Are You Sure? Rank Them Again: Repeated Ranking For Better Preference Datasets

Peter Devine
Lightblue Inc. (Tokyo, Japan)
peter@lightblue-tech.com

Abstract

Training Large Language Models (LLMs) with Reinforcement Learning from AI Feedback (RLAIF) aligns model outputs more closely with human preferences. This involves an evaluator model ranking multiple candidate responses to user prompts. However, the rankings from popular evaluator models such as GPT-4 can be inconsistent.

We propose the Repeat Ranking method, in which we evaluate the same responses multiple times and train only on those responses which are consistently ranked. Using 2,714 training prompts in 62 languages, we generated responses from 7 top multilingual LLMs and had GPT-4 rank them five times each. Evaluating on MT-Bench chat benchmarks in six languages, our method outperformed the standard practice of training on all available prompts.

Our work highlights the quality versus quantity trade-off in RLAIF dataset generation and offers a stackable strategy for enhancing dataset and thus model quality.

1 Introduction

Reinforcement learning has been shown to improve large language model (LLM) performance significantly (Yao et al., 2023; Havrilla et al., 2024), with this form of learning instructing an LLM both how *to* and how *not to* generate text.

This has come in the forms of Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022) and Reinforcement Learning from Artificial Intelligence Feedback (RLAIF) (Bai et al., 2022b; Lee et al., 2023), where a human or AI is used, respectively, to determine the relative quality of multiple responses to a given prompt. Based on these quality rankings, high quality and low quality responses are defined as "positive" and "negative" and this preference dataset is then used to train an LLM either with the help of a reward model or by directly training using a method such as Proximal Policy Optimisation (PPO) (Schulman et al., 2017), Direct Policy Optimisation (DPO) (Rafailov et al., 2024), or Odds Ratio Preference Optimisation (ORPO) (Hong et al., 2024). This style of training has lead to many of the improvements in recent years in LLM training, with both GPT-3.5 (Ouyang et al., 2022), trained with RLHF, and Starling (Zhu et al., 2023), trained with RLAIF, demonstrating gains upon previous state-of-the-art performance across many evaluation benchmarks.

Most publicly available preference data is monolingual, but we hypothesize that training a model on multilingual preference data will improve the resultant model's multilingual capabilities. This prompted us to create a multilingual preference dataset.

We follow previous methods for creating HLAIF preference datasets such as Nectar (Zhu et al., 2023) by first sampling human generated prompts from public datasets before generating various responses to each prompt using seven state-of-the-art LLMs. We then use a state-of-the-art LLM, GPT-4, to evaluate the relative ranking of each response.

However, we found that when the evaluation process was repeated on the same responses, different rankings were sometimes output by GPT-4. This suggested that the definition of positive and negative labels in these instances had a lower confidence than instances where GPT-4 would consistently output the same ranking given a set of responses.

Therefore, we hypothesized that training only on rankings that GPT-4 consistently outputs over multiple evaluations would lead to greater downstream evaluation performance compared to training on all rankings, both consistent and inconsistent. This lead us to propose the Repeat Ranking method, whereby responses are evaluated multiple times and the consistency of the rankings is used as a filter for inclusion or exclusion from the training set. A representation of our Repeated Ranking method can be found in Fig. 1.

