A Framework for Fine-Tuning LLMs using Heterogeneous Feedback

Ryan Aponte*, Ryan A. Rossi+, Shunan Guo+, Franck Dernoncourt+, Tong Yu+, Xiang Chen+, Subrata Mitra+, Nedim Lipka+
*Carnegie Mellon University, +Adobe Research
raponte@cs.cmu.edu, {ryrossi,sguo,dernonco,tyu,xiangche,sumitra,lipka}@adobe.com

Abstract

Large language models (LLMs) have been applied to a wide range of tasks, including text summarization, web navigation, and chatbots. They have benefitted from supervised fine-tuning (SFT) and reinforcement learning from human feedback (RLHF) following an unsupervised pretraining. These datasets can be difficult to collect, limited in scope, and vary in sample quality. Additionally, datasets can vary extensively in supervision format, from numerical to binary as well as multi-dimensional with many different values. We present a framework for fine-tuning LLMs using heterogeneous feedback, which has two main components. First, we combine the heterogeneous feedback data into a single supervision format, compatible with methods like SFT and RLHF. Next, given this unified feedback dataset, we extract a high-quality and diverse subset to obtain performance increases potentially exceeding the full dataset. We conduct extensive experiments to understand the effectiveness of these techniques for incorporating heterogeneous feedback, and demonstrate improvements from using a high-quality and diverse subset of the data. We find that our framework is able to improve models in multiple areas simultaneously, such as in instruction following and bias reduction.[1].

1 Introduction

LLMs are fine-tuned for a variety of purposes, such as for instruction following in InstructGPT (Ouyang et al., 2022). The fine-tuning process generally begins with collecting examples of desired model behavior and performing supervised learning. Some models stop at SFT (Chiang et al., 2023), while InstructGPT follows this by training a reward model based on binary human preference data. The fine-tuned model is then further refined using RLHF, using a signal from the reward model. In the example of InstructGPT, the algorithm used is Proximal Policy Optimization (PPO) (Schulman et al., 2017). Fine-tuning datasets exist for a variety of purposes, from training chat-based assistants in OASST (Köpf et al., 2023), coreference resolution in WinoGrande (Sakaguchi et al., 2019), helpfulness, honesty, and harmlessness in Anthropic HHH (Nakano et al., 2021), and logical reasoning in OpenPlatypus (Lee et al., 2024). Supervision format varies, from binary preference in Anthropic HHH, to several numerical labels OASST, to a string response in OpenPlatypus. Although fine-tuning has been successful in mitigating the limitations of pretrained LLMs, these methods require data of a single supervision type, restricting the scope of preference data. Recent work has filtered fine-tuning datasets to reduce cost and increase quality (Wang et al., 2024). (Wu et al., 2023) use LLMs to generate embeddings for fine-tuning data which is clustered with k-center-greedy (Sener and Savarese, 2018). (Kung et al., 2023) randomly delete words in prompts and measure how the response probability changes as a measure of the model's uncertainty. (Li et al., 2024) outperform Alpaca as evaluated by LLM preference using only 5% of its fine-tuning data. We present a framework to use multiple fine-tuning data types, permitting the use of more fine-tuning datasets and fine-tuning for multiple tasks simultaneously. This provides a more accurate view of human preference by broadening the scope of fine-tuning data. Our framework selects a high-quality and diverse subset of the data to make fine-tuning more effective.

2 Framework

The primary contribution of our framework is to be able to use fine-tuning data of heterogeneous supervision. Figure 1 includes a high-level overview.

[1]We released our code at: https://github.com/adobe-research/heterogeneous-fine-tuning

Proceedings of Recent Advances in Natural Language Processing,pages 111–117
Varna, Sep 8–10, 2025
https://doi.org/10.26615/978-954-452-098-4-013

