Smaller Language Models are capable of selecting Instruction-Tuning Training Data for Larger Language Models

Anonymous ACL submission

Abstract

Instruction-tuning language models has become a crucial step in aligning them for general use. Typically, this process involves extensive training on large datasets, incurring high training costs. In this paper, we introduce a novel training data selection based on the learning percentage of the samples. We assert that current language models possess the capability to autonomously select high-quality training data, leading to comparable or improved performance compared to training on the entire dataset. Our experiments span different-sized models, revealing that this characteristic holds for models ranging from 1B (small) to 13B (large) in size. Moreover, we demonstrate an interesting finding that the data hardness transfers across model sizes, and a smaller 350M model can effectively curate high-quality training data with hard samples for a larger 13B model, resulting in an equally or superior instruction-tuned model compared to training on the complete dataset. Utilizing open-sourced OPT and Llama-2 models up to 13B in size, two publicly available instruction-tuning training datasets and evaluated by both automatic metrics & humans, our paper introduces a novel approach to training data selection, showcasing a more efficient alternative.

1 Introduction

Instruction tuning empowers large language models (LLMs) to generalize to novel tasks and instills an instruction-following characteristic, marking the initial stride towards aligning them for general use (Sanh et al., 2022; Wei et al., 2022, 2021; Chung et al., 2022; Wei et al., 2022). This process involves fine-tuning language models with extensive sets of real (Mishra et al., 2021; Wang et al., 2022b) and/or synthetic instructions (Wang et al., 2022a; Honovich et al., 2023). Given that these datasets are typically vast, encompassing thousands of samples, the training costs associated with this approach are notably high.

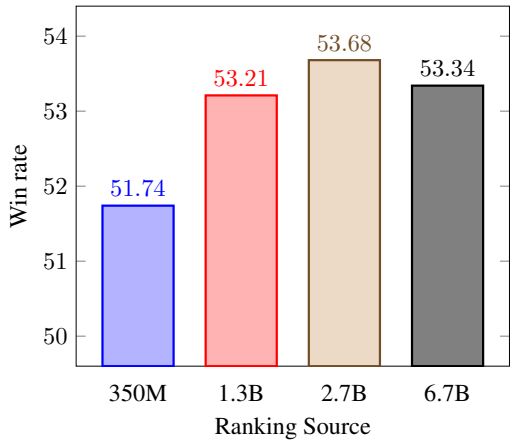


Figure 1: The win rate of OPT-13B model trained on 10% data sub-sampled by smaller OPT models (350M, 1.3B, 2.7B) from Alpaca Data, is compared against the OPT-13B model trained on the full dataset. All win rates exceed 50, indicating even a smaller 350M dataset can curate high-quality data for a larger 13B model.

Past research into the memorization effects of deep neural networks have revealed a tendency to memorize easy instances first and gradually learn more challenging instances towards the end (Arpit et al., 2017; Geifman et al., 2019; Zhang et al., 2021; Mekala et al., 2022a). Additionally, (Swayamdipta et al., 2020) show that ambiguous and hard samples in training data are sufficient for achieving good generalization. The process of data selection, wherein subsets are chosen from extensive training data to achieve superior performance, has attracted significant attention among researchers recently. Earlier studies involved manual feature engineering of various indicators from the data (Cao et al., 2023), training large custom models (Li et al., 2023a), or employing closed LLMs like GPT-3.5 (Chen et al., 2024) for data selection.

In this paper, we delve into the measurement of sample difficulty from the model's perspective. Drawing inspiration from the learning order metric in (Mekala et al., 2022a), we propose a novel

data selection method that utilizes the learning percentage as a difficulty metric that the model can use to self-rank its training data. Essentially, the more learning that occurs in earlier epochs, the easier the sample is considered. We then select the most difficult sample subsets based on this ranking and instruction-tune a language model. Our experiments involve two instruction-tuning datasets, Alpaca-Data (Taori et al., 2023), and Dolly (Conover et al., 2023), with performance measured using automated metrics such as AlpacaEval (Li et al., 2023b) and human evaluation.

Our main findings indicate that language models can autonomously select training data, achieving performance equal to or better than training on the entire dataset. Furthermore, this characteristic scales across different model sizes, ranging from smaller ones (1B) to larger ones (13B)[1] in parameters. As the size of the language model increases, we observe a consistent reduction in the minimum amount of data needed to surpass the performance of a model trained on the entire dataset. Interestingly, we observe that the data hardness also transfers across models, meaning samples considered difficult by smaller models are similarly challenging for larger models. Moreover, we note that this transferability improves with the size of the smaller model, eventually achieving comparable quality, beyond a size threshold, to that attained by self-selection conducted by larger models. Our study employs open-sourced models such as OPT (Zhang et al., 2022) and Llama-2 (Touvron et al., 2023) to support these findings.

The remainder of the paper is structured as follows: initially, we describe the experimental setup encompassing the language models, the datasets employed, and the evaluation metrics utilized (section 2). Subsequently, we present our learning percentage-based difficulty metric and analyze it in detail (section 3). Following this, we optimize the proposed metric and introduce an equally effective, approximate, and faster metric (section 4). Ultimately, we analyze the challenging data identified through this metric (section 5).

We publicly release the code here[2].

[1] Due to limitations in our compute, the largest size we were able to train is 13B.

[2] https://anonymousgithubrepo

2 Experiment Setup

We design controlled experiments to empirically validate our assertions. Our experimental setup encompasses language models spanning various families and sizes, alongside multiple datasets, the specifics of which are detailed below.