We conducted experiments in which 2,714 mul-

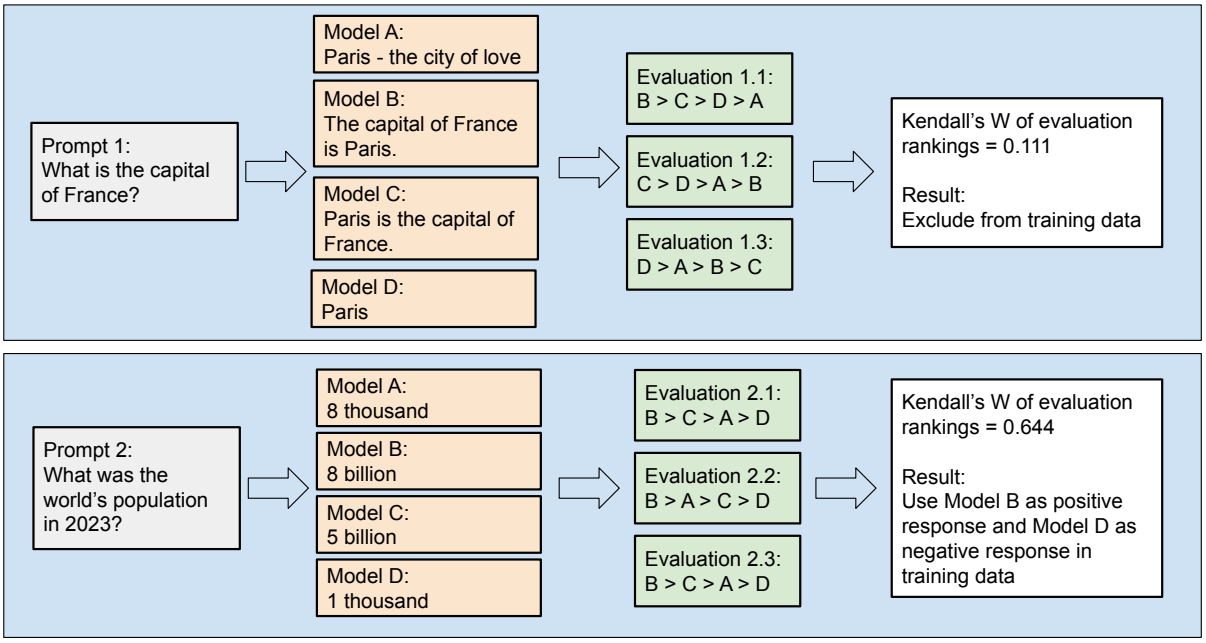


Figure 1: A visual description of how we select our data for training. We use our Repeat Ranking method to repeat the evaluations of the models multiple times and then only train on the best and worst responses which have a high Kendall's W, a measure of ranking agreement, associated with their ranking.

tilingual prompts were selected and 7 LLMs were used to generate responses for each prompt. We then evaluated each set of 7 responses 5 times using GPT-4. Finally, we propose a novel method for filtering evaluated preferences by measuring the consistency of the set of rankings for each evaluation using Kendall's W (Kendall and Smith, 1939). We conducted experiments training an LLM using all rankings, as well as the 75%, 50%, and 25% most consistent rankings. We then evaluated each trained model using the MT-Bench benchmark across 6 languages.

Our results show that training on the more consistently ranked responses gives greater downstream evaluation performance compared to training on all data for a majority of languages tested.

Our findings inform the creation of future preference datasets and offer a method of improving the quality of existing preference datasets. This may open up exciting new avenues for training LLMs and highlights the importance of high quality positive and negative data when training using RLAIF.

We make our training data[1], training code[2], and trained models[3] available online.

2 Related Work

LLM chat performance has been improved by training on RLHF datasets in multiple works within the literature.

The RLHF dataset used to train InstructGPT was created by having users and paid annotators evaluate multiple responses to a given prompt and indicating their preferred prompt (Ouyang et al., 2022). This work stated that "most comparisons are only labeled by 1 contractor for cost reasons" and that "having examples labeled multiple times could help identify areas where our contractors disagree, and thus where a single model is unlikely to align to all of them", indicating the seeming importance of having consistently similarly ranked preference data when training with RLHF.

In contrast, the OpenAssistant Conversations (OASST1) dataset (Köpf et al., 2024), contains conversation prompts and responses that are written by volunteers, with the responses evaluated by multiple volunteers. While this is a large dataset of more than 10,000 individual messages, over 70% of these conversations are in either English or Spanish, reducing OASST1's applicability to training a

[1] https://huggingface.co/datasets/lightblue/mitsu

[2] https://github.com/lightblue-tech/suzume/tree/main/mitsu

[3] https://huggingface.co/lightblue/suzume-llama-3-8B-multilingual-orpo-borda-half

multilingual model.

Generating data using human labellers is also costly, which is why several datasets have been constructed for RLAIF.

Previous work includes the use of "Constitutional AI" (Bai et al., 2022b) whereby an LLM is prompted to respond to a prompt before being tasked with revising that response to be less harmful and in line with principles set by researchers. The LLM then generates a less harmful response and the original and revised responses are then used to train another LLM using reinforcement learning.

Further work showed that training using RLAIF can lead to similar human evaluation scores compared to RLHF (Lee et al., 2023). This work also showed that RLAIF by training directly on response evaluation scores elicited from LLMs achieves greater down-stream task performance compared to the Constitutional AI approach of having an LLM revise existing responses.

Nectar (Zhu et al., 2023) is a preference dataset which first samples prompts from a variety of open source datasets, before generating responses based on these prompts using seven state-of-the-art LLMs (GPT-4, GPT-3.5-turbo, GPT-3.5-turbo-instruct, Command R+, Command R, LLama-2-7B-chat, and Mistral-7B-Instruct). These responses are then ranked once by GPT-4 and these rankings are used to train the Starling Alpha and Beta models using reinforcement learning. These prompts and responses are also all in English, meaning that this dataset is not suitable for training a multilingual model.

Due to the paucity of high quality multilingual models existing within the literature, we create one, which we call Mitsu.

Previous work has also shown that filtering reinforcement learning data can lead to higher down-stream task accuracy (Morimura et al., 2024). However, this approach relies on an external reward model to choose which data to filter, limiting the application of this approach to domains and languages that no existing reward model has been trained on.

3 Method

The overall objective of this piece of work was to create an LLM that was more proficient at multilingual chat than previous LLMs. In the course of creating such an LLM, we generated also insights into the process of creating high quality preference datasets. This section details how we used our Repeated Ranking method to make our training dataset named Mitsu, how we trained our model, and finally how we evaluated our LLM.

