Fluent Alignment with Disfluent Judges: Post-training for lower-resource languages

Anonymous authors
Paper under double-blind review

Abstract

We propose a post-training method for lower-resource languages that preserves fluency of language models even when aligned by disfluent reward models. Preference-optimization is now a well-researched topic, but previous work has mostly addressed models for English and Chinese. Lower-resource languages lack both datasets written by native speakers and language models capable of generating fluent synthetic data. Thus, in this work, we focus on developing a fluent preference-aligned language model without any instruction-tuning data in the target language. Our approach uses an on-policy training method, which we compare with two common approaches: supervised finetuning on machine-translated data and multilingual finetuning. We conduct a case study on Norwegian Bokmål and evaluate fluency through native-speaker assessments. The results show that the on-policy aspect is crucial and outperforms the alternatives without relying on any hard-to-obtain data.

1 Introduction

Instruction-tuning and preference-optimization have become a cornerstone of modern language models, enabling base models to follow instructions and engage in helpful dialogue. However, this progress has been largely confined to high-resource languages like English and Chinese, which benefit from extensive human-written datasets and sophisticated language models capable of generating fluent synthetic data. Lower-resource languages face a fundamental challenge: they lack both instruction datasets written by native speakers and fluent models that could generate high-quality training data (Guo et al., 2025). This work addresses a critical question for the democratization of language technology: how can we create fluent preference-aligned language models for lower-resource languages without any instruction-tuning dataset in the target language?[1]

Current approaches to post-training language models for lower-resource languages mostly rely on static and predefined instruction-tuning datasets (Suzuki et al., 2023; Chouikhi et al., 2024; Lim et al., 2025), which are usually machine-translated from English (Pipatanakul et al., 2023; Santilli & Rodolà, 2023; Ranaldi & Pucci, 2023; Üstün et al., 2024; Nguyen et al., 2024; Bari et al., 2025; Zosa et al., 2025). While this approach shows promising results when evaluated on standard NLP benchmarks, the translation process introduces subtle linguistic artifacts – *translationese* – that make the resulting models disfluent in the target language (Yu et al., 2022; Dutta Chowdhury et al., 2022). Translationese is produced even by professional human translators, and machine-translation models are negatively impacted by it to an even larger degree (Bizzoni et al., 2020). Thus, we believe that post-training for lower-resource languages needs to shift away from such data.

Recent advances in reinforcement learning from AI feedback (RLAIF; Bai et al., 2022) offer a potential solution to this challenge. By training models through on-policy reinforcement learning, where the model learns from its own generated responses rather than from fixed datasets, we can potentially avoid exposing the model to disfluent text altogether. The key insight is that a model that has learned fluent generation through extensive pretraining on native texts can maintain this fluency as long as it never trains on unnatural examples during the alignment phase.

In this work, we propose a fluency-aware post-training method that leverages on-policy reinforcement learning to align language models for lower-resource languages without compromising their linguistic

[1] *Fluency* refers to the linguistic quality of text that makes it natural, grammatical, and easy to read. It should look like a text written by a native speaker. It is independent of other qualities such as factual accuracy.

naturalness. Crucially, we never train the model on any translated responses, preserving the fluent generation capabilities learned during pretraining. We demonstrate that even a disfluent judge model can successfully guide a fluent policy, as long as the judge sufficiently understands the target language to evaluate response quality.

Our approach is validated on Norwegian Bokmål, a relatively small language with about 5 million speakers. Norwegian is a good fit for the assessment because it is on one hand a language without any manually written instruction datasets, yet the research community has pretrained a number of Norwegian language models that can be leveraged for post-training. Lastly, we were able to employ five native Norwegian speakers for accurate manual evaluation of fluency.

Our main contributions are:

- We propose an on-policy reinforcement learning method for post-training in lower-resource languages that maintains fluency without requiring any instruction datasets in the target language.
- We demonstrate through extensive human evaluation with native Norwegian speakers that on-policy training produces more fluent models than supervised finetuning on translated data.
- We show that fluent aligned models can be bootstrapped using disfluent judge models, enabling post-training for languages without existing fluent instruction-tuned models.
- We provide comprehensive ablations revealing the critical importance of avoiding any exposure to translated responses during training, even in small quantities.

