets ask for higher quality samples*. Continuously increasing the size of the training dataset can undoubtedly harm the original ability of LLMs. It is advisable to carefully consider the trade-offs between dataset size and sample quality.

4.3.2 Overfitting

Overfitting is another insight we observe during the finetuning of large language models. Surprisingly, the commonly used 3-epoch finetuning setting (Chiang et al., 2023) results in overfitting for several popular models.

	Commonsense QA	
	NLL	Acc.
LLaMA-7b Before tuning	113.71	63.35
LLaMA-7b After tuning	155.50	60.36
BLOOM-7b Before tuning	129.57	56.38
BLOOM-7b After tuning	176.93	54.13
GPT-J-6b Before tuning	129.11	59.10
GPT-J-6b After tuning	162.65	54.82
OPT-6.9b Before tuning	124.14	58.94
OPT-6.9b After tuning	147.79	57.54

Table 5: The table shows various LLMs trained on sampled datasets for 3 epochs, demonstrating the general phenomenon of forgetting across different LLMs.

Models	Train	Dialog	Instruction
LLaMA-7b (1 epoch)	486 \pm 4.9	150 \pm 1.2	181 \pm 1.4
LLaMA-7b (2 epochs)	289 \pm 5.1	161 \pm 1.0	189 \pm 1.2
LLaMA-7b (3 epochs)	234 \pm 5.3	195 \pm 2.3	227 \pm 2.7
BLOOM-7b (1 epoch)	530 \pm 5.9	175 \pm 1.5	217 \pm 2.0
BLOOM-7b (2 epochs)	289 \pm 6.5	184 \pm 1.4	222 \pm 1.2
BLOOM-7b (3 epochs)	221 \pm 7.2	228 \pm 2.8	273 \pm 3.2
GPT-J-6b (1 epoch)	534 \pm 2.1	163 \pm 1.4	199 \pm 2.2
GPT-J-6b (2 epochs)	306 \pm 6.1	176 \pm 1.8	209 \pm 2.3
GPT-J-6b (3 epochs)	238 \pm 9.5	213 \pm 3.4	251 \pm 4.7

Table 6: Different LLMs were trained on a sampled dialog dataset, and it is evident that their performance on the dialog and instruction evaluation sets deteriorated as the number of epochs increased for each model.

Surprisingly, the commonly used 3-epoch fine-tuning setting (Chiang et al., 2023) resulted in overfitting for several popular models. As the model is trained for more epochs, it tends to generate responses that are similar to those present in the training data but perform poorly on the test data. In this study, we fine-tuned three models, LLaMA-7b, Bloom-7.1b, and GPT-J-6b, for three epochs and evaluated their performance in terms of dialog and instruction following. Similar to the previous section on forgetting, we used a sample of 25,000 for the training set and fine-tuned the model due to limited computing resources.

This overfitting phenomenon is elaborated in Table 6 and Figure 2. Table 6 reports the mean and standard deviation of NLL results over 3 runs for each combination of models and test sets. It can be observed that the NLL metrics continue to worsen for each model with an increasing number of epochs. In addition, Figure 2 further visualizes the statistics in the table, which clearly illustrates the trend of training performance improvement and testing performance drop as the number of epochs increases, indicating overfitting. Moreover, we used our proposed NLL metric to measure the overfitting issue, which is shown to clearly indicate the changing trend.

These experiments have taught us a valuable lesson. *It is recommended to switch from the commonly used 3-epoch fine-tuning setting to a single-epoch setting, as LLMs tend to generalize better with the latter.*

4.3.3 Hyperparameter Tuning

The last insight we offer is a set of recommended hyperparameters for LLaMA, one of the most popular foundation models adopted in the LLM community. By making the training details and result logs open source, we aim to assist practitioners with improving the training of their instruction-tuned LLMs and developing more task-oriented models. This aligns with the goal of making large language models accessible for all target applications.

This setting utilizes a 615M-sized high-quality dataset, which comprises \sim 130k dialog samples and \sim 52k

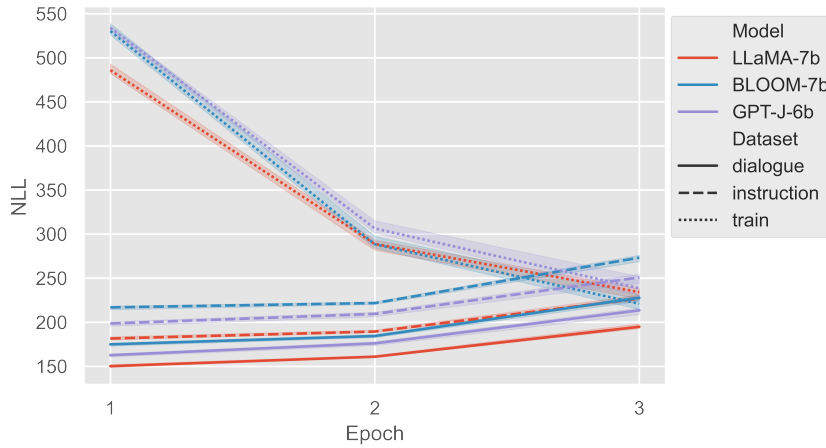


Figure 2: Visualization of the changing trends regarding the overfitting issue of various LLMs.