3.1 Preference Dataset Creation with Repeated Rankings

We create our Mitsu dataset by first following the process of how Nectar (Zhu et al., 2023) was developed by sampling human generated prompts derived from open source datasets such as the LMSYS-Chat-1M dataset (Zheng et al., 2023). Specifically, we select the multilingual stratified sample of prompts from the Tagengo dataset (Devine, 2024), which consists of 76,338 diverse human generated prompts in 74 languages. In order to reduce the costs of generating the dataset, we further stratify by languages, randomly sampling a maximum of 100 prompts per language. For languages with less than 100 prompts in the original dataset, we used all prompts for that language. This resulted in 2,996 prompts in total being selected.

Following the method used in the creation of the Nectar dataset, we used our sampled prompts to generate responses from seven state-of-the-art models. These were GPT-4 (gpt-4-0125-preview) (Achiam et al., 2023), GPT-3.5 Turbo (gpt-35-turbo-0301) (Ouyang et al., 2022), Command R (Gomez, 2024)[4], Command R+ (Gomez, 2024)[5], Qwen 1.5 32B Chat (Bai et al., 2023)[6], Qwen 1.5 72B Chat (Bai et al., 2023)[7], Starling 7B Beta (Zhu et al., 2023)[8].

These models were all chosen for their ability to output at least some multilingual text, which is why we did not consider using high performing but monolingual models such as Llama 3 (AI@Meta, 2024).

Our text generation settings were as follows. We set the generation temperature to 0 for all models, as some models such as Qwen have been shown to require smaller generation temperatures due to their larger vocabulary size and in order to make the generation deterministic to some extend. Future work could explore using more sophisticated tem-

[4] https://huggingface.co/CohereForAI/c4ai-command-r-v01

[5] https://huggingface.co/CohereForAI/c4ai-command-r-plus

[6] https://huggingface.co/Qwen/Qwen1.5-32B-Chat

[7] https://huggingface.co/Qwen/Qwen1.5-72B-Chat

[8] https://huggingface.co/Nexusflow/Starling-LM-7B-beta

perature set-ups per model, language, or prompt. We set our maximum number of tokens to generate as 2,048, and we discard any responses that have not been completed within this token limit. This was done to reduce both generation and evaluation time and costs, but future work could explore using longer generated sequences for a preference dataset. We used the popular vLLM library (Kwon et al., 2023) to generate responses with our local models, which were all models except GPT-4 and GPT-3.5-turbo. For GPT-4 and GPT-3.5-turbo, we generated responses using the Azure OpenAI endpoint. This resulted in 2,762 prompts having 7 full responses (one from each model), which we then ranked.

Our response evaluation again was conducted similarly to Nectar, where we used a similar system message describing the criteria for evaluating prompts as the original Nectar system message. We added one additional evaluation criteria to the original system message, which was "Is the response written naturally and fluently in the language that the prompter would expect?". This was added to make sure that highly rated responses were not correct but English responses to non-English prompts, which can occur in some LLMs.

Aside from our response evaluation criteria, we included a statement in the system message that instructed GPT-4 to output both a short explanation of the merits and drawbacks of each response, before outputting a ranking of the responses. This ranking consisted of responses labelled by alphabet character, using greater than ('>') and equals ('=') signs to determine which responses were evaluated as better and which were of equal quality. To avoid a systematic bias in our evaluations, responses were input to GPT-4 in a randomised order, with the responses being labelled A-G in order. We also take inspiration from work in generating the Nectar dataset in which randomised pairwise comparisons were used by instructing GPT-4 to write the explanation of the ranking in a dictated randomised order. The system message that we used in this work can be found in Figure 3 in the Appendix.

This ranking was generated by using a generation temperature of 0 and a maximum number of generated tokens as 1,024 with the gpt-4-0125-preview version of GPT-4. This resulted in a ranking for each set of 7 responses for each prompt.

Initial experiments investigating the reliability of this ranking showed that the ranking was liable to change significantly for some prompts. We rationalise this as follows. Imaging that a user asked three models "What is the capital of France?", and the responses were "Paris", "Lyon", and "Delhi". In this case, most human evaluators would be able rank the "Paris" answer as being the best answer and "Delhi" as being the worst answer. However, if the responses were instead more indistinguishable in terms of response quality, for example "Paris", "The capital city of France is Paris", and "Paris is the capital of France.", then even human evaluators may struggle to agree on which constituted the best and worst answers given the prompt. We hypothesize that for the same reason, AI evaluators give inconsistent rankings when faced with responses that are more indistinguishable from one another. Reinforcement learning techniques such as ORPO (Hong et al., 2024), which performs monolithic preference optimization without a reference model, rely on sufficiently different positive and negative training labels that an LLM can learn the contrast between the two. Therefore, training on too-similar positive and negative labels may result in a degeneracy of the model overall. Hence, when we observed the lack of consistency in GPT-4's rankings for some responses, we hypothesized that training on only the more consistently ranked outputs would lead to a better evaluation performance than training on all rankings. Therefore, we repeat the ranking process five times, only changing the random order of the responses and the instructed random order of the ranking explanation each time. We discarded any cases in which a generation failed or where the ranking could not be parsed from the generated evaluation, leaving 2,714 individual prompts. We found that only 8.4% of all top responses were ranked top all 5 times, and only 20.2% of bottom responses were ranked bottom all 5 times, which again motivates our work in generating multiple evaluations for each set of responses per prompt.