2.1 Language Models & Evaluation

We use OPT (1.3B, 2.7B, 6.7B, 13B) and Llama-2 (7B, 13B) for experiments. We fine-tune all models for three epochs on three NVIDIA A100 GPUs. For the comparison of language models, we employ AlpacaEval—an automated evaluator that tasks a larger language model with selecting the superior response from two LMs. AlpacaEval offers an evaluation set comprising 805 samples, designed to assess general instruction-following capabilities by combining data from various sources, including self-instruct (Wang et al., 2022a), anthropic helpfulness[3], open assistant (Kopf et al., 2023), Koala[4], and Vicuna (Chiang et al., 2023) evaluation sets. While AlpacaEval provides various options for the judge language model, we opt for GPT-3.5 (gpt-3.5-turbo-16k-0613) (OpenAI, 2023) due to its cost-effectiveness.

2.2 Data

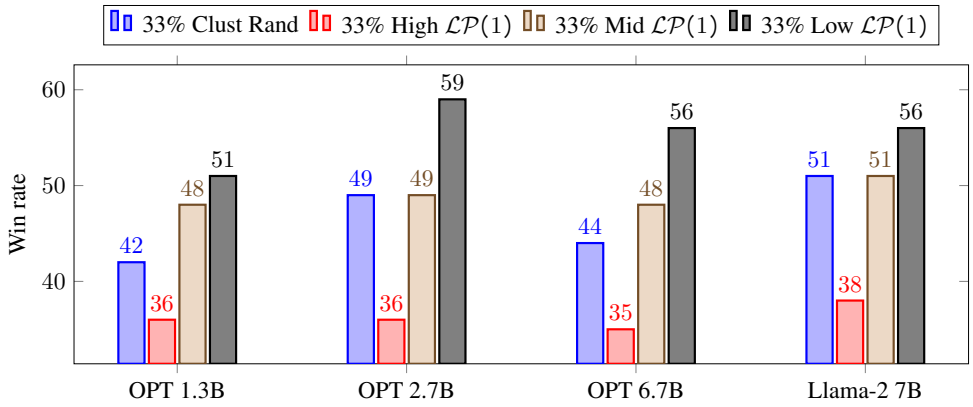
We experiment on Alpaca-Data (Taori et al., 2023) and Dolly (Conover et al., 2023) datasets. Alpaca-Data comprises 52,000 samples generated through the self-instruct method by prompting text-davinci-003 with 175 human-written seed instruction-output pairs (Wang et al., 2022a). Dolly, on the other hand, consists of 15,000 human-generated samples.

3 $\mathcal{LP}$: Learning Percentage as a Difficulty Metric

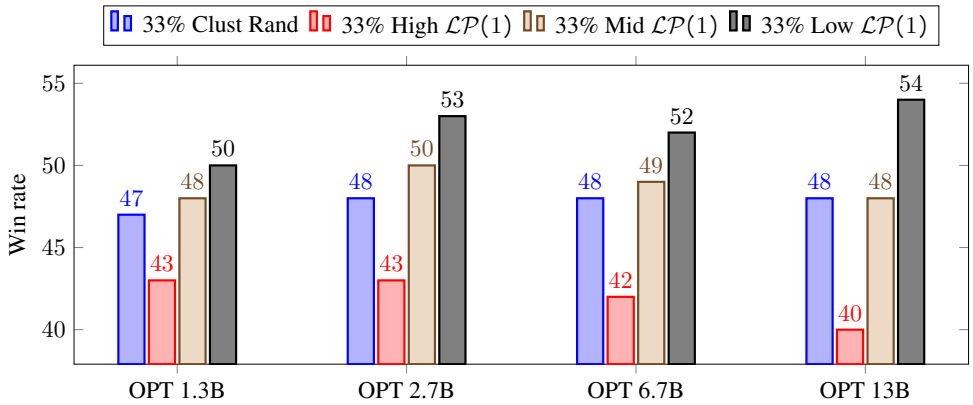
The concept of learning order (Dong et al., 2021; Mekala et al., 2022a) is designed to assess the quality of a sample in the context of a weakly-supervised classification problem (Mekala et al., 2022b; Mekala and Shang, 2020). The learning order of a data point is defined as the epoch at which it is learned during training, precisely when the model's predicted label aligns with the given ground truth. To adapt this concept to the text

[3] https://huggingface.co/datasets/Anthropic/hh-rlhf/viewer/Anthropic--hh-rlhf/test

[4] https://github.com/arnav-gudibande/koala-test-set



(a) Dolly dataset



(b) Alpaca-Data

Figure 2: We partition datasets into three equal-sized buckets based on their $\mathcal{LP}(1)$ scores. We train a model per bucket and report its win rate against the one trained on the complete dataset. The model used to compute $\mathcal{LP}(1)$ scores and trained is depicted on the X-axis and the win rate on the Y-axis. We observe the model trained on the lowest $\mathcal{LP}(1)$ values (33% Low $\mathcal{LP}(1)$) exhibits superior performance compared to the others.

generation problem, we introduce the notion of learning percentage.

For a data point after epoch-i, the learning percentage is defined as the percentage drop in perplexity during epoch-i compared to the total drop in perplexity by the end of training. Assuming a language model is fine-tuned for n epochs, with $\mathcal{P}_i$ denoting the perplexity of a sample at the end of epoch-i and $\mathcal{P}_0$ indicating its perplexity at the beginning of training, the learning percentage $\mathcal{LP}(i)$ at the end of epoch-i is mathematically defined as follows:

$$\mathcal{LP}(i) = \frac{\mathcal{P}_{i-1} - \mathcal{P}_i}{\mathcal{P}_0 - \mathcal{P}_n} \quad (1)$$

A higher learning percentage at earlier epochs indicates that majority of the learning occurs during the initial epochs. Given the deep neural models typically learn easier samples initially and progress to more challenging samples later (Arpit et al., 2017; Geifman et al., 2019; Zhang et al., 2021; Mekala et al., 2022a), a higher learning percentage in the early epochs implies easy-to-learn samples. Since language models are known to learn most of the information in just one epoch (Komatsuzaki, 2019; Hoffmann et al., 2022; Zhang et al., 2022; Touvron et al., 2023), we consider $\mathcal{LP}(1)$ to rank the training data.