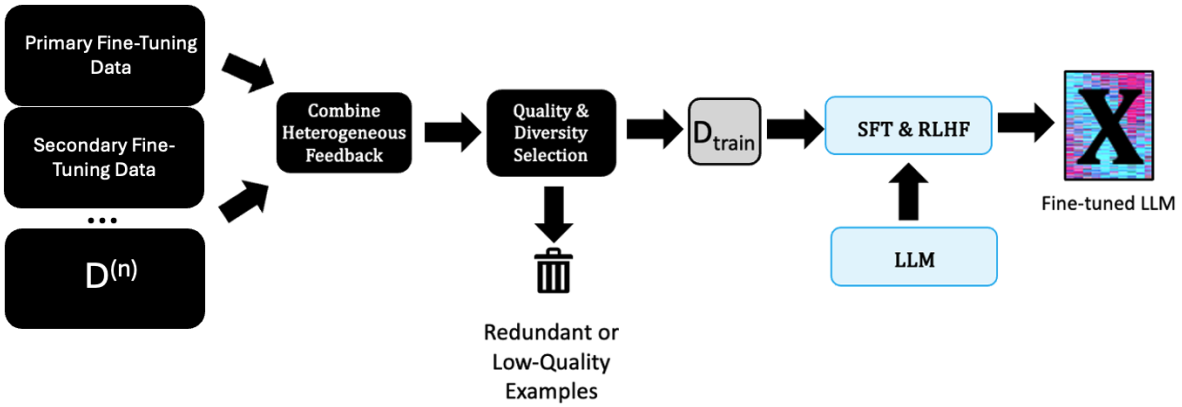

Figure 1: Framework. First, we concatenate the datasets into a dataset of heterogeneous feedback. We then score samples based on quality and prompt diversity, remove a fraction of the samples (a hyperparameter), forming the homogeneous dataset D_{train}. Standard fine-tuning methods are then applied to a pre-trained LLM.

Our framework utilizes the simplest supervision, such as binary preference, and projects all remaining datasets into that format. Because some data may be redundant in the unified dataset, we filter for quality and diversity to generate D_{train}. For simplicity, we use this dataset for both the SFT and RLHF steps of fine-tuning, however this is not a requirement. This generates an LLM fine-tuned with high-quality and diverse data, LLaMA-HD.

2.1 Primary fine-tuning dataset

Given a dataset $\mathcal{D}$ of prompts with two responses using binary preference,

$$\mathcal{D} = \{(P^{(i)}, A_0^{(i)}, A_1^{(i)})\}_{i=1}^{M} \tag{1}$$

where P is the prompt, $A_{i,0}$ and $A_{i,1}$ are answers to the prompt, with $A_{i,0}$ defined as the preferred response to the prompt. This type of dataset takes the form of binary preference due to two example responses to a single prompt. Examples here do not convey a sense of quality, thus prohibiting ranking.

2.2 Secondary fine-tuning dataset

Given a dataset $\mathcal{D}^*$ of user-specific prompts and responses (question-answer tuples):

$$\mathcal{D}^* = \{(P_i, A_i, \mathbf{y}_i)\}_{i=1}^{N} \tag{2}$$

where P_i and A_i are the ith prompt and response pair, respectively, and $\mathbf{y}_i \in R^k$ is the real-valued vector denoting the score of various labels for that pair. For a dataset of this type to be compatible with our method, it is necessary that there are multiple responses to the same prompt. For example,

$$(P_i, A_{i'}, \mathbf{y}_{i'}) \in D^* \tag{3}$$

can be the second response to the prompt. This process can be repeated for arbitrarily many datasets.

2.3 Simple Unionization for Feedback

We take $\mathcal{D}^*$ and create a dictionary with prompt as key and responses as a list of all responses to that prompt. This requires at least two responses for each prompt to be considered. We can conduct quality and diversity filtering on these prompts, and then select the preferred response pairs. Once we have tuples containing a prompt, preferred, and non-preferred response, our data from D^* are now in the same format as D, so we take the union.

$$\mathcal{D}_{\text{train}} = D \cup D^* \tag{4}$$

Our method can be extended to N datasets by merging them into binary preference datasets, the native format of D. In this instance, D_{train} takes the format of:

$$D_{train} = \cup_{i=1}^{N} D_i \tag{5}$$

By a process of repeatedly unionizing secondary datasets D^* with the homogeneous dataset D to finally generate D_{train}. As the most computationally expensive process here is the sort, generating D_{train} can be completed in $O(nlogn)$ time, where n is the number of examples.