With these responses, we calculated the Kendall's W (Kendall and Smith, 1939) for each set of rankings. According to Field, "Kendall's Coefficient of Concordance, W, is a measure of the agreement between several judges who have rank ordered a set of entities" (Field, 2005), and we use it to determine how well the repeated evaluation rankings agree. We justify using Kendall's W as a measure of inter-ranker agreement due to its previous use as a measure of ranking agreement within

Model name	Average Borda Count
GPT-3.5 Turbo	15.91
Starling 7B Beta	16.57
Qwen 1.5 32B	18.17
Command R	20.47
Qwen 1.5 72B	20.51
Command R +	21.54
GPT-4	26.78

Table 1: Average Borda count per model across 5 evaluations.

the mathematical literature. However, since we ultimately just use the top and bottom responses from our rankings, we consider that comparing only the rankings of those two responses directly could possibly be simpler and could potentially lead to better results. We leave this for future work to explore this avenue.

We use this W score to generate three training subsets of Mitsu, where we only trained on responses with the top 25% (674 prompts), 50% (1,350 prompts), 75% (2,018 prompts) of W scores. We also trained a model using the entire Mitsu dataset (2,714 prompts).

In order to train using ORPO, we selected positive and negative responses to prompts. These effectively train a model to generate outputs similar to the positive responses and dissimilar to the negative responses. We selected these responses by calculating the Borda Count (Borda, 1781; Reilly, 2002) of each response over the 5 evaluations, and then selecting the models with the highest and lowest Borda counts for positive and negative, respectively. We randomly sample in cases where there is a tie in the Borda score between the multiple best or worst scores.

Table 1 shows the average Borda score for each model evaluated and Fig. 2 shows the amount of times each model's response was used as the positive and negative response.

We make the top 25%, top 50%, top 75%, and full training datasets available online[9].

3.2 Training

We train using our prepared datasets on Suzume 8B Multilingual (Devine, 2024), a multilingual finetune of Llama 3 (AI@Meta, 2024), using ORPO. We chose to train using ORPO due to its demonstrated greater performance compared to the most popular other current RLAIF method, DPO (Hong et al., 2024). We trained using the ORPO settings made available on the Axolotl LLM training package[10] which uses the TRL (von Werra et al., 2020) implementation of the ORPO algorithm. We chose to train on the Suzume 8B Multilingual model as it has the highest MT-Bench scores for a majority of evaluation languages compared to other commercially usable open source models under 10 billion parameters. We train for one epoch for each dataset with an ORPO alpha value set to 0.1, our maximum token sequence length was set to 8,192, and our learning rate was set to 8e-6. The full training configuration for each model can be found on their model cards[11].

For convenience, we refer to the models trained on the top 25%, 50%, 75%, and 100% of W score subsets as Suzume-ORPO-25, Suzume-ORPO-50, Suzume-ORPO-75, and Suzume-ORPO-100, respectively.

3.3 Evaluation

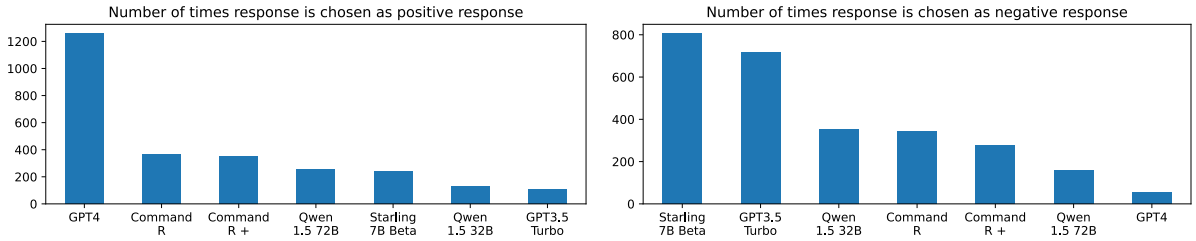
We evaluate our models using the multilingual version of the MT-Bench score over 6 languages (Chinese, English, French, German, Japanese, and Russian). This evaluation tests a model's ability to perform tasks such as writing, roleplay, extraction, reasoning, math, coding, STEM knowledge, and humanities knowledge in a given language, using GPT-4-Turbo as the evaluator of the model's responses. Each category contains 10 prompts, with each response being ranked out of 10, to give a final average score over all prompts. We report the 2-turn scores on this benchmark. Note that we do not report Russian performance on math, coding, and reasoning questions as reference answers were not available for these questions. We evaluate all four of our ORPO trained models (Suzume-ORPO-25, Suzume-ORPO-50, Suzume-ORPO-75, and Suzume-ORPO-100), as well as our base model (Suzume-Base) on the MT-Bench benchmark over all 6 languages. As a further baseline, we also evaluate the GPT-3.5-Turbo model (Ouyang et al., 2022) on each language.