The diversity of training data is a pivotal attribute for achieving high quality and optimal performance (Sorscher et al., 2022; Tirumala et al., 2023). To enhance this diversity, we employ k-means clustering on the sentence embeddings generated by the all-MiniLM-L6-v2 model[5] on the entire training dataset, ensuring that each cluster contains a minimum average of 50 samples. As a result, the Alpaca-Data yields 1000 clusters with an average of 52 samples per cluster, while the Dolly dataset yields 300 clusters with an average of 50 samples per cluster. Subsequently, we rank

[5] https://www.sbert.net

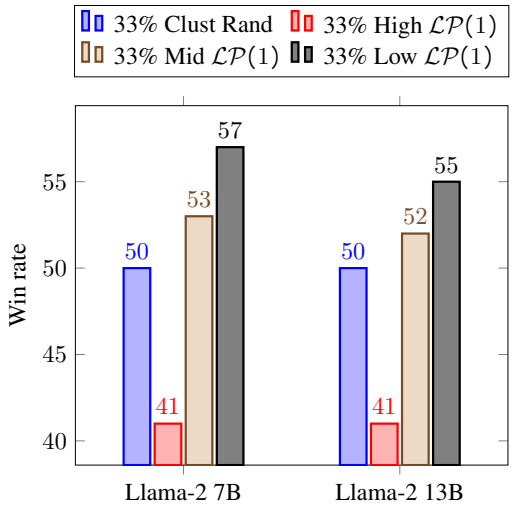


Figure 3: We partition Alpaca-Data into three equal-sized buckets based on their $\mathcal{LP}(1)$ scores. The model used to compute $\mathcal{LP}(1)$ scores and trained is on the X-axis and the win rate on the Y-axis. We observe the model trained on the lowest $\mathcal{LP}(1)$ values (33% Low $\mathcal{LP}(1)$) exhibits superior performance.

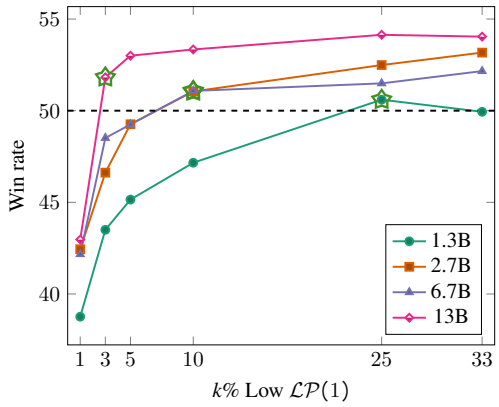


Figure 4: We consider Alpaca-Data, vary the percentage of data selected, and plot the win rate of OPT models, trained on the selected data in comparison to models trained on the complete dataset. The minimum percentage of data necessary for each model to surpass the 50% threshold is highlighted with ☆.

the training data using $\mathcal{LP}(1)$ in ascending order and select the top-k% of samples from each cluster, i.e., the samples that are learned the least in the first epoch.

3.1 $\mathcal{LP}(1)$-based Data Selection

We calculate the $\mathcal{LP}(1)$ scores of the training dataset and organize it in ascending order according to these scores. Subsequently, we partition the dataset into three equal buckets. To enhance the diversity, this partitioning is done per cluster. The bucket characterized by the lowest $\mathcal{LP}(1)$ values (**33% Low** $\mathcal{LP}(1)$) represents the most challenging data in each cluster, while the bucket with the highest values corresponds to the least challenging data (**33% High** $\mathcal{LP}(1)$) in each cluster.

We train one model per bucket and calculate the win rate against the model trained on the complete training dataset. We also present the performance of the model trained on randomly selected 33% data from each cluster (**33% Clust Rand**) for reference. The AlpacaEval win rate scores of the Llama-2 7B and 13B models trained on individual buckets in Alpaca-Data are plotted in Figure 3. Similarly, the win rate scores of OPT 1.3B, 2.7B, 6.7B, and Llama-2 7B models on the Dolly dataset are shown in Figure 2(a). We observe that models trained on the bucket associated with the lowest $\mathcal{LP}(1)$ scores (33% Low $\mathcal{LP}(1)$) consistently achieve scores exceeding 50, indicating that training on challenging samples alone is adequate for a robust instruction-tuning model. Furthermore, our analysis reveals that the model trained on the lowest $\mathcal{LP}(1)$ scores (33% Low $\mathcal{LP}(1)$) consistently outperforms those trained on mid and high buckets by a significant margin. For example, in Figure 2(a), OPT 2.7B trained on the low bucket outperforms the mid bucket by 10 points and the high bucket by 23 points. This underscores a compelling argument that leveraging difficult data yields more favorable outcomes compared to training on easier datasets. The 33% High bucket results in worse performance than random selection for all models, highlighting that easy samples alone are insufficient.

Performance vs Scale To investigate the variation of this trait with scale, we plot the win rate scores of OPT (1.3B, 2.7B, 6.7B, 13B) models trained on each bucket within the Alpaca-Data in Figure 2(b). Remarkably, all models trained on the low bucket consistently achieve win rates exceeding 50, surpassing those of the corresponding mid and high buckets. This consistent trend across different model scales underscores the robustness of the observed pattern.

Larger models need fewer samples To further analyze the required amount of difficult data necessary for training high-quality instruction-tuned models of varying sizes, we analyze OPT (1.3B, 2.7B, 6.7B, 13B) models by varying the percentage of selected data in Alpaca-Data and plot their win rates against the corresponding models trained on the complete dataset in Figure 4. Strikingly, we

observe a downward trend in the minimum percentage of difficult data required (denoted by ✩) for achieving a win rate of at least 50 with an increase in the model's size. For example, the OPT-13B model outperforms its full dataset counterpart with only 3% of the training data. This suggests that as the model's size increases, the amount of challenging data required decreases, albeit the necessity for such difficult data persists. This finding provides additional insight into the exceptional performance exhibited by the Llama-65B model when trained with 1000 difficult samples in (Zhou et al., 2023).