Setting	Best Choice
Mini-batch size	64
Learning rate	1e-5
Training epochs	1
Block size	2048

Table 7: Best group of hyperparameters on training LLaMA-7b.

	Dialog	Inst	CS-NLL	CS-Acc
reference model	138	170	114.43	67.13
mini-batch size 8	174	211	113.72	63.08
mini-batch size 32	138	169	114.79	64.42
learning rate 1e-4	144	172	120.39	62.51
learning rate 5e-4	173	204	147.86	52.59
training epochs 2	141	169	116.29	64.70
training epochs 3	144	170	119.47	64.59
block size 512	141	171	115.00	64.28
block size 1024	140	171	115.08	64.60

Table 8: An ablation study was conducted for different hyperparameter settings. The reference model used the best settings for training LLaMA, which were found in Table 7, i.e., the settings of Robin-7b as shown in Table 1. Besides, "CS" means commonsense tasks in this table.

instruction following samples. Similar to Chiang et al. (2023), dialog samples are gathered from ShareGPT.com with public APIs. Also, we collect an English subset of Peng et al. (2023) to provide instruction following training samples. For details about the collection of instruction tuning data, please refer to the appendix C.

By adjusting different hyperparameters, one can observe that the most influential parameters are mini-batch sizes and learning rates, where too small mini-batch sizes or too large learning rates all lead to significant performance degradation. On the other hand, the overfitting phenomenon becomes less severe under this setting, as a consequence of the large size of our dataset. Also, it is worth noting that the NLL and accuracy metrics exhibit a strong correlation, which further demonstrates the effectiveness of the NLL metric. With the guidance of the NLL metric, we were able to develop our powerful Robin series model.

5 Conclusion

This paper introduces LMFlow Benchmark, a versatile evaluation paradigm that can automatically, efficiently, and scalably measure multiple aspects of Large Language Models (LLMs). The core of this framework is a novel view of Negative Log Likelihood (NLL), followed under the principle of “good prediction reflects good generation”. By defining the measuring aspect with a set of reference samples, the NLL metric can be used to perform comprehensive comparisons among multiple existing LLMs, which are presented with their strengths in three different aspects: dialog, instruction following, and commonsense. Besides the convenience of NLL, our experimental results further demonstrate a strong correlation between NLL and traditional metrics, rendering it a proper alternative for evaluation in the era of LLM. On top of that, we discovered helpful insights for finetuning practices based on NLL: 1) using small-sized high-quality datasets can alleviate forgetting, 2) the commonly used 3-epoch setting often leads to overfitting, and 3) hyperparameter choice can significantly affect the finetuning performance of LLaMA.

Limitations

One limitation of LMFlow Benchmark is that it only provides datasets on three aspects of model evaluation. With the general metrics of NLL, more dimensions of models can be examined, such as multilingual ability, logical reasoning, different styles, ethics, sentiments, performance in various domains, etc. These all require high-quality reference dataset prepared by human. The good news is, according to our experience, a set of hundreds of samples normally suffices to form a valid reference dataset, which usually takes an expert only few hours to prepare. We hope more types of such reference datasets can be collected and proposed in the future.

References

- S. AI. Stablelm: Stability ai language models, April 2023. URL <https://github.com/Stability-AI/StableLM>.
- E. Almazrouei, H. Alobeidli, A. Alshamsi, A. Cappelli, R. Cojocaru, M. Debbah, E. Goffinet, D. Heslow, J. Launay, Q. Malartic, B. Noune, B. Pannier, and G. Penedo. Falcon-40B: an open large language model with state-of-the-art performance. 2023.
- R. Anil, A. M. Dai, O. Firat, M. Johnson, D. Lepikhin, A. Passos, S. Shakeri, E. Taropa, P. Bailey, Z. Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.
- Anthropic. Introducing claude. *Anthropic Blog*, 2022. URL <https://www.anthropic.com/index/introducing-claude>.
- Y. Bai, A. Jones, K. Ndousse, A. Askell, A. Chen, N. DasSarma, D. Drain, S. Fort, D. Ganguli, T. Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.
- S. Biderman, H. Schoelkopf, Q. Anthony, H. Bradley, K. O’Brien, E. Hallahan, M. A. Khan, S. Purohit, U. S. Prashanth, E. Raff, A. Skowron, L. Sutawika, and O. van der Wal. Pythia: A suite for analyzing large language models across training and scaling. *CoRR*, abs/2304.01373, 2023. doi: 10.48550/arXiv.2304.01373. URL <https://doi.org/10.48550/arXiv.2304.01373>.
- Y. Bisk, R. Zellers, J. Gao, Y. Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439, 2020.
- S. Black, L. Gao, P. Wang, C. Leahy, and S. Biderman. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow, Mar. 2021. URL <https://doi.org/10.5281/zenodo.5297715>. If you use this software, please cite it using these metadata.