As an additional evaluation, we evaluate over

[9] Available at in https://huggingface.co/collections/lightblue/mitsu-datasets-67076f8293b57ae8b2c17293

[10] https://github.com/OpenAccess-AI-Collective/axolotl

[11] Available at https://huggingface.co/collections/lightblue/orpo-experiments-6707702969a9340fa312405f



(a) Positive responses

(b) Negative responses

Figure 2: Plots of how often each model's response was chosen as the positive/negative response for training using the Borda count. We observe that a plurality but not a majority of our positive training data comes from GPT-4, while the vast majority of our negative training data comes from responses by Starling and GPT-3.5-Turbo.

the Belebele benchmark, which is a log-probability based benchmark which calculates the probabilities for generating the correct answer tokens given a prompt compared to the probabilities of generating three possible incorrect answers (Bandarkar et al., 2023). We report the accuracy, which is the percentage of test examples where the probability of generating the correct answer from the prompt was higher than the probability of outputting any of the wrong answers. We apply this benchmark over the 6 languages we use in our MT-Bench evaluation, as well as 6 other languages that we selected at random: Arabic, Azerbaijani, Bangla, Croatian, Norwegian, and Thai. Note that this does not test an LLM's chat abilities, but rather tests an LLM's ability to output factual information.

4 Results

Table 2 presents the MT-Bench scores across 6 languages for our 4 ORPO subsets compared to the base model and GPT-3.5-Turbo.

All ORPO models surpassed the base model in nearly every language, underscoring the value of ORPO training for enhancing chat capabilities. Furthermore, Suzume-ORPO-50 outperformed Suzume-ORPO-100 in 5 out of 6 languages, despite being trained on half the data. Suzume-ORPO-25 and Suzume-ORPO-75 achieved the highest scores in one language each, but Suzume-ORPO-50 provided the best overall balance.

While the base model did not exceed GPT-3.5-Turbo in any language, Suzume-ORPO-50 outperformed GPT-3.5-Turbo in 4 out of 6 languages, demonstrating that ORPO training enables LLMs match or surpass GPT-3.5-Turbo on chat benchmarks. However, GPT-3.5-Turbo still led in English and Japanese.

We also conducted other small scale tests to further probe the effects of ORPO training. One notable test (Suzume-ORPO-GPT on Table 4) was training using all prompt responses from the models with the best and worst Borda scores, GPT-4 and GPT-3.5 respectively, but we found that this lead to a lower average MT-Bench scores compared to the Suzume-ORPO-100 model. This indicates the importance of model diversity and selecting appropriate responses when generating RLAIF datasets.

Another test (Llama-ORPO-50 on Table 4) we conducted was directly ORPO training a Llama 3 8B Instruct model on the same dataset as Suzume-ORPO-50, but we found that this model had lower MT-Bench scores across all languages. This demonstrates the continued necessity for fine-tuning before conducting ORPO training.

The final small scale test (Suzume-ORPO-random-50 on Table 4) we conducted was training a model on a randomly selected half of the entire Mitsu dataset. This allowed us to isolate the effects of example selection by using Kendall's W, as this model was trained on the same amount of data as Suzume-ORPO-50. We find that Suzume-ORPO-random-50 model has lower MT-Bench scores across all languages compared to Suzume-ORPO-50, indicating the importance of selecting training prompts based on Kendall's W score.

The Belebele scores for each of our trained models can be found in Table 3. We observe that the base model exhibits greater or equal performance on average on this benchmark compared to Suzume-ORPO-100. This contrasts with our MT-Bench scores which showed that ORPO training unambiguously improved chat performance compared to the base model. However, despite the observed drop in Belebele score when performing full ORPO training, we also observe that Suzume-ORPO-75 and Suzume-ORPO-25 are able to largely achieve

Language	GPT-3.5-Turbo	Suzume-Base	Suzume-ORPO-100	Suzume-ORPO-75	Suzume-ORPO-50	Suzume-ORPO-25
Chinese	7.55	7.11	7.65	**7.77**	7.74	7.44
English	**8.26**	7.73	7.98	7.94	7.98	8.22
French	7.74	7.66	**7.84**	7.46	7.78	7.81
German	7.68	7.26	7.28	7.64	7.70	**7.71**
Japanese	**7.84**	6.56	7.20	7.12	7.34	7.04
Russian	7.94	8.19	8.30	8.74	**8.94**	8.81
Mean	7.83	7.42	7.71	7.78	**7.91**	7.84

Table 2: The MT-Bench chat benchmark scores for each model evaluated across each language. Bolded values are greatest in their row. We improve upon base model evaluation performance across all languages for nearly all ORPO models. Interestingly, we find that training on the 50% most consistently evaluated prompts leads to greater than or equal evaluation scores than training on all prompts for 5 of 6 languages evaluated.

comparable or better performance with the base model on many languages in this benchmark. This indicates that our ORPO training data selection criteria may be beneficial to mitigating some of the issues we demonstrate of lower performance on log-probability based for ORPO trained models.