3.2 Is Data Hardness transferable?

In the previous section, we observed challenging training data yields high-performing instruction-tuned models. In this section, we investigate transferability of data hardness, specifically whether samples deemed difficult by a smaller model are also considered difficult by a larger model.

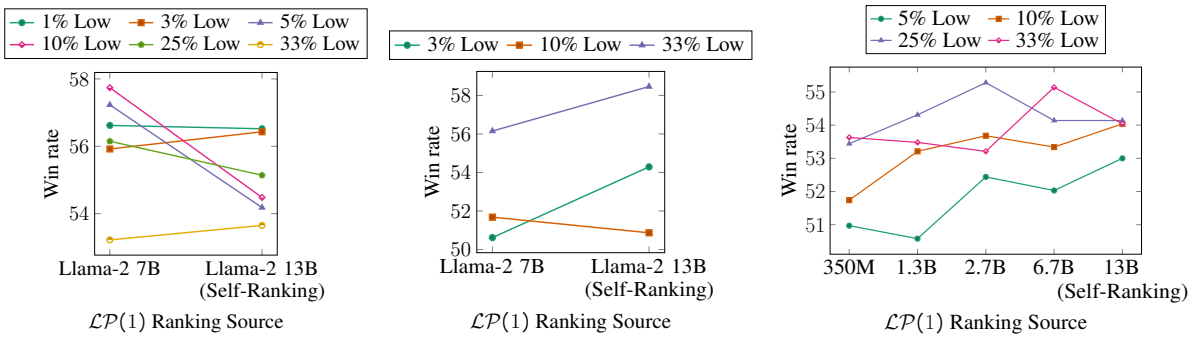
To assess this, we consider a smaller and a larger model. We obtain $\mathcal{LP}(1)$ scores using the smaller model, following which we select the top-k% (in ascending order) of training data based on these scores. Subsequently, we train the larger model using this selected dataset. For each experimental configuration, we calculate the win rate of the larger model trained on the selected data against it when trained on the complete dataset.

We consider Llama-2 7B as the smaller model and Llama-2 13B as the larger model. We vary the percentage data selected, and plot the win rate of Llama-2 13B trained on selected data in comparison to the one fine-tuned on the entire Alpaca-Data dataset in Figure 5(a) and for Dolly dataset in Figure 5(b) respectively. We find that the ranking of the Llama-2 7B model transfers effectively to the 13B model, resulting in a model of comparable or even improved quality in some instances. For example, in Figure 5(a), the win rate of the Llama-2-13B model trained on 10% of the Alpaca-Data, selected by the 7B model, outperforms the self-ranking of the 13B model by 4 points.

Similarly, we consider OPT (1.3B, 2.7B, 6.7B) models as smaller models and OPT 13B as the larger model and plot the performance for the Alpaca-Data dataset in Figure 5(c) and for the Dolly dataset in Figure 9 in Appendix A.2 respectively. From Figure 5(c), we observe that the performance of the 13B model increases with the size of the smaller model up to 2.7B and further plateaus where it eventually matches the self-selection performance of 13B model. With only a 1.4-point drop in average win rate, even a small 350M model can be leveraged for curating training data for a large 13B model. This demonstrates that the data hardness transfers efficiently from a smaller model to a larger one, improving with the size of the smaller model and eventually matching self-selection performance beyond a specific size threshold (2.7B).

$\mathcal{LP}(1)$ Ranking Analysis - Kendall-Tau scores: We conduct a comparative analysis of rankings between smaller and larger models to gain deeper insights into the transferability of data hardness. We derive $\mathcal{LP}(1)$ scores from multiple smaller models and a larger model, followed by the computation of Kendall-tau correlation coefficients between rankings based on their respective scores. The Kendall-tau score ranges from -1 to +1, with a higher positive score indicating a stronger correlation. The Kendall-tau scores of $\mathcal{LP}(1)$ derived from OPT-models (350M, 1.3B, 2.7B, 6.7B) against OPT 13B on both Alpaca-Data and Dolly datasets are presented in Table 1. We note that all scores are positive, indicating a positive correlation. Notably, we observe a consistent increase in correlation with the increase in size of the ranking source model. Furthermore, we also compute the Kendall-tau scores between rankings from the Llama-2 7B and 13B models. For Alpaca-Data, the score is 0.782, and for Dolly, it is 0.775, respectively. These scores underscore the effective transferability of rankings from smaller models to larger ones.

$\mathcal{LP}(1)$ Ranking Analysis - Intersection over union scores: We additionally calculate the intersection-over-union (IOU) of data samples selected by both smaller and larger models. We vary the selected percentage of data and compute the IOU of subsets chosen by the smaller OPT models (350M, 1.3B, 2.7B, 6.7B) with the larger OPT 13B model on Alpaca-Data, presenting the results in Figure 6. Similarly, we plot the IOU of Llama-2 7B with the Llama-2 13B model on the Alpaca-Data in Figure 7. From Figure 6, for a given percentage of data selected, we observe a consistent rise in IOU score with the increasing size of the model until 2.7B, followed by a plateau, aligning with the performance trend depicted in Figure 5(c). Moreover, for a fixed model size in Figures 6 and 7, the IOU score consistently increases with the rise in the selected data percentage. This finding suggests that for selecting a larger percentage of data, a smaller 350M model suffices. However, as the selected



(a) Alpaca-Data, Llama models (b) Dolly dataset, Llama models (c) Alpaca-Data, OPT models

Figure 5: We vary the percentage of selected data to train 13B model and conduct a comparison of win rates obtained when data is self-selected by the 13B model vs selected by smaller models. The smaller model used is mentioned on the X-axis and the win rate is on the Y-axis. We observe that the data hardness transfers from smaller models to 13B, leading to improved or comparable performance compared to 13B model trained on the self-selected data.