- T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- W. G. Chase and H. A. Simon. Perception in chess. *Cognitive psychology*, 4(1):55–81, 1973.
- E. Chen, R. Huang, H.-S. Chen, Y.-H. Tseng, and L.-Y. Li. Gptutor: a chatgpt-powered programming tool for code explanation. *arXiv preprint arXiv:2305.01863*, 2023.
- W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, I. Stoica, and E. P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL <https://lmsys.org/blog/2023-03-30-vicuna/>.
- C. Clark, K. Lee, M.-W. Chang, T. Kwiatkowski, M. Collins, and K. Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*, 2019.
- P. Clark, I. Cowhey, O. Etzioni, T. Khot, A. Sabharwal, C. Schoenick, and O. Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.
- Databricks. Dolly, March 2023. URL <https://github.com/databrickslabs/dolly>.
- A. D. De Groot. Thought and choice in chess. In *Thought and Choice in Chess*. De Gruyter Mouton, 2014.
- T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*, 2023.
- A. E. Elo. The proposed uscf rating system. its development, theory, and applications. *Chess Life*, 22(8):242–247, 1967.
- A. E. Elo. *The rating of chessplayers, past and present*. Arco Pub., 1978.
- Z. Guo, P. Wang, Y. Wang, and S. Yu. Dr. llama: Improving small language models in domain-specific qa via generative data augmentation. *arXiv preprint arXiv:2305.07804*, 2023.
- S. Hegselmann, A. Buendia, H. Lang, M. Agrawal, X. Jiang, and D. Sontag. Tabllm: Few-shot classification of tabular data with large language models. In *International Conference on Artificial Intelligence and Statistics*, pages 5549–5581. PMLR, 2023.
- D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.
- J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. d. L. Casas, L. A. Hendricks, J. Welbl, A. Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.
- J. Jang, S. Ye, S. Yang, J. Shin, J. Han, G. Kim, S. J. Choi, and M. Seo. Towards continual knowledge learning of language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL <https://openreview.net/forum?id=vfsRB5MImo9>.
- R. Li, L. B. Allal, Y. Zi, N. Muennighoff, D. Kocetkov, C. Mou, M. Marone, C. Akiki, J. Li, J. Chim, Q. Liu, E. Zheltonozhskii, T. Y. Zhuo, T. Wang, O. Dehaene, M. Davaadorj, J. Lamy-Poirier, J. Monteiro, O. Shliazhko, N. Gontier, N. Meade, A. Zebaze, M.-H. Yee, L. K. Umapathi, J. Zhu, B. Lipkin, M. Oblukulov, Z. Wang, R. Murthy, J. Stillerman, S. S. Patel, D. Abulkhanov, M. Zocca, M. Dey, Z. Zhang, N. Fahmy, U. Bhattacharyya, W. Yu, S. Singh, S. Luccioni, P. Villegas, M. Kunakov, F. Zhdanov, M. Romero, T. Lee, N. Timor, J. Ding, C. Schlesinger, H. Schoelkopf, J. Ebert, T. Dao, M. Mishra, A. Gu, J. Robinson, C. J. Anderson, B. Dolan-Gavitt, D. Contractor, S. Reddy, D. Fried, D. Bahdanau, Y. Jernite, C. M. Ferrandis, S. Hughes, T. Wolf, A. Guha, L. von Werra, and H. de Vries. Starcoder: may the source be with you! 2023a.