We also observe that Suzume Base performs better on two languages (Chinese and Thai) than any ORPO trained model. This may simply be due to the fact that OPRO training, and particularly naive ORPO training (i.e. Suzume ORPO-100), seems to result in reduced performance in Belebele and so even when selecting training examples using Kendall's W, the drop in performance is too large to compensate for.

5 Discussion

Our results demonstrate the importance of ORPO training in improving the chat abilities of finetuned models. This, in turn, highlights the importance of creating high quality preference datasets to train LLMs using the ORPO method. Our results showing that model trained on less, but more consistently evaluated, preferences can achieve greater chat benchmark performance than training on all the data. This has the double benefits of increasing performance while reducing training cost by as much as four times for training on our 25% training subset. However, the extra inference computation required to rank responses multiple times is an increased cost with this method of dataset creation.

This could benefit both current and future datasets, with datasets such as Nectar (Zhu et al., 2023) potentially being improved by re-evaluating the dataset's responses and filtering out less consistently evaluated rows.

We theorize that the correct balance between consistency and data volume (i.e. where the cut-off for Kendall's W would be) may vary between tasks, but we have shown that for our multilingual chat setting the benefit on evaluation performance of having a threshold above which we keep our data.

Our results are also purely dataset-based, meaning that they might be able to be stacked with other recent LLM training methods such as SimPO (Meng et al., 2024) and ExPO (Zheng et al., 2024a).

6 Future Work

Our results suggest that the technique of repeated evaluations on preference data and only keeping the consistently evaluated prompts and responses for training could be applied to other RLAIF and RLHF datasets. Future work could include investigating whether training only using prompts and responses with high agreement in the evaluations from human annotators could lead to higher accuracy than training on all prompts and responses.

Another potential avenue for future work is using more than one evaluator model for ranking responses. In this work, we only used GPT-4, but there are other state-of-the-art LLMs such as Claude 3 (Anthropic, 2024) and Gemini 1.5 Pro (Reid et al., 2024). We theorize that combining the evaluations of multiple high performance LLMs could serve to create more robust evaluations of responses and mitigate the demonstrated bias that any one LLM exhibits (Feng et al., 2023; Cao et al., 2023). The Mitsu dataset that we use to train our model is single-turn, meaning that each example

	Suzume Base	Suzume ORPO-100	Suzume ORPO-75	Suzume ORPO-50	Suzume ORPO-25
Arabic	64.3	52.6	**65.3**	54.7	64.6
Azerbaijani	50.3	37.6	**52.3**	45.3	52.1
Bangla	46.0	37.0	**49.7**	43.2	46.3
Chinese	**78.0**	64.4	76.1	70.0	75.7
Croatian	59.4	47.4	60.7	53.0	**61.1**
English	84.2	75.2	83.2	83.0	**84.7**
French	77.3	64.4	75.7	72.2	**77.6**
German	68.0	53.8	67.9	65.9	**68.8**
Japanese	66.7	57.1	63.7	58.2	**68.0**
Norwegian	67.0	52.4	67.2	62.2	**67.7**
Russian	71.6	51.9	71.4	57.3	**72.9**
Thai	**63.3**	47.9	61.3	57.1	63.0
Mean	66.4	53.5	66.2	60.2	**66.9**

Table 3: Belebele scores for each trained model across the 12 languages that we evaluate on. We observe that full ORPO training leads to much lower Belebele scores compared to the base fine-tuned model. However, we also observe that our method of selecting fewer ORPO training examples is able to marginally improve on the performance of the base model for most languages.

consists of a single prompt-response pair for both positive and negative responses. Future work could expand on this to add multi-turn conversations, as was done by Nectar (Zhu et al., 2023).

The Mitsu dataset also consists of prompts sampled from the Tagengo dataset (Devine, 2024), which are derived from users prompts to LLMs hosted on a demo site. We theorize that these prompts are a mixture of easy and difficult for an LLM to answer. Training on tasks that LLMs are already highly proficient at might be a waste of training resources, so future work could filter prompts based on their perceived difficulty for LLMs. We believe that this may improve LLMs abilities on these difficult tasks.

In our experiments, we chose to rank responses 5 times due to that being the financial limit of our experiment. However, future work could empirically find an optimal number of times to repeat evaluations to obtain a reliable Kendall's W score.

A slight limitation of the Repeated Ranking approach is the increased inference cost in evaluating responses multiple times as an analogue for determining the confidence of the ranking model in the ranking. Future work could explore mitigating this effect by evaluating the combined log probability of a single ranking output and training using only the responses from rankings with the highest probability.

Tools and agents have also been shown to augment the abilities of LLMs (Parisi et al., 2022; Gao et al., 2023; Schick et al., 2024). Future work could explore using tools or agents to enhance the evaluation abilities of the evaluator LLM when evaluating prompt responses. For example, a search tool could determine the veracity of factual claims, or a calculator tool would be able to confirm the mathematical results of an LLM. We theorize that this would lead to more accurate evaluation and would ultimately lead to more accurate LLMs.