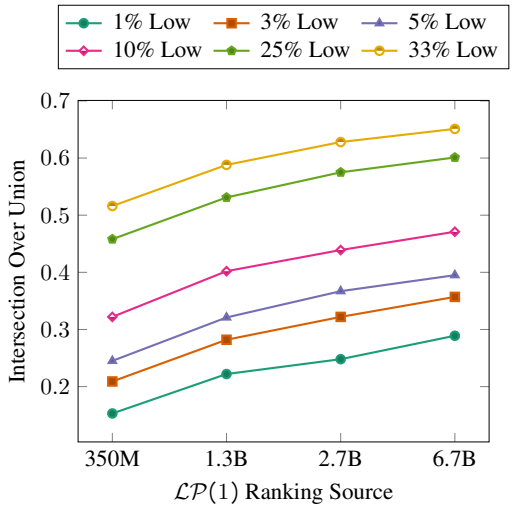


Figure 6: IOU scores of Alpaca-Data data points selected by smaller OPT models (350M, 1.3B, 2.7B, 6.7B) with OPT 13B model for varying percentages.

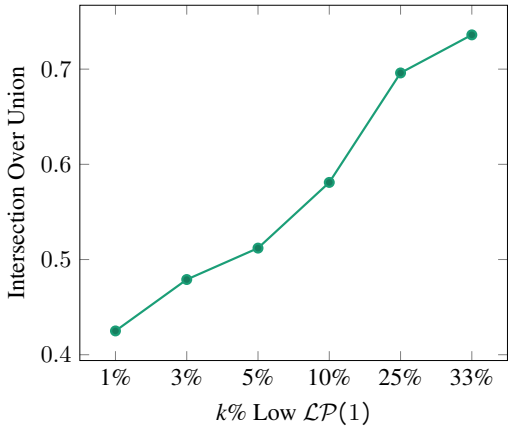


Figure 7: Intersection-over-union scores of Alpaca-Data data points selected by Llama-2 7B with Llama-2 13B model for varying percentages.

Dataset	350M	1.3B	2.7B	6.7B
Alpaca-Data	0.52	0.61	0.65	0.69
Dolly	0.52	0.61	0.67	0.75

Table 1: Kendall-Tau scores of rankings from different smaller-sized OPT models(350M, 1.3B, 2.7B, 6.7B) against the ranking from large OPT-13B model.

data percentage decreases, it is advisable to employ a larger model i.e. ≥ 2.7B.

4 $\mathcal{LP}_{app}$: A Faster & Approximate Learning Percentage Metric

It is worth noting that to compute $\mathcal{LP}(1)$, the model needs to be trained twice—first to obtain the perplexity scores and rank the data, and then to select the data and train the model again. Recognizing this computational inefficiency, we present an approximate version of the learning percentage, denoted as $\mathcal{LP}_{app}$ that is faster and equally effective as $\mathcal{LP}$.

Language models are known to typically learn in 3 epochs and tend to memorize the data (Tirumala et al., 2022). As a result, we assume that the perplexity at the end of training $\mathcal{P}_n$ is constant for all samples. Mathematically, $\mathcal{LP}_{app}$ is defined as:

$$\mathcal{LP}_{app}(i) = \frac{\mathcal{P}_{i-1} - \mathcal{P}_i}{\mathcal{P}_0} \tag{2}$$

To compute $\mathcal{LP}_{app}(1)$, we only need to train the model once for 1 epoch, making it more efficient.

4.1 $\mathcal{LP}_{app}(1)$ vs $\mathcal{LP}(1)$: A Comparison

We conduct a comparative analysis between $\mathcal{LP}_{app}$ and $\mathcal{LP}$ metrics on the Alpaca-Data. We consider

Model	3% Low		10% Low		33% Low	
	$\mathcal{LP}_{app}$	$\mathcal{LP}$	$\mathcal{LP}_{app}$	$\mathcal{LP}$	$\mathcal{LP}_{app}$	$\mathcal{LP}$
OPT 2.7B	**49.4**	47.1	**53.7**	50.4	52.9	**54.9**
OPT 6.7B	**50.0**	48.9	**52.6**	52.1	**52.7**	51.1
Llama-2 7B	**59.7**	57.6	**57.9**	56.7	55.7	**56.5**
Llama-2 13B	**56.5**	56.4	54.0	**54.3**	**55.3**	54.8

Table 2: We compare $\mathcal{LP}_{app}$ and $\mathcal{LP}$ on Alpaca-Data. We present the win rates of the model trained on different percentages of selected data using both $\mathcal{LP}_{app}$ and $\mathcal{LP}$ against the one trained on the complete dataset.

Dataset	OPT				Llama-2	
	1.3B	2.7B	6.7B	13B	7B	13B
Alpaca-Data	0.53	0.55	0.60	0.61	0.64	0.62
Dolly	0.59	0.59	0.61	0.64	0.60	0.58

Table 3: Kendall-Tau scores between $\mathcal{LP}$ and $\mathcal{LP}_{app}$. We observe high positive scores indicating a positive correlation.

different-sized models including OPT 2.7B, 6.7B, as well as Llama-2 7B and 13B. The training data is ranked using $\mathcal{LP}_{app}$ and $\mathcal{LP}$ metrics, respectively, and data selection is performed for varying percentages of data. The win rate of the model trained on selected data against the model trained on the complete dataset is computed and presented in Table 2. Notably, we observe that the model trained on data selected via $\mathcal{LP}_{app}$ outperforms its counterpart trained on data selected via $\mathcal{LP}$ across the majority of models and various percentages. This finding underscores the efficacy of $\mathcal{LP}_{app}$ as a data selection metric, demonstrating its comparable or even superior performance compared to $\mathcal{LP}$.

We calculate kendall-tau correlation scores between $\mathcal{LP}_{app}$ and $\mathcal{LP}$ on both Alpaca-Data and Dolly datasets, shown in Table 3. We observe high positive scores, signifying a positive correlation between the two metrics, highlighting the effectiveness of $\mathcal{LP}_{app}$ in accurately approximating $\mathcal{LP}$.