2 Fluency-aware post-training

This section describes our proposed method for post-training language models on a target lower-resource language without any dedicated dataset in that language. The overall approach consists of three stages: i) pretraining on the target language, ii) short SFT alignment on English, and iii) on-policy alignment on the target language.

The key principle is to *never train the language model on any unnatural text*.

Pretraining on target language The first stage is essential to ensure that the base language model learns all necessary linguistic knowledge and is able to generate fluent outputs in the target language. Our study focuses on the subsequent training stages and does not cover this pretraining stage – that has already been studied in detail by Gururangan et al. (2020); Ibrahim et al. (2024); Kim et al. (2024); Samuel et al. (2025). Specifically for Norwegian, we build upon the multi-stage continual pretraining from Samuel et al. (2025).

Short SFT alignment on English The second stage is crucial to teach the model to respond to user prompts and follow the specific chat format (Appendix D.1). Typically, supervised finetuning would be done on a carefully curated set of conversations in the target language, but such resources are usually not available for lower-resource languages – so we instead opt for a short alignment on a small high-quality English dataset. Specifically, we use the 1 000 curated prompt-response pairs from LIMA (Zhou et al., 2023) and train the base model on this dataset for one epoch. The short training ensures that the model does not catastrophically forget its innate knowledge of the target language.

On-policy alignment on target language Finally, in the third stage, the language model is aligned to respond in a helpful, truthful and safe way – without losing its fluency in the target language that has been learned in the initial pretraining stage. This is achieved by training with online on-policy reinforcement learning where the model is trained solely on responses sampled from itself. In this way, the model is never pushed away from the subspace of fluent outputs is has learned to prefer during pretraining. A key observation is that we do not need to train any reward model as it suffices to use an LLM-as-a-judge system to for the reward signal; *as long as the judge understands the target language, it does not have to be fluent to produce a fluent policy.* We evaluate this hypothesis later in Section 4.1. This effectively allows us to bootstrap fluent models in languages without any instruction datasets and without any existing fluent language models.

2.1 Online on-policy reinforcement learning

This section describes the on-policy training in more detail. It is important to note that any implementation of online on-policy training with feedback from an LLM-as-a-judge system should work just as well in terms of fluency of the final policy network – our implementation is chosen to be simple and comparable to the baseline approach of supervised finetuning, as detailed later in Section 3.

Reward model The standard approach to on-policy alignment – reinforcement learning from human feedback (RLHF; Christiano et al., 2017; Stiennon et al., 2020; Ouyang et al., 2022) – first trains a Bradley-Terry reward model on a preference dataset from the target domain. Since we are restricted to a lower-resource language, we suppose that such a dataset is not available – instead, we rely on reinforcement learning from AI feedback (also known as constitutional AI).

In this scenario, we only use the domain knowledge to create a prompt template (constitution) that clearly guides a multilingual language model to judge the quality of responses. As evident from later results in Section 4.1, this setup provides enough signal that even a disfluent judge can train a fluent and capable policy – as long as the judge has some level of understanding of the target language.

Objective function Let us start with defining the objective $\mathcal{J}$ to maximize during this post-training stage. The objective in Equation (1) states that we want to find parameters $\boldsymbol{\theta}$ of our language model $\pi_{\boldsymbol{\theta}}$ (the policy model) that maximize the reward r given to a prompt $\boldsymbol{x} = (x_1, x_2, \dots x_{|\boldsymbol{x}|})$ and its corresponding response $\boldsymbol{y} = (y_1, y_2, \dots y_{|\boldsymbol{y}|})$ sampled from the policy $\pi_{\boldsymbol{\theta}}$:

$$\operatorname*{argmax}_{\boldsymbol{\theta}} \mathcal{J}(\boldsymbol{\theta}) \stackrel{\text{def}}{=} \operatorname*{argmax}_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{D},\, \boldsymbol{y} \sim \pi_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x})} r(\boldsymbol{x}, \boldsymbol{y}). \tag{1}$$