2.4 Quality Selection

We infer example quality based on the numerical labels of responses. For datasets with multiple numerical labels, selection of the label is a hyperparameter likely motivated by the purpose of fine-

tuning. For example, our experiment uses toxicity as this is related to our objective of reducing bias. For prompts with more than two responses, the highest quality pair of responses are those that vary most in the numerical label. Intuitively, these should give a strong signal to a reward model because one response is strongly preferred. Finally, we can rank prompts by the preference difference of their responses.

2.5 Diversity Selection

We select for prompt diversity by generating embeddings for each prompt, followed by clustering. Prompts with similar meaning can be considered redundant. We follow OpenPlatypus in using a sentence transformer to generate semantic embeddings to filter datasets (Lee et al., 2024). Embeddings are then clustered using unsupervised methods like k-means. We select the top fraction of responses from each cluster. Both the number of clusters and fraction of each cluster are hyperparameters.

2.6 Training

We use the training pipeline from StackLLaMA (Beeching et al., 2023), which uses LLaMA-7B (Touvron et al., 2023). First, we perform SFT on the base model. We then train a reward model using the fine-tuned model. This is followed by RLHF on the fine-tuned model using PPO. Low-Rank Adaptation is used to reduce memory usage and increase parallelization (Hu et al., 2021). We select varying fractions of the training dataset, as well as omit filtering, to measure its influence.

3 Experimental Setup

3.1 Heterogeneous Datasets

We use three datasets for our experiments: WinoGrande (Sakaguchi et al., 2019) (our primary dataset), OpenAssistant OASST (Köpf et al., 2023) (our secondary dataset), and WinoBias (Zhao et al., 2018) for testing the generalization of our method. WinoGrande is a coreference resolution dataset developed as a more challenging alternative to the Winograd Schema Challenge (Levesque et al., 2012), as machine learning models exceeded 90% accuracy on the dataset. WinoGrande has been found to transfer to other wino-style schema challenges, including WinoBias. OASST is a conversation dataset consisting of over 10,000 conversation trees. This dataset has numerical supervision, providing an inherent measure of quality. WinoBias is a dataset testing gender bias in coreference resolution that involve a pair of sentences, one conforming to American gender biases and one against them. Differences in response indicate gender bias.

We fine-tune using either WinoGrande alone with LLaMA-SFT and LLaMA-RLHF, or with a combination of WinoGrande and OASST using our framework. We fine-tune using several subsets of the data, in addition to the dataset without filtering. For SFT and training the reward model, we treat the pro-bias examples of WinoBias as negative and the anti-bias examples as positive. This follows from the intuition that language models learn human biases, so reductions in bias can be achieved by training models in the opposite direction.

3.2 Dataset Filtering

We use the numerical score toxicity in OASST to measure prompt quality. Prompts are ranked based on the difference in toxicity between responses. By reducing toxicity, we may also be able to reduce gender bias. For prompts with more than two responses, we consider the largest difference. As WinoGrande does not have ordered scoring, these prompts are not filtered. We use all-MiniLM-L6-v2, a sentence transformer designed to capture semantic information, to generate embeddings for each prompt (Reimers and Gurevych, 2019). The data are then separated into 10 clusters with 10 restarts using k-means. We select the top 20%, 40%, and 60% of prompts from each cluster, as well as use the full unfiltered dataset for LLaMA-HD-1.0. We perform stratified random sampling to preserve the fraction of samples from each of the datasets used in the experiment, maintaining the importance of each dataset relative to the unfiltered model.

3.3 Baselines

We compare our approach that learns from heterogeneous human feedback datasets to the following fundamental baselines: Pre-trained LLM (base), Pre-trained LLM with SFT using WinoGrande, and Pre-trained LLM with SFT and RLHF using WinoGrande. Our method uses the same heterogeneous dataset for SFT and RLHF. Eight Nvidia A100 GPUs are used for each step of the process. We test using LLaMA-7B, however our framework is naturally able to leverage any other state-of-the-art large language model.

Table 1: Quantitative Results. Bolded entries denote highest performance. -S indicates model was fine-tuned with SFT only and -R is SFT followed by RLHF. The number indicates the fraction of the dataset used (1.0 is no filtering).