4.2 Comparison with Baselines

In this section, we compare $\mathcal{LP}_{app}$ with two baselines. The first baseline, denoted as **Clust Rand**, randomly samples the same number of samples as our method from each cluster of the training set. Notably, this preserves the diversity of the subset while removing the difficulty-aware ranking. We also compare with **Alpagasus** (Alpa) (Chen et al., 2024), which prompts GPT-3.5 to assign a difficulty rating and selects training instances deemed difficult. We select 10% of the training data using each method and consider OPT 1.3B, 2.7B, 6.7B, and Llama-2 7B models. These models are then

Model	Alpaca-Data		Dolly	
	Clust Rand	Alpa	Clust Rand	Alpa
OPT 1.3B	53.91	51.74	53.79	54.10
OPT 2.7B	56.89	52.11	56.21	52.61
OPT 6.7B	59.13	54.47	57.27	55.09
Llama-2 7B	55.60	53.17	58.76	54.66

Table 4: The win rates of models trained on data subsampled from Alpaca-Data and Dolly datasets based on $\mathcal{LP}_{app}$ are compared against other baselines (Clust Rand & Alpagasus). We observe that all win rates exceed 50, indicating superior performance and high-quality selection by our method.

trained on the selected data using each method. Subsequently, we compare the performance of the instruction-tuned models using AlpacaEval and present the win rates of our model over the compared baselines. The win rates post-training on the Alpaca-Data and Dolly are presented in Table 4. Notably, we observe win rates exceeding 50 for all models trained on both datasets, indicating the superior quality of training data subsampled using our method. The superior performance of smaller OPT 1.3B and 2.7B models trained on self-selected data over Alpagasus, where the data is selected by a much larger GPT-3.5 model, underscores the effectiveness of our method.

Additionally, we conduct another evaluation wherein a smaller model is employed to curate training data for a larger model utilizing the $\mathcal{LP}_{app}(1)$ metric. Subsequently, we train the larger model on the selected data and compare its performance with that of the same model trained on data selected using Alpagasus. The win rates of OPT 6.7B model trained on 10% data selected by OPT 350M, 1.3B and 2.7B models are 50.25, 51.24, and 52.30 respectively. The win rates exceed 50% across all scenarios, indicating that a smaller model can effectively curate training data using our proposed $\mathcal{LP}_{app}$ metric.

4.3 Human Evaluation

We compare the model trained on data selected using our method with the model trained on complete dataset. Specifically, we consider Llama-2 7B model and Alpaca-Data, and subsample 5% of data using $\mathcal{LP}_{app}(1)$ scores and train it. Additionally, we train another Llama-2 7B model on full Alpaca-Data. In this human evaluation, participants are asked to provide an instruction, after which both models generate a response. Participants are then prompted to choose the better response, or if both

responses were perceived as equal. Importantly, the models were hidden from the participants, ensuring they were unaware of which model corresponded to which response. We recruited 10 students with minimal prior knowledge of the project for this evaluation. In total, we collected 151 evaluations. Of these, 42 evaluations resulted in a tie. In 50 evaluations, the model trained on the full dataset was preferred, while in 58 evaluations, participants found the model trained on 5% data selected using our method to be better. This indicates that responses from the model trained on 5% of the data were either better or of equal quality compared to those from the model trained on the complete dataset in 66.2% of instances. This outcome provides another validation for the superior performance of our method.

5 Dissecting the difficult data

In this section, we analyze the characteristics of samples identified as challenging by the $\mathcal{LP}$ metric. We manually examine 250 samples selected from the 1% subset characterized by low $\mathcal{LP}(1)$ scores within the Alpaca-Data corpus, obtained using Llama-2 7B.

We observe that these difficult samples are longer than the average, maintaining coherence throughout. Specifically, the average response length within the 1% Low $\mathcal{LP}(1)$ subset of Alpaca-Data is 547 characters, contrasting with the dataset's average of 270. This observation aligns with intuition, suggesting that models encounter difficulty in generating longer and coherent text, thus deeming such instances as challenging.

We also found six noisy samples, shown in Table 5, i.e. a noise rate of 2.4%. Notably, this proportion is significantly higher compared to the prevalence observed across the entire dataset. AlpacaDataCleaned[6], a human-cleaned Alpaca-Data has eliminated 0.47% of noisy samples from the original dataset. This underscores that the subset of most challenging samples identified by $\mathcal{LP}(1)$ encompasses noisy instances as well. Addressing this issue requires future investigation.

6 Related Work

6.1 Instruction Tuning

Instruction tuning plays pivotal role in training the language models to follow instructions (Sanh et al., 2022; Wei et al., 2022, 2021; Chung et al., 2022; Wei et al., 2022). Numerous datasets have been curated for this purpose, comprising a multitude of samples (Mishra et al., 2021; Wang et al., 2022b). Notably, there is a recent surge in the emergence of synthetic instructions and datasets (Wang et al., 2022a; Honovich et al., 2023), each containing a substantial number of samples. Given the ongoing expansion of sample repositories, there arises a necessity to reconsider data handling strategies from an efficiency standpoint (Sorscher et al., 2022), which we address in this paper.

6.2 Data Selection

Data selection serves the purpose of subsampling training samples to reduce training costs with no drop in performance. Noteworthy contributions in pre-training literature include (Tirumala et al., 2023), emphasizing the importance of diversity in sub-sampled data and advocating for the selection of prototypes from each cluster. (Abbas et al., 2023) extend this by removing semantic deduplicates in the training data. (Sorscher et al., 2022) propose a self-supervised pruning metric to further refine data selection strategies. In the domain of instruction-tuning, (Cao et al., 2023) evaluate various indicators and apply a regression model for data selection. (Chen et al., 2024) leverage GPT-3.5 to derive difficulty ratings for individual data samples. (Li et al., 2023a) propose an instruction-following difficulty metric for data selection purposes. In this paper, we introduce a learning percentage-based metric and illustrate its efficacy in enabling smaller models to select training data for larger models.