- T. Li, S. Shetty, A. Kamath, A. Jaiswal, X. Jiang, Y. Ding, and Y. Kim. Cancergpt: Few-shot drug pair synergy prediction using large pre-trained language models. *arXiv preprint arXiv:2304.10946*, 2023b.
- P. Liang, R. Bommasani, T. Lee, D. Tsipras, D. Soylu, M. Yasunaga, Y. Zhang, D. Narayanan, Y. Wu, A. Kumar, et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022.
- C.-Y. Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL <https://aclanthology.org/W04-1013>.
- H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning, 2023.
- M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993. URL <https://aclanthology.org/J93-2004>.
- T. Mihaylov, P. Clark, T. Khot, and A. Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*, 2018.
- E. Nijkamp, B. Pang, H. Hayashi, L. Tu, H. Wang, Y. Zhou, S. Savarese, and C. Xiong. Codegen: An open large language model for code with multi-turn program synthesis. *arXiv preprint arXiv:2203.13474*, 2022.
- OpenAI. Introducing chatgpt. 2022. URL <https://openai.com/blog/chatgpt>.
- OpenAI. Gpt-4 technical report, 2023.
- K. Papineni, S. Roukos, T. Ward, and W. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL, 2002. doi: 10.3115/1073083.1073135. URL <https://aclanthology.org/P02-1040/>.
- B. Peng, C. Li, P. He, M. Galley, and J. Gao. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*, 2023.
- R. Pi, J. Gao, S. Diao, R. Pan, H. Dong, J. Zhang, L. Yao, J. Han, H. Xu, and L. K. T. Zhang. Detgpt: Detect what you need via reasoning. *arXiv preprint arXiv:2305.14167*, 2023.
- A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
- J. W. Rae, S. Borgeaud, T. Cai, K. Millican, J. Hoffmann, F. Song, J. Aslanides, S. Henderson, R. Ring, S. Young, et al. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*, 2021.
- P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.
- K. Sakaguchi, R. L. Bras, C. Bhagavatula, and Y. Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.
- T. L. Scao, A. Fan, C. Akiki, E. Pavlick, S. Ilic, D. Hesslow, R. Castagné, A. S. Luccioni, F. Yvon, M. Gallé, J. Tow, A. M. Rush, S. Biderman, A. Webson, P. S. Ammanamanchi, T. Wang, B. Sagot, N. Muennighoff, A. V. del Moral, O. Ruwase, R. Bawden, S. Bekman, A. McMillan-Major, I. Beltagy, H. Nguyen, L. Saulnier, S. Tan, P. O. Suarez, V. Sanh, H. Laurençon, Y. Jernite, J. Launay, M. Mitchell, C. Raffel, A. Gokaslan, A. Simhi, A. Soroa, A. F. Aji, A. Alfassy, A. Rogers, A. K. Nitzav, C. Xu, C. Mou, C. Emezue, C. Klammer, C. Leong, D. van Strien, D. I. Adelani, and et al. BLOOM: A 176b-parameter open-access multilingual language model. *CoRR*, abs/2211.05100, 2022. doi: 10.48550/arXiv.2211.05100. URL <https://doi.org/10.48550/arXiv.2211.05100>.

- R. Schaeffer, B. Miranda, and S. Koyejo. Are emergent abilities of large language models a mirage? *arXiv preprint arXiv:2304.15004*, 2023.
- T. Sellam, D. Das, and A. P. Parikh. Bleurt: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696*, 2020.
- R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, Oct. 2013. Association for Computational Linguistics. URL <https://aclanthology.org/D13-1170>.
- A. Srivastava, A. Rastogi, A. Rao, A. A. M. Shueb, A. Abid, A. Fisch, A. R. Brown, A. Santoro, A. Gupta, A. Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022.
- Y. Sun, S. Wang, Y. Li, S. Feng, H. Tian, H. Wu, and H. Wang. ERNIE 2.0: A continual pre-training framework for language understanding. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8968–8975. AAAI Press, 2020. URL <https://ojs.aaai.org/index.php/AAAI/article/view/6428>.
- R. Taylor, M. Kardas, G. Cucurull, T. Scialom, A. Hartshorn, E. Saravia, A. Poulton, V. Kerkez, and R. Stojnic. Galactica: A large language model for science. *CoRR*, abs/2211.09085, 2022. doi: 10.48550/arXiv.2211.09085. URL <https://doi.org/10.48550/arXiv.2211.09085>.
- M. N. Team. Introducing mpt-7b: A new standard for open-source, commercially usable llms, March 2023. URL <https://www.mosaicml.com/blog/mpt-7b>.
- Together. Releasing 3b and 7b redpajama-incite family of models including base, instruction-tuned & chat models, May 2023. URL <https://www.together.xyz/blog/redpajama-models-v1>.
- H. Touvron, T. Lavril, G. Izacard, X. Martinet, M. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971, 2023. doi: 10.48550/arXiv.2302.13971. URL <https://doi.org/10.48550/arXiv.2302.13971>.
- A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.
- B. Wang and A. Komatsuzaki. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. <https://github.com/kingoflolz/mesh-transformer-jax>, May 2021.
- L. Wang and E.-P. Lim. Zero-shot next-item recommendation using large pretrained language models. *arXiv preprint arXiv:2304.03153*, 2023.
- P. Wang, L. Li, L. Chen, D. Zhu, B. Lin, Y. Cao, Q. Liu, T. Liu, and Z. Sui. Large language models are not fair evaluators. *arXiv preprint arXiv:2305.17926*, 2023a.
- X. Wang, X. Zhang, Y. Cao, W. Wang, C. Shen, and T. Huang. Seggpt: Segmenting everything in context. *arXiv preprint arXiv:2304.03284*, 2023b.
- Y. Wang, Z. Yu, Z. Zeng, L. Yang, C. Wang, H. Chen, C. Jiang, R. Xie, J. Wang, X. Xie, et al. Pandalm: An automatic evaluation benchmark for llm instruction tuning optimization. *arXiv preprint arXiv:2306.05087*, 2023c.
- J. Wei. 137 emergent abilities of large language models. 2022. URL <https://www.jasonwei.net/blog/emergence>.