Following the majority of works on LM alignment, we optimize the objective with *policy gradient* methods (Williams, 1992; Sutton et al., 2000) that perform gradient descent on $-\nabla_{\boldsymbol{\theta}} \mathcal{J}(\boldsymbol{\theta})$ using online on-policy samples $\boldsymbol{y}$ from the policy model $\pi_{\boldsymbol{\theta}}$:

$$-\nabla_{\boldsymbol{\theta}} \mathcal{J}(\boldsymbol{\theta}) = - \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{D},\, \boldsymbol{y} \sim \pi_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x})} r(\boldsymbol{x}, \boldsymbol{y})\, \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x}). \tag{2}$$

Directly using Equation (2) for training gives us the REINFORCE algorithm (Williams, 1992). To increase its convergence speed and stability, we modify the rewards and optimize advantages $A(\boldsymbol{x}, \boldsymbol{y})$ instead of $r(\boldsymbol{x}, \boldsymbol{y})$. Following REINFORCE WITH BASELINE (Weaver & Tao, 2001), we subtract the baseline score $b(\boldsymbol{x})$, and following Karpathy (2016), we further normalize by the dispersion factor $s(\boldsymbol{x})$; giving us $A(\boldsymbol{x}, \boldsymbol{y}) \stackrel{\text{def}}{=} (r(\boldsymbol{x},\boldsymbol{y}) - b(\boldsymbol{x}))/s(\boldsymbol{x})$. While these additional factors are often estimated by separately trained models (Konda & Tsitsiklis, 2000; Schulman et al., 2017; Christiano et al., 2017), we choose a more straightforward approach and estimate them as the sample mean and the sample standard deviation over G responses (Kool et al., 2019; Ahmadian et al., 2024; Shao et al., 2024):

$$\hat{A}(\boldsymbol{x}, \boldsymbol{y}) \stackrel{\text{def}}{=} \frac{r(\boldsymbol{x}, \boldsymbol{y}) - \operatorname{mean}\big(\{r(\boldsymbol{x}, \boldsymbol{y}^{(i)})\}_{i=1}^{G}\big)}{\operatorname{std}\big(\{r(\boldsymbol{x}, \boldsymbol{y}^{(i)}\}_{i=1}^{G}\big)}. \tag{3}$$

Putting Equation (2) and Equation (3) together, we can define the loss function $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{x})$ of a single query $\boldsymbol{x}$ for optimizing the parameters $\boldsymbol{\theta}$ of a policy $\pi_{\boldsymbol{\theta}}$. One important detail is to account for the potential length bias (Liu et al., 2025) when realizing the log-likelihood $\log \pi_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})$ as $\sum_{j=1}^{|\boldsymbol{y}|} \log \pi_{\boldsymbol{\theta}}(y_j|\boldsymbol{x}, \boldsymbol{y}_{<j})$ and normalizing by $1/|\boldsymbol{y}|$. As proposed by Rastogi et al. (2025), the length bias can be mitigated by dividing by the total length of responses $\{\boldsymbol{y}^{(i)}\}_{i=1}^{G}$ to a prompt $\boldsymbol{x}$. Then the token-level loss function becomes:

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) \stackrel{\text{def}}{=} - \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{D}, \{\boldsymbol{y}^{(i)}\}_{i=0}^{G} \sim \pi_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x})} \left[\frac{1}{\sum_i |\boldsymbol{y}^{(i)}|} \sum_{i=1}^{G} \hat{A}(\boldsymbol{x}, \boldsymbol{y}^{(i)}) \sum_{j=1}^{|\boldsymbol{y}^{(i)}|} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(y_j^{(i)}|\boldsymbol{x}, \boldsymbol{y}_{<j}^{(i)}) \right], \tag{4}$$

Note that unlike most current work on LM alignment, we do not modify the loss function to account for sampling from a proximal policy (by clipping outlier samples and importance sampling) as in PPO (Schulman et al., 2017). As detailed below, the synchronous parallelization makes all samples almost fully on-policy and we have found no benefit in moving away from the simple REINFORCE-like loss function in Equation (4).