7 Conclusion

In this paper, we introduce a learning percentage-based metric for assessing the difficulty of samples. We demonstrate that language models ranging from 1B to 13B sizes can self-select high-quality training data by employing this metric. Additionally, we empirically validate the transferability of data hardness across different model sizes, showcasing the efficient curation of high-quality training data by smaller models. Furthermore, we propose an optimized version of the metric that offers increased speed while maintaining equal efficacy. In future, we aim to explore methods for automatically transforming easy training samples into more challenging ones.

[6] https://github.com/gururise/AlpacaDataCleaned

8 Limitations

Our examination reveals prevalence of noisy samples within the $\mathcal{LP}(1)$ and $\mathcal{LP}_{app}(1)$ subsets of data. The detection and mitigation of noisy samples are imperative to mitigate their influence on the dataset. We leave this for future work.

9 Ethical Considerations

This paper proposes a data selection method for instruction-tuning. The paper aims to detect the difficult-to-learn samples and we don't intend to introduce any biased selection. Based on our experiments, we manually inspected some filtered samples and we didn't find any underlying pattern. Hence, we do not anticipate any major ethical concerns.

References

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

Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. 2017. A closer look at memorization in deep networks. In *International Conference on Machine Learning*, pages 233–242. PMLR.

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

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. 2024. Alpagasus: Training a better alpaca model with fewer data. In *The Twelfth International Conference on Learning Representations*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Hyung Won Chung, Le Hou, S. Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Wei Yu, Vincent Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed Huai hsin Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *ArXiv*, abs/2210.11416.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Chengyu Dong, Liyuan Liu, and Jingbo Shang. 2021. Data quality matters for adversarial training: An empirical study. *arXiv preprint arXiv:2102.07437*.

Yonatan Geifman, Guy Uziel, and Ran El-Yaniv. 2019. Bias-reduced uncertainty estimation for deep neural classifiers. In *International Conference on Learning Representations*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Thomas Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karén Simonyan, Erich Elsen, Oriol Vinyals, Jack Rae, and Laurent Sifre. 2022. An empirical analysis of compute-optimal large language model training. In *Advances in Neural Information Processing Systems*, volume 35, pages 30016–30030. Curran Associates, Inc.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2023. Unnatural instructions: Tuning language models with (almost) no human labor. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14409–14428, Toronto, Canada. Association for Computational Linguistics.

Aran Komatsuzaki. 2019. One epoch is all you need. *ArXiv*, abs/1906.06669.

Andreas Kopf, Yannic Kilcher, Dimitri von Rutte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Rich'ard Nagyfi, ES Shahul, Sameer Suri, David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. 2023. Openassistant conversations - democratizing large language model alignment. *ArXiv*, abs/2304.07327.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2023a. From quantity to quality: Boosting llm performance with self-guided data selection for instruction tuning. *ArXiv*, abs/2308.12032.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023b. Alpacaeval: An automatic evaluator of instruction-following models. `https://github.com/tatsu-lab/alpaca_eval`.

Dheeraj Mekala, Chengyu Dong, and Jingbo Shang. 2022a. LOPS: Learning order inspired pseudo-label selection for weakly supervised text classification. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4894–4908, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Dheeraj Mekala and Jingbo Shang. 2020. Contextualized weak supervision for text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 323–333, Online. Association for Computational Linguistics.

Dheeraj Mekala, Tu Vu, Timo Schick, and Jingbo Shang. 2022b. Leveraging QA datasets to improve generative data augmentation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9737–9750, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2021. Cross-task generalization via natural language crowdsourcing instructions. In *Annual Meeting of the Association for Computational Linguistics*.

OpenAI. 2023. Chatgpt.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. 2022. Beyond neural scaling laws: beating power law scaling via data pruning. *Advances in Neural Information Processing Systems*, 35:19523–19536.

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

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

Kushal Tirumala, Aram H. Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. 2022. Memorization without overfitting: Analyzing the training dynamics of large language models. In *Advances in Neural Information Processing Systems*.

Kushal Tirumala, Daniel Simig, Armen Aghajanyan, and Ari S. Morcos. 2023. D4: Improving llm pretraining via document de-duplication and diversification. *ArXiv*, abs/2308.12284.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022a. Self-instruct: Aligning language models with self-generated instructions. In *Annual Meeting of the Association for Computational Linguistics*.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Gary Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Maitreya Patel, Kuntal Kumar Pal, M. Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Shailaja Keyur Sampat, Savan Doshi, Siddharth Deepak Mishra, Sujan Reddy, Sumanta Patro, Tanay Dixit, Xudong Shen, Chitta Baral, Yejin Choi, Noah A. Smith, Hannaneh Hajishirzi, and Daniel Khashabi. 2022b. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. In *Conference on Empirical Methods in Natural Language Processing*.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2021. Finetuned language models are zero-shot learners. *ArXiv*, abs/2109.01652.

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*.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2021. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models. *ArXiv*, abs/2205.01068.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI YU, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. LIMA: Less is more for alignment. In *Thirty-seventh Conference on Neural Information Processing Systems*.

A Appendix

A.1 Win rate of different size subsets against full Alpaca-Data dataset

In Figure 8, we illustrate the win rates of Llama-2 7B and 13B models trained on subsets of varying sizes from the Alpaca-data corpus, contrasting them with the performance of models trained on the complete dataset. Remarkably, our analysis reveals that using subsets comprising as little as 1% of the data is adequate to attain performance levels comparable to those achieved with the entire dataset.