- S. Wu, O. Irsoy, S. Lu, V. Dabrovolski, M. Dredze, S. Gehrmann, P. Kambadur, D. Rosenberg, and G. Mann. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*, 2023.
- H. Yang, X.-Y. Liu, and C. D. Wang. Fingpt: Open-source financial large language models. *arXiv preprint arXiv:2306.06031*, 2023.
- R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, and Y. Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.
- S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. T. Diab, X. Li, X. V. Lin, T. Mihaylov, M. Ott, S. Shleifer, K. Shuster, D. Simig, P. S. Koura, A. Sridhar, T. Wang, and L. Zettlemoyer. OPT: open pre-trained transformer language models. *CoRR*, abs/2205.01068, 2022. doi: 10.48550/arXiv.2205.01068. URL <https://doi.org/10.48550/arXiv.2205.01068>.
- L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. P. Xing, H. Zhang, J. E. Gonzalez, and I. Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023a.
- L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. P. Xing, H. Zhang, J. E. Gonzalez, and I. Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023b.
- D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

A GPT4 prompt

B Compare to ChatGPT Evaluation

On top of Human evaluation and GPT-4 evaluation, we further use ChatGPT (gpt-3.5-turbo) to rank the conversational performance of different models. Based on the evaluation setting of Vicuna (Chiang et al., 2023), we prompt ChatGPT to score models’ generated responses pairwise. Given a query and two responses generated by two different models, ChatGPT is asked to rate a score out of ten for both responses. We compare 9 models in pairs, which in total, gives 36 battle results. According to the win-lose rate (excluding ties), we summarize the results in Figure 3.

According to the results, the trend generally matches the rank produced by NLL metric where Vicuna-7b is far better than other base models, followed by LLaMA-13b and LLaMA-7b. Also, Bloom-3b and OPT-2.7b are the two worst-performing models.

However, it is still observed that ChatGPT has verbosity bias compared to GPT-4 and human evaluation (Zheng et al., 2023b). As shown in Table 10, OPT-6.9b gives a longer, verbose response while Pythia-6.9b gives a shorter but more clear and more accurate response. ChatGPT shows its favor for longer but low-quality responses by rating 8 for Pythia-6.9b and rating 9 for OPT-6.9b. GPT-4 is much more robust to verbosity bias and only rates 1 for such repetitive, meaningless responses.

C Detailed Evaluation Results

In order to make the full evaluation results more accessible to readers, we only show average results for each dimension (“Dialog”, “Commonsense NLL” and “Commonsense Accuracy”) on Table 1.

For computing NLL, we have two evaluation formats. One is termed as “text-only” next. It only has one input text and will compute NLL for all the tokens in the input text. Another is termed “text-to-text”, which has an input text and a target output text. During computing NLL, two texts will be concatenated and NLL computation will be blocked on the input text part and begin computing on the output text. The detailed results are shown in this section.

Dialog. Dialog dataset is a series of Human-Assistant conversations collected to evaluate LLM’s multi-turn chat ability. It is carefully selected by human annotators to construct high-quality conversation evaluation