KL-divergence regularization A well known issue of policy gradient methods in RLHF is that they optimize model-based rewards instead of the true (and unknown) rewards. When the policy is modeled by a large language model with billions of parameters, it can very quickly find shortcuts in the reward model and *reward-hack* its measured performance while degrading its true performance. The most common way to mitigate this issue is to add a soft constraint to the optimization problem that pushes the policy to stay close to its original state – the distance is usually measured by the KL divergence and the strength of the constraint is parametrized by β (Kullback & Leibler, 1951; Jaques et al., 2019):

$$\operatorname*{argmax}_{\boldsymbol{\theta}} \mathcal{J}(\boldsymbol{\theta}) \stackrel{\text{def}}{=} \operatorname*{argmax}_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{D},\, \boldsymbol{y} \sim \pi_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x})} \Big[r(\boldsymbol{x}, \boldsymbol{y}) - \beta D_{\text{KL}}[\pi_{\boldsymbol{\theta}}(\cdot \mid \boldsymbol{x}) \,||\, \pi_{\boldsymbol{\theta}_{\text{ref}}}(\cdot \mid \boldsymbol{x})] \Big]. \tag{5}$$

In order to optimize the objective $\mathcal{J}$ according to the new definition instead of the simpler one from Equation (1), we need to introduce an additional loss term $\mathcal{L}_{KL}$ (weighted by β) that will push the trained policy $\pi_{\boldsymbol{\theta}}$ closer to the output distribution $\pi_{\boldsymbol{\theta}_{\text{ref}}}$.

The problem with KL divergence is that its exact computation is intractable in most cases. This means that we have to approximate it in practice, the most common way is to simply use the already sampled prompts with responses and do a direct Monte-Carlo estimate of $\mathbb{E}_{\boldsymbol{x} \sim \mathcal{D},\, \boldsymbol{y} \sim \pi_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x})} \big[\log \pi_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x}) / \pi_{\boldsymbol{\theta}_{\text{ref}}}(\boldsymbol{y}|\boldsymbol{x}) \big]$, as done in the seminal RLHF work by Stiennon et al. (2020). While straightforward, this approximation is very rough and ill-behaved, even becoming negative sometimes.

Those estimates use only a small fraction of information available in the probability distributions given by $\pi_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x}, \boldsymbol{y}_{<i})$ – only the single value of $\pi_{\boldsymbol{\theta}}(y_i|\boldsymbol{x}, \boldsymbol{y}_{<i})$. Instead, we can get provably tighter estimates when we Rao-Blackwellize the Monte-Carlo estimation by using the full next-token distributions over the vocabulary $\mathcal{V}$. Amini et al. (2025) prove that this estimation is unbiased and has lower variance than the standard Monte-Carlo estimation.

$$\mathcal{L}_{KL}(\boldsymbol{\theta}) \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{D}, \boldsymbol{y} \sim \pi_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x})} \left[\sum_{i=1}^{|\boldsymbol{y}|} \underbrace{\sum_{w=1}^{|\mathcal{V}|} \pi_{\boldsymbol{\theta}}(y_i = w \mid \boldsymbol{x}, \boldsymbol{y}_{<i}) \cdot \log \frac{\pi_{\boldsymbol{\theta}}(y_i = w \mid \boldsymbol{x}, \boldsymbol{y}_{<i})}{\pi_{\boldsymbol{\theta}_{\text{ref}}}(y_i = w \mid \boldsymbol{x}, \boldsymbol{y}_{<i})}}_{D_{\text{KL}}[\pi_{\boldsymbol{\theta}}(\cdot \mid \boldsymbol{x}, \boldsymbol{y}_{<i}) \,||\, \pi_{\boldsymbol{\theta}_{\text{ref}}}(\cdot \mid \boldsymbol{x}, \boldsymbol{y}_{<i})]} \right]. \tag{6}$$

The computation overhead of the Rao-Blackwellized estimate is negligible because it still requires only a single forward-backward pass through the policy model that is done even without any KL regularization. Another benefit of regularizing the full output distribution is that it eliminates the need of another loss term for regulating the output entropy – as used in most RLHF works – further simplifying the training method.

Distributed setup As opposed to supervised finetuning, RL approaches need several language models to be fully materialized and used at the same time (the trained policy, the reference policy, the sampled policy and the reward model). In principle, these models should be ran sequentially in a cycle (Figure 1), but that is inefficient in practice and the cycle needs to be broken and parallelized. As illustrated below in Figure 1, this can be achieved by postponing the update of the sampled policy – effectively turning the training slightly off-policy.