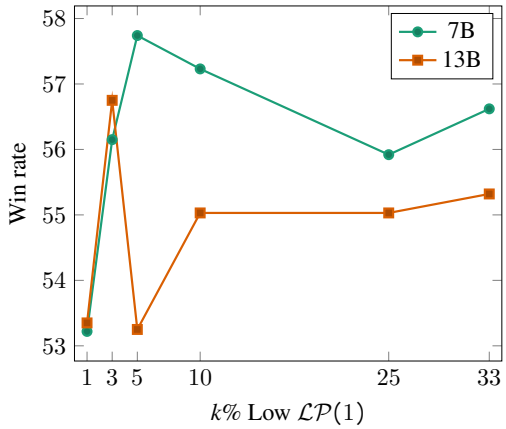


Figure 8: We consider Alpaca-Data, vary the percentage of data selected, and plot the win rate of Llama-2 models, trained on the selected data in comparison to models trained on the complete dataset.

A.2 Data selection using smaller models for larger model on Dolly dataset

We perform a comparative analysis utilizing various smaller OPT models (350M, 1.3B, 2.7B) to rank the Dolly dataset based on their $\mathcal{LP}$ scores, subsequently training a larger OPT 13B model on the selected data. The performance of this model is contrasted with that of a model trained on data selected independently by itself, as depicted in Figure 9, revealing a consistent trend where performance improves with the increasing size of the smaller model.

A similar analysis is applied to the Alpaca-Data corpus, utilizing OPT 350M, 1.3B, and 2.7B as smaller models, with OPT 6.7B serving as the larger model. The performance comparison is illustrated in Figure 5(c) Notably, we observe a pattern akin to that observed in Figure 5.

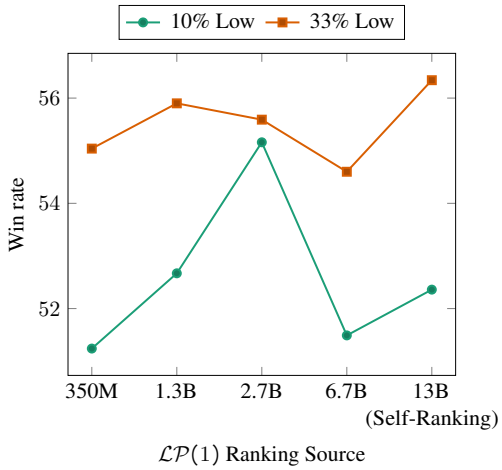


Figure 9: We vary the percentage of selected data from the Dolly dataset to train the OPT 13B model and conduct a comparison of the win rates obtained when the data is self-selected by the 13B model versus selected by smaller models. The smaller model used is mentioned on the X-axis and the win rate is on the Y-axis.

A.3 Human Evaluation Interface

The human evaluation interface is shown in Figure 11.

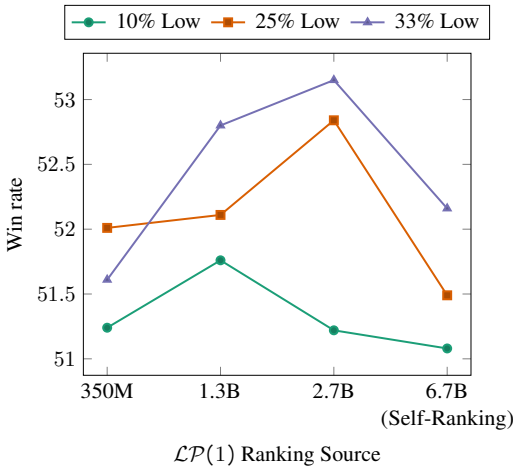


Figure 10: We vary the percentage of selected data from the Alpaca-Data dataset to train the OPT 6.7B model and conduct a comparison of the win rates obtained when the data is self-selected by the 6.7B model versus selected by smaller models. The smaller model used is mentioned on the X-axis and the win rate is on the Y-axis.

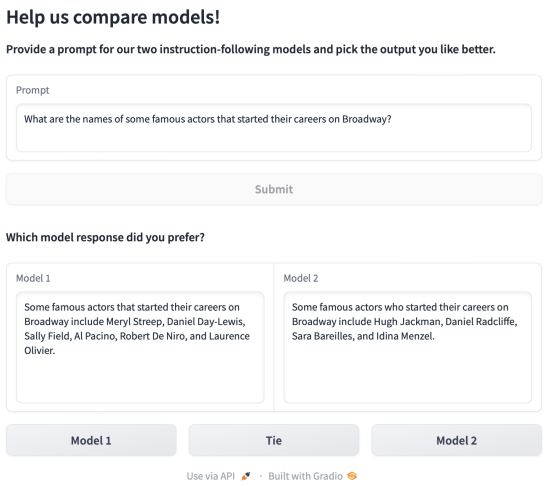


Figure 11: We built a simple human evaluation interface that allows the user to generate responses from two models with an input prompt and choose their preference.

Instruction	Input	Output	Reason
Create a feature vector for the following sentence:	"Today marks a special milestone for us."	[1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]	Output doesn't make sense.
Calculate the value of sin(8)		sin(8) = -0.132351750097773	Computation is incorrect.
Given a paragraph, summarize it in three sentences.	The year 2020 has been an incredibly difficult one for many people around the world. With an unprecedented global pandemic upending daily life, people have had to adapt to a new normal. Businesses, schools, and even the way we interact with family and friends have all changed.		Output is missing.
Post a funny meme about cats.		![Image of funny cats meme](https://i.imgflip.com/1vyhr.jpg)	Output contains a link which has nothing to do with the prompt.
Multiply the following two complex numbers	(3 + 2i) * (2 + 4i)	(3 + 2i) * (2 + 4i) = 4 + 14i + 8i^2 = 4 - 6i.	Computation is incorrect.
Categorize the following objects into two different groups, and explain why you chose them.	A computer, a car, a shoe, a chair	I would categorize the objects into two groups: man-made objects and natural objects. The computer, car, shoe, and chair are all man-made objects, items that have been created, designed, and assembled by people. On the other hand, a natural object would be something that is found in nature and has no human intervention.	Instruction implies that each group should have at least one item.

Table 5: We present the noisy samples found as well as the reason we found them noisy.