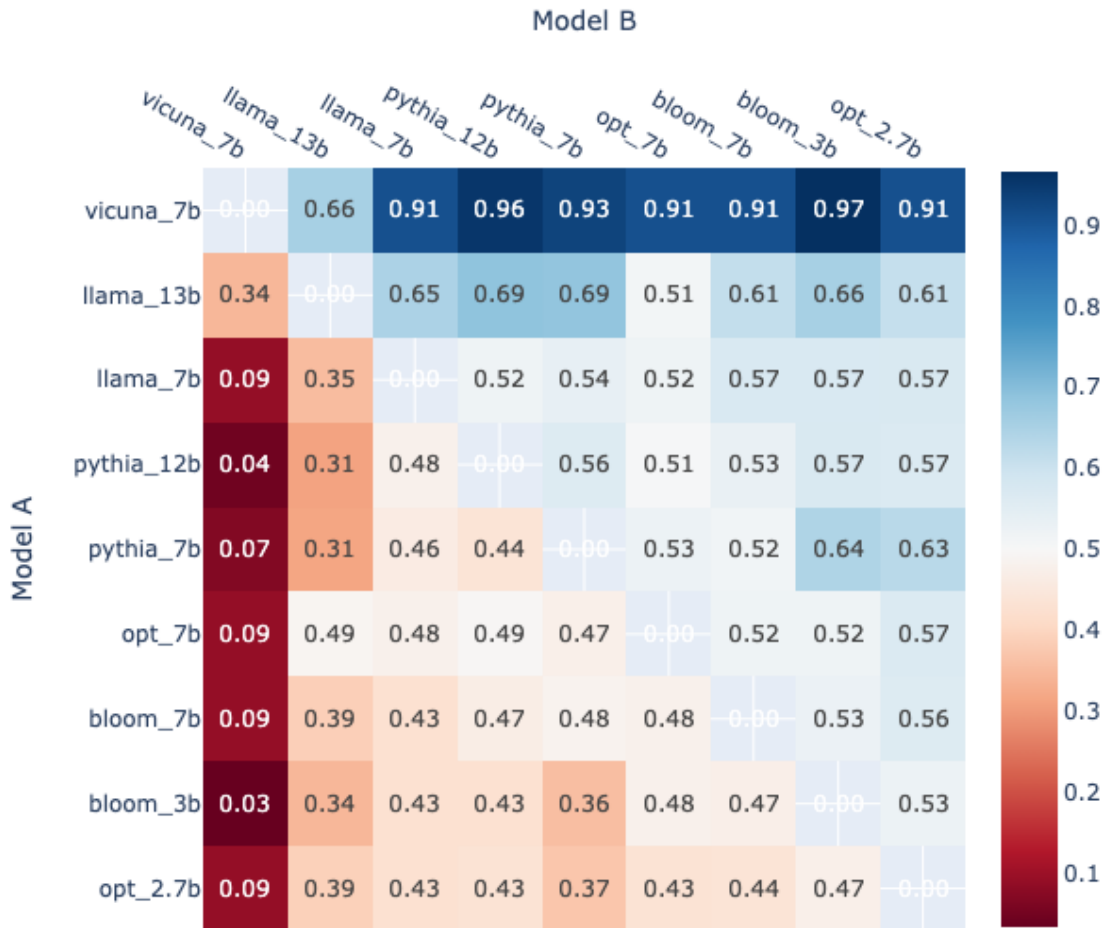
[Conversation Context]
[Query Context]
[The Start of Assistant 1’s Response]
[Model A’s Response]
[The End of Assistant 1’s Response]
[The Start of Assistant 2’s Response]
[Model B’s Response]
[The End of Assistant 2’s Response]
[System]
<p>We would like to request your feedback on the performance of two AI assistants in response to the user question displayed above.</p> <p>Please rate the helpfulness, relevance, accuracy, level of details of their responses. Each assistant receives an overall score on a scale of 1 to 10, where a higher score indicates better overall performance.</p> <p>Please first output a single line containing only two values indicating the scores for Assistant 1 and 2, respectively. The two scores are separated by a comma. In the subsequent line, please provide a comprehensive explanation of your evaluation, avoiding any potential bias and ensuring that the order in which the responses were presented does not affect your judgment.</p>

Table 9: GPT-4 evaluation prompt template with three slots (Conversation Context, Model A’s Response and Model B’s Response), which is first introduced in Chiang et al. (2023). We adapted tag messages and templates for conversation evaluation.

benchmarks. From 183 high-quality multi-turn conversation records, we collect 477 multi-turn dialog response generation examples and 477 single-turn dialog examples. More specifically, suppose we have a record with "Q1,A1,Q2,A2,Q3,A3", single-turn examples are like "Q1—A1; Q2—A2; Q3—A3", while multi-turn examples are like "Q1—A1; Q1,A1,Q2—A2; Q1,A1,Q2,A2,Q3—A3". By setting these contextual dialogs as input and the final answer as targeted output, we can use NLL to evaluate LLMs’ performance in the generation of correct answers. This dialog evaluation has the input-output format and used the "text-to-text" way to compute the results. Detailed results of various LLMs’ dialog NLL evaluation are shown in Table 11.

Instruction. GPT-4-LLM (Peng et al., 2023) provides ~52,000 high-quality instruction following examples by calling GPT-4 API from 1,000 seed examples. We manually filtered 767 examples from 1,000 examples to construct an evaluation for LLMs’ instruction following ability. Here, examples with too many nonsense words, incomplete responses and highly domain-specific content will be removed. This instruction evaluation also has the input-output format and used the "text-to-text" way to compute the results. Detailed results have been shown on Table 1.

Commonsense. Commonsense knowledge is often acknowledged as a core evaluation setting for LLMs. We follow Brown et al. (2020); Touvron et al. (2023) on evaluating multiple choice grades based on these datasets: BoolQ (Clark et al., 2019), PIQA (Bisk et al., 2020), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2021), ARC easy and challenge (Clark et al., 2018) and OpenBook QA (Mihaylov et al., 2018). Besides, we transform these data examples into plain text by concatenating questions with their correct solutions. By using this plain text, we can use NLL to evaluate LLMs’ commonsense knowledge. We could observe strong agreement between Commonsense QA accuracy and NLL’s average in Table 1. Table 12 and Table 13 are detailed results of LLMs on 7 commonsense tasks. Here, the "text-only" method is applied to get NLL results.