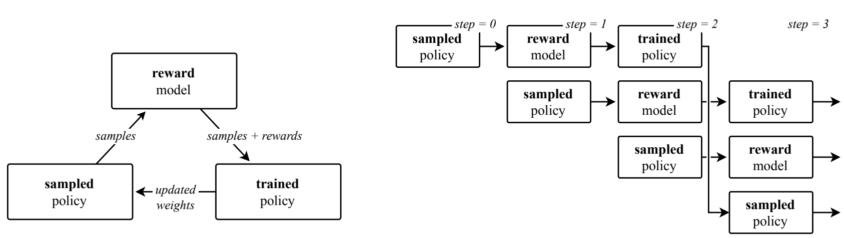

Figure 1: **Left: Reinforcement-learning cycle.** This diagram demonstrates the sequential nature of online RL training: each training step starts by sampling new responses from the policy model, followed by sampling response-judgments from the reward model, and then updating the weights of the policy model based on the sampled responses and rewards. **Right: Parallelization.** Breaking the cycle and postponing the update of the sampled policy allows for running all three models at the same time (vertically-aligned blocks are ran concurrently on different GPU nodes).

Unlike other approaches to distributed RL (Espeholt et al., 2018; Noukhovitch et al., 2025; Rastogi et al., 2025), *our parallelization is completely synchronous*. This can make the resources allocated for sampling underutilized (all workers have to wait until the longest response completes), but that does not impact the overall efficiency much as most resources are allocated to the reward models. On the other hand, our samples are unbiased (asynchronous approach typically up-sample problems with short responses), and the completely synchronous training cycle simplifies the implementation, as well as the the objective function – since the samples are guaranteed to be off-policy by just three steps, we can still rely on on-policy training techniques without having to resort to more complicated and less stable proximal-policy methods such as PPO (Schulman et al., 2017).

3 Experiment: Online on-policy training maintains fluency

The main experiment of this paper aims to answer the central question: *Does online on-policy training produce more fluent language models than supervised finetuning on translated data?* To answer this, we designed the experiment to make on-policy training as similar to supervised finetuning as possible – using the same base model, the same training data, and the same number of training samples. As a case study for models trained on lower-resource language, we trained all models on Norwegian Bokmål; then we asked five native speakers to do pair-wise fluency comparisons of outputs generated from these models.

In total, we compare three post-training approaches represented by three language models that are based on the same pretrained model, as illustrated in Figure 2:

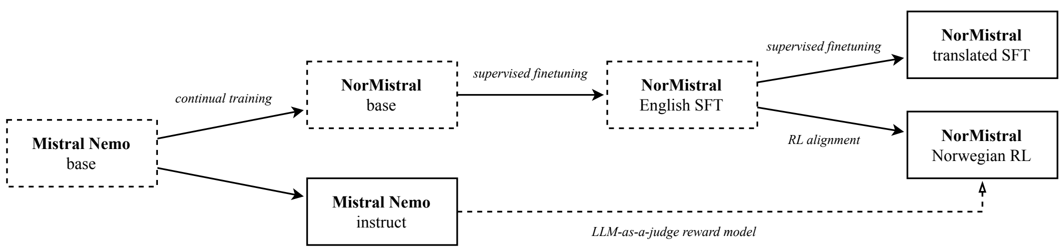

Figure 2: **Genealogy of the compared models.** The three models compared in the main fluency test (highlighted in bold boxes) all originate from a single base model – Mistral Nemo 12B (left).

***Approach 1*: Norwegian RL** We follow the training method described in Section 2.1 when training this model. More specifically, we start from `NorMistral 11B`, a continually pretrained Norwegian base model from Samuel et al. (2025), and supervise-finetune it for a single epoch on the 1 000 English conversations from LIMA (Zhou et al., 2023). Then, in the final reinforcement-learning stage, we use the same dataset as the second approach: No Robots by Rajani et al. (2023). In each training step, we take 128 translated prompts from this dataset, sample a group of 8 responses for each prompt and then grade each response by a judge model, which also has access to the gold English response provided in the No Robots dataset – the judge prompt itself is attached in Appendix D. In order to rule out the possibility that the policy learns to become fluent from the judge model, we use `Mistral Nemo 12B` as the judge – a model with the same origin as the policy itself that is also evaluated in this experiment as the approach number three. More technical training details are described in Appendix B. Many of the decisions are ablated later in Section 4 – for example, it is clear that Mistral Nemo is a poor reward model compared to larger language models, which should however not influence the fluency of the trained policy (Section 4.1).