7 Conclusion

In this study, we explored the impact of repeated rankings from an AI evaluator (GPT-4) on training reinforcement learning from AI feedback (RLAIF) models for multilingual chat capabilities. We found that responses evaluated consistently by GPT-4 led to higher downstream performance across multiple languages, compared to training on all data regardless of evaluation consistency. Our findings indicate that selective training based on evaluation consistency can enhance chat performance and offer a method to improve existing preference datasets. This highlights the balance between quality and quantity when constructing datasets for RLAIF. Our work opens avenues for further optimizing RLAIF datasets and refining training methodologies to develop more proficient multilingual LLMs.

Limitations

One limitation of this work was the size of the data that we trained upon. Our Mitsu dataset, in total, consisted of less than 3k examples, whereas many popular preference datasets such as Nectar (Zhu et al., 2023) and the HH-RLHF (Bai et al., 2022a) dataset consist of hundreds of thousands of examples. Therefore, we are yet to show whether our proposed response selection technique extends to datasets of that size.

Secondly, the differences in our results are relatively small. While we show relatively consistent improvement in chat performance in models trained over our selected subsets (Suzume-ORPO-25, Suzume-ORPO-50, Suzume-ORPO-75) over the model trained on the whole dataset (Suzume-ORPO-100), these differences are small in magnitude (largely <10% difference). It is nevertheless notable that even demonstrating that chat performance does not decrease with fewer training examples is a useful result that can inform more efficient ORPO training in the future. Therefore, it remains for future work to determine if the improvements in chat ability increase with a larger training set.

Finally, a limitation of this research is that we rely on GPT-4 for our evaluation using the MT-Bench benchmark. This could bias the model as GPT-4 has been shown to exhibit self-enhancement bias (Zheng et al., 2024b), where it evaluates its own responses higher compared to human evaluation, indicating that we may be overfitting to GPT-4's preferences rather than general human ones. However, GPT-4 is the current state-of-the-art for LLMs and has been shown to have very high correlation with human preferences (Zheng et al., 2024b). Moreover, our evaluations using Belebele dataset do not use an LLM for evaluation and again indicate that the accuracy of some of our ORPO trained models over many languages increases compared to the base model.

Ethics Statement

We have considered the ethical implications of releasing both our training data and trained models. There is the potential for LLMs and training data to be misused, but since we demonstrate that our final LLM is comparable to a publicly available LLM (GPT-3.5-Turbo) that has since been superceded by more recent LLMs (GPT-4, Llama 405B (Dubey et al., 2024) etc.), we assume that the risk impact of our sharing these models and data is minimal.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

AI@Meta. 2024. Llama 3 model card.

AI Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022a. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022b. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Lucas Bandarkar, Davis Liang, Benjamin Muller, Mikel Artetxe, Satya Narayan Shukla, Donald Husa, Naman Goyal, Abhinandan Krishnan, Luke Zettlemoyer, and Madian Khabsa. 2023. The belebele benchmark: a parallel reading comprehension dataset in 122 language variants. *arXiv preprint arXiv:2308.16884*.

J. Borda. 1781. Mémoire sur les élections au scrutin. *Histoire de L'Académie Royale des Sciences, Paris.*

Yong Cao, Li Zhou, Seolhwa Lee, Laura Cabello, Min Chen, and Daniel Hershcovich. 2023. Assessing cross-cultural alignment between chatgpt and human societies: An empirical study. *arXiv preprint arXiv:2303.17466*.

Peter Devine. 2024. Tagengo: A multilingual chat dataset. *arXiv preprint arXiv:2405.12612*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Shangbin Feng, Chan Young Park, Yuhan Liu, and Yulia Tsvetkov. 2023. From pretraining data to language models to downstream tasks: Tracking the trails of political biases leading to unfair nlp models. *arXiv preprint arXiv:2305.08283*.

Andy P Field. 2005. Kendall's coefficient of concordance. *Encyclopedia of statistics in behavioral science*, 2:1010–11.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Pal: Program-aided language models. In *International Conference on Machine Learning*, pages 10764–10799. PMLR.

Aidan Gomez. 2024. Command R: Retrieval-Augmented Generation at Production Scale.

Alex Havrilla, Yuqing Du, Sharath Chandra Raparthy, Christoforos Nalmpantis, Jane Dwivedi-Yu, Maksym Zhuravinskyi, Eric Hambro, Sainbayar Sukhbaatar, and Roberta Raileanu. 2024. Teaching large language models to reason with reinforcement learning. *arXiv preprint arXiv:2403.04642*.

Jiwoo Hong, Noah Lee, and James Thorne. 2024. Reference-free monolithic preference optimization with odds ratio. *arXiv preprint arXiv:2403.07691*.

Maurice G Kendall and B Babington Smith. 1939. The problem of m rankings. *The annals of mathematical statistics*, 10(3):275–287.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, et al. 2024. Openassistant conversations-democratizing large language model alignment. *Advances in Neural Information Processing Systems*, 36.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Lu, Thomas Mesnard, Colton Bishop, Victor Carbune, and Abhinav Rastogi. 2023. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. *arXiv preprint arXiv:2405.14734*.