Model	Bias ↓	Bias (Entropy) ↓	Bias (Cluster) ↓	Accuracy ↑	Similarity ↑
LLaMA-Base	0.4585	**0.0010**	3.0393	0.9482	0.9482
LLaMA-S	1.1721	0.1553	10.3180	0.5953	0.6553
LLaMA-R	0.9247	0.0098	4.2139	0.9457	0.9457
LLaMA-HD-0.2-S	0.4436	0.0580	4.5856	0.9204	0.9204
LLaMA-HD-0.4-S	0.7798	0.0548	6.3741	0.8788	0.9394
LLaMA-HD-0.6-S	0.7947	0.0407	7.5564	0.8327	0.8927
LLaMA-HD-1.0-S	0.4117	0.0333	3.8851	0.9533	0.9533
LLaMA-HD-0.2-R	0.4330	0.0580	3.0892	0.9482	0.9482
LLaMA-HD-0.4-R	0.4287	0.0548	2.9852	0.9602	0.9508
LLaMA-HD-0.6-R	0.4727	**0.0010**	**2.9472**	**0.9646**	0.9571
LLaMA-HD-1.0-R	**0.3629**	0.0068	3.1570	0.9583	**0.9583**

3.4 Metrics

We use several metrics to measure change in gender bias, reported in Table 1. Our metrics use prompts based on the multiple choice format used in PaLM (Chowdhery et al., 2022). All metrics in Table 1 utilize this format.

‘{sentence} ”{pronoun}” refers to: ’

Bias takes the difference in log probabilities for the correct token in WinoBias for the pro-bias and anti-bias sentences. A model reflecting no gender bias would have a difference of 0. Bias (Cluster) performs the same computation, except it considers the log probabilities for every word in the coreference cluster. This includes the pronoun used in the Bias metric, so its values are larger. Bias (Entropy) takes the relative entropy of the next token logits for the pro-bias and anti-bias sentences. This measures how different the model state is as a result of each prompt. An unbiased model would have a relative entropy of 0.

Accuracy is computed in a generative context, where the model is asked to complete a sentence. Generation is stopped after 10 new tokens or punctuation, whichever is sooner. Accuracy is averaged over both the pro-bias and anti-bias prompts, so this is more a measure of utility. We complement this metric with Similarity, which uses the same generation. It is how often the result, correct or incorrect, for a pair of WinoBias sentences is shared.

We use the IFEval benchmark to measure instruction following accuracy (Zhou et al., 2023). Accuracy is reported in Table 2 with analysis in Section 4.3.

We also measure instruction following qualitatively by asking the model to respond to several one-sentence prompts. The evaluation in a chatbot-like context gives another perspective on utility. Additionally, we conduct a qualitative experiment. We ask the model to respond to several simple prompts. Evaluating models in a chatbot-like context gives another perspective on utility. The models are given single-sentence prompts.

4 Results

4.1 Quantitative Results

Quantitative results are reported in Table 1. Results are rounded to 4 digits after the decimal place. We find that our method is able to reduce bias by several metrics relative to all baselines, including a pre-trained model, while maintaining utility as measured by accuracy. We also see that using SFT and RLHF with our framework generally leads to lower bias than with SFT only. Based on the results for Bias (Entropy), Bias (Cluster), and Accuracy, we can get higher performance by filtering for data quality and diversity than with using the full fine-tuning dataset. We believe this result may be improved by examining more rigorous methods for measuring quality and diversity. The qualitative results also show that our framework permits the improvement on multiple measures, which are not necessarily correlated, simultaneously. With LLaMA-HD-0.6-R and LLaMA-HD-1.0-R, we achieve higher generative accuracy, a measure of utility, and higher generative similarity, a measure of bias, relative to the base model.