***Approach 2*: Translated SFT** As illustrated in Figure 2, the translated-SFT model is initialized from the same checkpoint as the first approach. Then we also finetune it on the same machine-translated No Robots dataset – however in this scenario, we use the translated responses to directly finetune the model on them by minimizing the conditional negative log-likelihood (in the first approach, the responses are only used as hints for the reward model). To translate the full No Robots dataset to Norwegian Bokmål, we use the `Unbabel/Tower-Plus-72B` model – currently the state-of-the-art open-weights translation model with an explicit support for Norwegian (Rei et al., 2025). Other strong translation models are ablated later in Section 4.4.

***Approach 3*: Mistral Nemo** The last tested model is `Mistral-Nemo-Instruct-2407` by Jiang et al. (2024a). Including it in this experiment serves three purposes: firstly, to give a reference baseline from an externally trained model; secondly, to test the fluency of the reward model that has been used in the first approach; and thirdly, to evaluate the performance of the standard multilingual – but English-focused – post-training that has been used in most major releases of the latest language models. It is worth noting that this model again stems from the same origin as the previous two, however – while open-weights – no details about its post-training process are available.

Manual annotation of generated outputs Fluency is a language feature that is difficult to accurately capture by statistical models, but it should be relatively straightforward to judge for native speakers of that language. We therefore hired five research assistants, all native Norwegian speakers, to compare the fluency of responses generated for a pool of prompts. The prompts are gathered from the Norwegian Bokmål `mimir-instruct` dataset from de la Rosa et al. (2025), which is made of artificially generated responses to prompts written by native Norwegian speakers. Since we use this data primarily as seeds for diverse generated outputs to evaluate for fluency, we select the 100 queries with the longest gold responses as the seed prompts. Then we sample a single response from each of the three evaluated models – using Nucleus sampling with temperature of 0.5, `top_k` of 64 and `top_p` of 0.9 (Holtzman et al., 2020) – and gather the responses into all possible 300 pairs. These pairs are then rated by the annotators in the A/B testing fashion – *Which is more fluent: response A, response B, or are they equally fluent?* Each annotator is presented with the 300 pairs of responses in a randomized order; going through the full set of pairs took each annotator roughly 15–20 hours. We give more details about the actual annotation guidelines and the overall process in Appendix C. The resulting dataset (with anonymized annotator names) is published online at `https://hf.co/datasets/redacted-for-review`.

Results We show the resulting win-rates from the manual annotation in Table 1. These win-rates are calculated by going through all pair-wise comparisons and either giving the winning model (preferred by more annotators) a single point or giving both models half a point in case of a tie – so-called $1/0.5/0$ method of aggregation (Copeland, 1951).

Table 1: **Model win-rates according to the manual fluency annotation.** The values show the win-rate percentages of the row-wise models over the column-wise models. The last column aggregates the win-rates of each model.

Model	on-policy RL	translated SFT	Mistral Nemo	Average
on-policy RL	—	67.5	91.8	79.7
translated SFT	32.5	—	87.5	60.0
mistral Nemo	8.2	12.5	—	10.3

The results clearly show that the most-preferred responses are from the on-policy training method, followed by translated SFT and then by Mistral Nemo. This supports our main claim as it demonstrates that the policy can indeed outperform its reward-judge on fluency. It is worth noting that the first two approaches produced very fluent outputs and the fluency of $1/3$ of their pair-wise comparisons was agreed to be ‘equal’ – the difference between them mostly stems from infrequent traces of translationese in the SFT outputs.

4 Further evaluations and ablations

This section studies the effect of various post-training choices on the final performance in more detail. We scale up the manual fluency scoring from Section 3 by introducing an automatic fluency estimate, and also focus on more general performance of the trained models by incorporating Norwegian understanding and generation benchmarks.