Fraction of Model A Wins for All Non-tied A vs. B Battles

Figure 3: Win fraction of Model A over Model B for all Non-tied battles

Question	Please help me write a copywriting for an oolong tea brand phone case
Pythia-6.9b's response	What is the name of the oolong tea brand?
OPT-6.9b's response	<p>Hello, I am a professional copywriter with over 5 years of experience. I have read your project description and I am very interested in working on this project. I am a native English speaker and I have a strong background in writing for the web. I have worked on a variety of projects, including copywriting, SEO, content, and more. I am a fast learner and I am always up for a challenge. I am a very hard worker and I am always looking for ways to improve my skills. I am looking forward to hearing from you. Regards, Mais</p> <p>Hello, I am a professional copywriter with over 5 years of experience. I have read your project description and I am very interested in working on this project. I am a native English speaker and I have a strong background in writing for the web. I have worked on a variety of projects, including copywriting, SEO, content, and more. I am a fast learner and I am always up for a challenge. I am a very hard worker and I am always looking for ways to improve my skills. I am looking forward to hearing from you. Regards, Mais</p> <p>Hello, I am a professional copywriter with over 5 years of experience. I have read your project description and I am very interested in working on this project. I am a native English speaker and I have a strong background in writing for the web. I have worked on a variety of projects, including copywriting, SEO, content, and more. I am a fast learner and I am always up for a challenge. I am a very hard worker and I am always looking for ways to improve my skills. I am looking forward to hearing from you. Regards, Mais</p> <p>Hello, I am a professional copywriter with over' (verbose response)</p>
ChatGPT's Review	8 for Pythia-6.9b , 9 for OPT-6.9b Both assistants provided concise and natural revisions of the original script. Assistant 1's response was helpful in providing context for the scene, but Assistant 2's response was more accurate and detailed in describing the character's clothing and the specific type of buttons on his collar. Overall, both assistants performed well, but Assistant 2's response was slightly more relevant and accurate." [WRONG]
GPT4's Review	7 for Pythia-6.9b , 1 for OPT-6.9b Assistant 1 has provided a useful follow-up question to clarify the user's request, which is critical to tailoring the copywriting text according to the specific brand's characteristics. Although it's a short response, it's a good starting point for producing a meaningful and effective piece of copywriting. On the other hand, Assistant 2 seems to have misunderstood the user's request completely. It provides a pre-written, generalized response about their capabilities as a copywriter instead of helping with the task at hand. The response is not only irrelevant but is also repeated four times, which appears to be an error. The user wanted help in writing a copy for a phone case related to an oolong tea brand, but Assistant 2 did not address this at all. Thus, Assistant 2 receives a low score due to irrelevance and repetition. [CORRECT]

Table 10: Verbosity bias example of ChatGPT evaluation.

	single-turn	multi-turn	average
CodeGen-6b-mono	262	288	275
CodeGen-6b-multi	251	282	266.5
StableLM-3b	244	266	255
CodeGen-16b-mono	226	253	239.5
StableLM-7b	226	252	239
Dolly-12b	222	250	236
GPT-2-XL	221	247	234
CodeGen-16b-multi	219	246	232.5
Galactica-1.3b	222	237	229.5
BLOOM-3b	202	228	215
OPT-2.7b	200	225	212.5
Dolly-7b	201	223	212
Dolly-3b	199	221	210
GPT-neo-2.7b	196	220	208
Galactica-6.9b	194	218	206
BLOOM-7.1b	188	213	200.5
pythia-2.8b	188	211	199.5
OPT-6.9b	184	210	197
OPT-13b	179	205	192
pythia-6.9b	180	203	191.5
pythia-12b	178	201	189.5
GPT-J-6b	176	199	187.5
RedPajama-3b	163	193	178
RedPajama-7b	161	187	174
LLaMA-7b	163	184	173.5
LLaMA-13b	157	177	167
Vicuna-7b	155	175	165
Vicuna-13b	150	170	160
Robin-7b	137	152	144.5
Robin-13b	127	147	137

Table 11: Detailed evaluation results of different LLMs on Dialog tasks.