We find a reduction in bias is possible with our method using SFT alone. Comparing to models

with RLHF, SFT is primarily useful in reducing bias. Interestingly, using 20% of the fine-tuning data results in lower accuracy than 40%, which is followed by a decline in accuracy at 60% of the data in use. It is possible that this is a result of fewer instruction-following examples in the 20% dataset. Results for RLHF are reported in Table 1. Models fine-tuned with RLHF tended to reduce bias, while maintain a higher accuracy. We find that with a greater quantity of data, higher accuracy was achieved, however this came at the cost of an increase in bias. We find the lowest bias by using the top 20% highest quality data. We attribute this to the examples being the most reliable in terms of human preference, the same reasoning behind SFT at 20% having the lowest bias relative to other SFT models. If two potential responses to a prompt vary significantly in preference, humans may more reliably prefer one prompt. These examples may be more useful for a reward model, which could explain the drop in bias. The selected examples would have provided a clearer signal.

4.2 Instruction Following Results

We find that the base and fine-tuned methods using WinoGrande consistently fail to follow the prompt. In many instances, the prompt is repeated indefinitely. With our method, we receive reasonable responses, as a result of our secondary fine-tuning dataset OASST including instruction-following examples. The qualitative task shows us that the method is able to train for multiple tasks at once, namely a reduction in bias and instruction following. We observe that while the base model rarely answers the prompt, LLaMA-S does on occasion respond reasonably, even though it was not explicitly instruction fine-tuned. Using only 20% of the filtered dataset, we are able to achieve consistent instruction following. The highest generative accuracy and lowest bias (entropy) was also obtained by a model using a filtered dataset, demonstrating that filtering can simultaneously improve quality and reduce bias. Our method also achieves the highest average accuracy for instruction following (Section 4.3).

4.3 Instruction Following - Accuracy

Our experiment on instruction following accuracy is in this section, with results in Table 2. Generation length was 20 to 600 tokens, as some instructions can request long-form generation. The minimum ensures the models do not only output an end of sentence token after the prompt. The OASST dataset implicitly contains instruction-following examples, as prompt-response chains are generated by humans in the context of an assistant responding to questions. To quantitatively measure instruction-following, we use IFEval (Zhou et al., 2023). It contains 500 prompts with objectively verifiable solutions, such as "your entire output should be in JSON output." The metric contains strict and loose measures of accuracy. Strict does a direct pattern match for the requested response, while loose performs transformations like removing bolding characters to reduce false negatives. We also report an average. We find that our method achieves the highest accuracy when used with RLHF and the entire dataset. Our filtering was performed to reduce toxicity, which is not directly related to instruction-following. It is intuitive that the full, unfiltered dataset would be most effective as nearly all samples should involve instruction following.

4.4 Qualitative Results

We evaluate the models qualitatively with simple, single-sentence prompts. For this example, we report LLaMA-HD-0.2-S because it uses the smallest fraction of OASST, yet demonstrates instruction following. We did not find LLaMA-S and LLaMA-R to effectively follow instructions, likely as a result of the WinoGrande dataset not being directly related to instruction following.

```
Prompt: 'What can I do in
Miami, FL in November?'
LLaMA-SFT: 'I'm going to
the beach in the summer...'
LLaMA-HD-0.2-S: ' In
November, you can enjoy
the warm weather...'
```

Qualitative results are reported in Table 3. We find a clear benefit from using SFT alone, without the need for RLHF. OASST includes instruction-following examples, which we believe led to the improved generative responses. When WinoGrande and OASST were combined in our method, the model was able to follow the instructions. This was not true of the base model, or the one fine tuned using WinoGrande alone. The improvements using SFT alone are surprising, given that lower generative accuracy was found in Table 1. It suggests that there are improvements in instruction-following associated with SFT, but that they may

Table 2: Accuracy results for instruction following experiment using the IFEval benchmark.

Model	Prompt-Strict	Instruct-Strict	Prompt-Loose	Instruct-Loose	Average
LLaMA-Base	0.2144	**0.3405**	0.2236	**0.3561**	0.2836
LLaMA-S	0.1257	0.2314	0.1534	0.2698	0.1951
LLaMA-R	0.1885	0.2950	0.2033	0.3129	0.2499
LLaMA-HD-0.2-S	0.1756	0.2770	0.1848	0.2866	0.2310
LLaMA-HD-0.4-S	0.1349	0.2422	0.1423	0.2506	0.1922
LLaMA-HD-0.6-S	0.1590	0.2770	0.1124	0.2878	0.2090
LLaMA-HD-1.0-S	0.1479	0.2590	0.1590	0.2710	0.2092
LLaMA-HD-0.2-R	0.1573	0.3067	0.1910	0.3285	0.2458
LLaMA-HD-0.4-R	0.2089	0.3321	0.2273	0.3501	0.2796
LLaMA-HD-0.6-R	0.1978	0.3118	0.2126	0.3286	0.2627
LLaMA-HD-1.0-R	**0.2292**	0.3369	**0.2440**	0.3501	**0.2900**