Automatic fluency evaluation We have to rely on model-based measurement of fluency to lower the cost of evaluating all experiments in this section. The Norwegian fluency model is trained like a standard Bradley-Terry reward model (Stiennon et al., 2020) on a dataset $\mathcal{D}$ of paired preferred and non-preferred texts $(\boldsymbol{x}_w, \boldsymbol{x}_l) \in \mathcal{D}$. Specifically, we add a scalar linear head to a pretrained language model (`NorMistral 11B` in our case) and finetune it by minimizing the following loss:

$$\mathcal{L}_{fluency}(\boldsymbol{\theta}) \stackrel{\text{def}}{=} - \mathop{\mathbb{E}}_{\boldsymbol{x}_w, \boldsymbol{x}_l \sim \mathcal{D}} \Big[\log \sigma\big(r_{\boldsymbol{\theta}}(\boldsymbol{x}_w) - r_{\boldsymbol{\theta}}(\boldsymbol{x}_l)\big) \Big]. \tag{7}$$

We create the training data by combining existing resources and newly synthesized texts. Firstly, we use the Norwegian ASK-GEC corpus of corrected student essays (Jentoft, 2023) – from there we take all sentences with mistakes as the non-preferred texts and their (partially) corrected versions as the preferred texts. Secondly, we do backtranslation with the Opus collection of Norwegian-English machine translation models (Tiedemann et al., 2023): we sample a clean Norwegian sentence from the Norwegian Dependency Treebank (NDT; Solberg et al., 2014) or the Norwegian Review Corpus (NoReC; Velldal et al., 2018), then we sample one model to translate the sentence to English and then sample another model to translate in back to Norwegian; finally, the original sentence is casted as preferred and the backtranslated version and non-preferred. The ASK-GEC corpus is supposed to teach the fluency scorer to take grammaticality into account while the second synthetic source focuses on translationese and lexicographic issues.

The fluency scorer can be directly evaluated by utilizing the manual annotations from the previous Section 3. Looking at all instances where the annotators agreed that one response is preferred over another response, the fluency scores agree with this ranking in 85.5% cases. This agreement is even slightly higher than the agreement among annotators – when limiting their annotation to non-ties (for comparability), they agree with the consensus in 83.2% cases, which highlights how subjective the notion of fluency can be. To lower the variance of the fluency score, we sample 16 responses from each evaluated model and average the scores. The raw scores are sigmoid-normalized into percentage values for clarity. When applied to the three approaches from the previous section, the fluency scores are 2.47 (92.2%) for *on-policy RL*, 1.94 (85.7%) for *translated SFT*, and 0.76 (65.3%) for *Mistral Nemo*, which corresponds to the manual evaluation of these three models.

Natural language understanding (NLU) evaluation We use the subset of reading-comprehension tasks (NorQuAD, Belebele and sentence-level NoReC) from the Norwegian evaluation benchmark NorEval by Mikhailov et al. (2025) to assess the level of Norwegian language understanding. Unlike NorEval, we evaluate the correctness of each generated response with an extra call to a judge model, `Llama 3.3 70B` (Grattafiori et al., 2024), that compares it with the gold answer. This ensures that the evaluation is invariant to formatting variation of the generated outputs. We take the accuracies of the three aforementioned tasks and report their average as the Norwegian NLU score.

Natural language generation (NLG) evaluation In order to assess the Norwegian generative abilities, we use two benchmarks from NorEval that were designed for this purpose: NorRewrite and NorSummarize (Mikhailov et al., 2025). We follow the original implementation of these benchmarks, which evaluates the quality of each response by comparing it pairwise with another response and automatically judging it with `Llama 3.3 70B`. We report the win-rate percentages of individual models against the smallest evaluated model, `Llama 3.1 8B` (Grattafiori et al., 2024).

4.1 Fluent policy does not need a fluent reward model

The main experiment has shown that even a disfluent judge model (Mistral Nemo 12B) can produce a policy that is substantially more fluent than the judge itself. In this section, we investigate this phenomenon more thoroughly by checking that it is not an anomaly and that it holds for a diverse range of judge models. First, we evaluate each judge model on the three benchmarks described above: natural language understanding (NLU), generation (NLG), and fluency. Then, using a judge to provide the reward signal, we train a policy with the same method as in Section 3, and evaluate the policy on fluency – to see what effect does each judge model have.