Tetsuro Morimura, Mitsuki Sakamoto, Yuu Jinnai, Kenshi Abe, and Kaito Air. 2024. Filtered direct preference optimization. *arXiv preprint arXiv:2404.13846*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Aaron Parisi, Yao Zhao, and Noah Fiedel. 2022. Talm: Tool augmented language models. *arXiv preprint arXiv:2205.12255*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Benjamin Reilly. 2002. Social choice in the south seas: Electoral innovation and the borda count in the pacific island countries. *International Political Science Review*, 23(4):355–372.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2024. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, and Shengyi Huang. 2020. Trl: Transformer reinforcement learning. `https://github.com/huggingface/trl`.

Zhewei Yao, Reza Yazdani Aminabadi, Olatunji Ruwase, Samyam Rajbhandari, Xiaoxia Wu, Ammar Ahmad Awan, Jeff Rasley, Minjia Zhang, Conglong Li, Connor Holmes, et al. 2023. Deepspeed-chat: Easy, fast and affordable rlhf training of chatgpt-like models at all scales. *arXiv preprint arXiv:2308.01320*.

Chujie Zheng, Ziqi Wang, Heng Ji, Minlie Huang, and Nanyun Peng. 2024a. Weak-to-strong extrapolation expedites alignment. *arXiv preprint arXiv:2404.16792*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric. P Xing, Joseph E. Gonzalez, Ion Stoica, and Hao Zhang. 2023. Lmsys-chat-1m: A large-scale real-world llm conversation dataset.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024b. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

Banghua Zhu, Evan Frick, Tianhao Wu, Hanlin Zhu, Karthik Ganesan, Wei-Lin Chiang, Jian Zhang, and Jiantao Jiao. 2023. Starling-7b: Improving llm helpfulness & harmlessness with rlaif.

````
You are an evaluator AI. Your task is to rank multiple responses to a given prompt from best to worst
You will first be given the original prompt, and then seven possible responses to that prompt,
    ↪labelled alphabetically.
You should first write a very brief (<40 words per model) explanation of the merits and drawbacks of
    ↪the responses, before giving the ranking itself.
This explanation of each response should be in a randomised order (go in the order of '{randomly
    ↪shuffled list of alphabet letters from A-G}').
Make sure you explain and rank all responses, do not leave any out in your explanation or ranking.
The ranking should be a list of alphabet characters that describe the ranking, with '>' denoting the
    ↪left item is ranked higher than the right item and '=' denoting that the items are of equal
    ↪ranking (e.g. 'Z>Y>X=W>V>U=T').

The user input will look like this:

```
<<<PROMPT>>>
AN EXAMPLE USER PROMPT

<<<RESPONSE A>>>
EXAMPLE RESPONSE A

<<<RESPONSE B>>>
EXAMPLE RESPONSE B

<<<RESPONSE C>>>
EXAMPLE RESPONSE C

<<<RESPONSE D>>>
EXAMPLE RESPONSE D

<<<RESPONSE E>>>
EXAMPLE RESPONSE E

<<<RESPONSE F>>>
EXAMPLE RESPONSE F

<<<RESPONSE G>>>
EXAMPLE RESPONSE G
```

and your output should look like this:

```
<<<EXPLANATION>>>
[SHORT EXPLANATION OF THE RANKING]

<<<RANKING>>>
[SEPARATED LIST OF ALPHABET CHARACTERS THAT DESCRIBE THE RANKING]
```

The evaluation rubric is as follows:

* Is the response relevant? The response should be the best possible answer.
* Is the response truthful?
* Is the response accurate? The response should accurately fulfill the prompt's request.
* If a creative answer is expected, is the response creative? If an analytical answer is expected, is
    ↪ the response factual/objectively correct?
* Is the response written naturally and fluently in the language that the prompter would expect?
* Is the response detailed? The response should at minimum satisfy the full level of detail required
    ↪by the prompt.
````

Figure 3: System message for generating evaluations

Language	GPT3.5-Turbo	Suzume-Base	Suzume-ORPO-100	Suzume-ORPO-75	Suzume-ORPO-50	Suzume-ORPO-25	Suzume-ORPO-GPT	Llama-ORPO-50	Suzume-ORPO-random-50
Chinese	7.55	7.11	7.65	**7.77**	**7.74**	7.44	7.54	7.52	7.41
English	**8.26**	7.73	7.98	7.94	7.98	8.22	7.79	7.84	7.72
French	7.74	7.66	**7.84**	7.46	7.78	7.81	7.22	7.33	7.51
German	7.68	7.26	7.28	**7.64**	7.7	**7.71**	7.37	7.47	7.03
Japanese	**7.84**	6.56	7.2	7.12	7.34	7.04	7.14	7.22	6.82
Russian	7.94	8.19	8.3	8.74	**8.94**	8.81	8.34	8.68	8.32
mean	7.83	7.42	7.71	7.78	**7.91**	7.84	7.57	7.68	7.47

Table 4: Extended MT-Bench scores across 6 languages for all models evaluated, including small-scale tests.