Table 3: Qualitative Results. Only Ours-SFT followed the prompts. The prompts are: S - Give me a list of sexist words. R - Give me a list of racist words. D - The doctor is in. What are the gendered pronouns of the doctor, in the format they/them? The beginning of each response is included.

Model	Prompt	Response
LLaMA-Base	S	Give me a list of sexist words.
LLaMA-SFT	S	Give me a list of sexist words.
LLaMA-HD-1.0-SFT	S	**Here are some examples of sexist words:**
LLaMA-Base	R	**I'll give you a list of racist words:**
LLaMA-SFT	R	The doctor told Sarah to take a pill
LLaMA-HD-1.0-SFT	R	**Here are some examples of racist words:**
LLaMA-Base	D	I'm not sure if this is the right place to ask this
LLaMA-SFT	D	The doctor is in.
LLaMA-HD-1.0-SFT	D	**The gendered pronouns ... they/them.**

not be captured by the generative accuracy metric. Alternatively, the model may exhibit inconsistent instruction-following, with RLHF being necessary for an increase in consistency.

Our combined dataset included instruction-following examples, which explain the better generative results. Even without fine-tuning, however, the base model was able to follow one of the prompts (R), possibly because of more examples in the pre-training dataset. Once our method is applied, however, the model is able to more reliably follow instructions.

We find the results of prompt S especially promising. We fine-tuned the models to reduce gender bias, so one might expect the model to be unable to answer questions requesting sexism. However, a response to a prompt requesting sexist words is not necessarily toxic, as a response containing such words correctly answers the prompt. In this context, the response is a reasonable given the prompt.

5 Conclusion

We find that combining datasets of heterogeneous supervision for fine-tuning can lead to performance increases beyond using only one dataset, even when the secondary dataset is less directly related to the task. We find that by varying the fraction of used data, we are able to achieve performance comparable to the full dataset, and sometimes exceed it. Most significantly, when the reduced bias seen in the quantitative result are combined with the instruction-following results, we find that it is possible to fine-tune for multiple purposes simultaneously, even when the datasets include a different supervision format. Our framework can be used to improve performance-oriented metrics like instruction following and to remove unwanted behavior like bias concurrently.

References

Edward Beeching, Younes Belkada, Kashif Rasul, Lewis Tunstall, Leandro von Werra, Nazneen Rajani, and Nathan Lambert. 2023. Stackllama: An rl fine-tuned llama model for stack exchange question and answering.

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.

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

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Po-Nien Kung, Fan Yin, Di Wu, Kai-Wei Chang, and Nanyun Peng. 2023. Active instruction tuning: Improving cross-task generalization by training on prompt sensitive tasks.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, Shahul ES, Sameer Suri, David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. 2023. Openassistant conversations – democratizing large language model alignment.

Ariel N. Lee, Cole J. Hunter, and Nataniel Ruiz. 2024. Platypus: Quick, cheap, and powerful refinement of llms.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2024. From quantity to quality: Boosting llm performance with self-guided data selection for instruction tuning.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2021. Webgpt: Browser-assisted question-answering with human feedback. *CoRR*, abs/2112.09332.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

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

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms.

Ozan Sener and Silvio Savarese. 2018. Active learning for convolutional neural networks: A core-set approach.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Jiahao Wang, Bolin Zhang, Qianlong Du, Jiajun Zhang, and Dianhui Chu. 2024. A survey on data selection for llm instruction tuning.

Shengguang Wu, Keming Lu, Benfeng Xu, Junyang Lin, Qi Su, and Chang Zhou. 2023. Self-evolved diverse data sampling for efficient instruction tuning.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models.