	winogrande	boolq	arc_easy	hellaswag	piqa	OBQA	arc_challenge	Average
CodeGen-6b-mono	51.07	61.28	44.19	36.14	60.12	26.2	23.72	43.25
CodeGen-6b-multi	54.78	62.26	47.52	41.22	62.35	29.4	24.83	46.06
StableLM-3b	52.64	57.65	44.74	38.31	63.82	29.4	25.77	44.62
CodeGen-16b-mono	56.83	62.29	52.44	44.62	64.91	30.6	26.96	48.38
StableLM-7b	50.55	60.06	50.55	41.22	67.36	32	27.05	46.97
Dolly-12b	61.48	57.19	67	70.69	75.24	40.2	37.37	58.46
GPT-2-XL	58.33	61.77	58.29	50.89	70.84	32	28.5	51.52
CodeGen-16b-multi	57.7	62.84	53.45	50.7	65.67	31.4	28.41	50.03
Galactica-1.3b	54.3	62.05	62.42	41.04	63.28	30.2	30.63	49.14
BLOOM-3b	58.72	61.62	59.47	52.72	70.84	42.2	30.55	53.74
OPT-2.7b	60.77	60.37	60.77	60.63	74.76	35.2	31.23	54.82
Dolly-7b	60.69	64.65	67.21	68.8	75.03	39.6	40.36	59.48
Dolly-3b	59.83	57.83	64.9	64.96	74.32	39.4	37.12	56.91
GPT-neo-2.7b	57.7	61.77	61.07	55.8	72.91	33.2	30.2	53.24
Galactica-6.9b	57.46	66.79	70.54	51.13	68.72	37	38.23	55.7
BLOOM-7.1b	64.4	62.91	65.03	59.64	73.56	35.8	33.36	56.39
pythia-2.8b	58.64	64.31	63.59	59.49	74.16	35	32.94	55.45
OPT-6.9b	65.19	66.06	65.57	67.19	76.5	37.4	34.64	58.94
OPT-13b	65.04	65.9	67.13	69.81	76.88	39	35.67	59.92
pythia-6.9b	62.67	64.4	68.52	65.79	76.82	38.8	35.58	58.94
pythia-12b	66.14	65.84	70.75	68.83	77.04	40.2	38.31	59.4
GPT-J-6b	64.09	65.44	66.92	66.25	76.17	38.2	36.6	59.1
RedPajama-3b	65.11	71.01	69.91	70.16	76.66	38.6	38.4	61.41
RedPajama-7b	65.59	65.02	71.76	69.51	77.48	41.6	39.68	61.52
LLaMA-7b	67.9	73.15	67.3	72.96	78.35	42.4	41.38	63.35
LLaMA-13b	70.01	68.53	74.54	76.23	79.11	42.2	44.54	65.03
Vicuna-7b	63.69	77.37	63.13	68.82	76.33	39.6	38.74	61.1
Vicuna-13b	66.22	79.85	64.65	73.04	77.64	41.6	40.36	63.34
Robin-7b	70.32	78.78	75.67	76.09	79.27	44.6	45.14	67.13
Robin-13b	70.8	70.49	74.41	76.27	79.82	43.6	45.9	65.9

Table 12: Detailed evaluation results of different LLMs on Commonsense QA accuracy tasks.

	winogrande	boolq	arc_easy	hellaswag	piqa	OBQA	arc_challenge	Average
CodeGen-6b-mono	105	314	110.5	314	169	78	129	174.22
CodeGen-6b-multi	104.5	298	110	306	169	77.5	128	170.43
StableLM-3b	92	304	101	274	167	69	120.5	161.08
CodeGen-16b-mono	99.5	284	103	290	159	74	120.5	161.43
StableLM-7b	91.5	284	98.5	246	173	68	118	154.15
Dolly-12b	92	252	92	296	157	68.5	112.5	152.86
GPT-2-XL	85.5	228	87.5	243	134	62.75	103.5	134.9
CodeGen-16b-multi	98.5	270	101.5	282	158	73.5	118	157.36
Galactica-1.3b	97	242	88	278	153	68	103.5	147.08
BLOOM-3b	86.5	228	86	245	134	64.5	101.5	135.08
OPT-2.7b	83	230	85	230	128	61.75	100	131.11
Dolly-7b	91	242	90.5	274	151	68	109	146.5
Dolly-3b	89.5	238	90.5	268	145	67	109	143.86
GPT-neo-2.7b	85	218	85	240	132	62.75	100.5	131.9
Galactica-6.9b	92.5	215	81	260	143	65.6	96	136.16
BLOOM-7.1b	85	215	81.5	237	130	62.5	96	129.58
pythia-2.8b	84	210	85.5	240	130	62.75	100.5	130.4
OPT-6.9b	81.5	200	81.5	224	125	61	96	124.15
OPT-13b	82	688	82.5	198	125	61.75	97	190.61
pythia-6.9b	83	202	81.5	232	127	61.5	96	126.15
pythia-12b	83	198	79.5	232	127	61	94	124.93
GPT-J-6b	83	232	80	227	126	60.75	95	129.11
RedPajama-3b	81.5	178	74.5	227	126	60.5	88	119.36
RedPajama-7b	82.5	178	75.5	227	125.5	59.75	89	119.61
LLaMA-7b	79.5	167	71.5	214	121	58	85	113.72
LLaMA-13b	79	153	70	207	119	57.25	83	109.75
Vicuna-7b	97	219	89	258	151	72	105	141.58
Vicuna-13b	95.5	207	88.5	262	149	73	105	140
Robin-7b	79.5	172	72	212	122	58.5	85	114.43
Robin-13b	79	159	70.5	207	120	58.25	83.5	111.04

Table 13: Detailed evaluation results of different LLMs on Commonsense QA NLL tasks.