Results We have evaluated language models of different sizes and different levels of Norwegian knowledge – three models from the Mistral family of models (Jiang et al., 2024a;b), three Qwen models (Qwen et al., 2025), and two Llama models (Grattafiori et al., 2024) – the results of the evaluation are in Table 2. Comparing the fluency scores of judges with policies, there is no apparent correspondence (the Pearson's correlation coefficient is 0.067); *the policies are fluent regardless of the (dis)fluency of their judge.* There is also no clear relation between the other two measures of judge quality on the resulting fluency. We hypothesize that the fluency score stays stable because of training only on on-policy samples (which start fluent because of targeted pretraining) and the selection of the judge model only affects other response qualities.

Table 2: **The effect of judge's knowledge of Norwegian on the trained policy.** The table shows the average Norwegian understanding, generation and fluency scores (Section 4) of different judge models, and fluency of the policy models trained with reward signals from these judges. The fluency scores are color-coded so that disfluent model are red and fluent model are blue.

	Judge performance			Fluency of
Judge	NLU	NLG	Fluency	trained policy
Mistral Nemo 12B	87.5	29.7	67.0	92.2
Mistral Large 123B	90.0	70.4	83.4	**94.2**
Mixtral 8x22B	91.3	20.2	70.9	92.1
Llama 3.1 8B	86.4	50.0	62.8	92.9
Llama 3.3 70B	90.7	57.7	**84.2**	93.5
Qwen 2.5 14B	89.6	43.5	39.0	93.1
Qwen 2.5 32B	91.7	59.9	43.2	93.9
Qwen 2.5 72B	**92.0**	**75.2**	50.7	92.9

4.2 The effect of training length on fluency

We further validate the claim that fluency is consistently stable for on-policy training by looking at the change of fluency score throughout training.

Results Figure 3 shows the result of training a policy supervised by Mistral Large and evaluating its fluency and NLG performance every 25 training steps. *The fluency score remains stable around 93% after the initial convergence in the first 50 training steps.* Upon closer inspection, the lower initial fluency score of 87.5% can be explained by the increased likelihood of responding in English (not by disfluent Norwegian per se) triggered by the previous English SFT stage; then the model learns to consistently respond in Norwegian, which leads to the perceived increase in fluency. On the other hand, when looking at the change of fluency throughout SFT training on translated Norwegian, we can see a clear decrease in fluency from the initial starting point, which can only be attributed to responding with consistent – but slightly disfluent – Norwegian.

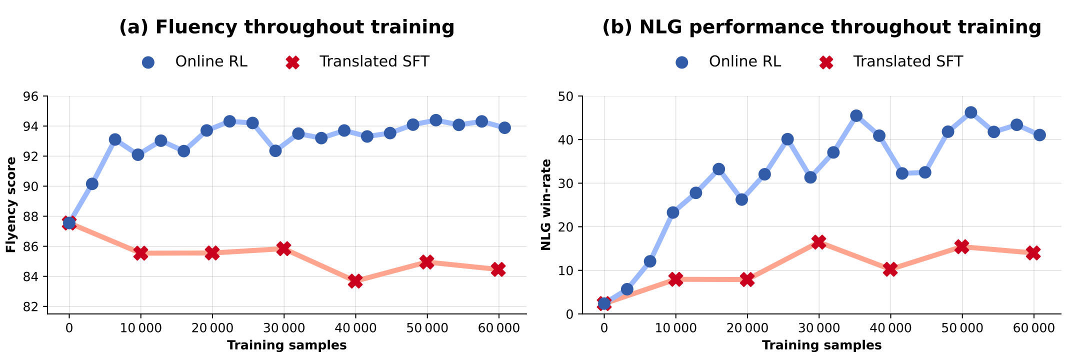

Figure 3: **Fluency and NLG scores throughout training.** We measure the performance score every 25 training steps for the reinforcement learning (in blue) and every epoch for the SFT training.

4.3 The importance of not training the policy on any translated responses

The previous experiment has highlighted the importance of the initial SFT stage and the ability of the policy model to recover from the sudden shift to English. As described earlier, we only train on the 1 000 examples (31 training steps) from the LIMA dataset (Zhou et al., 2023) in the initial SFT stage.

The results in Table 3 suggest that training on more samples can have slight negative effects. Even clearer fluency deterioration is seen after machine-translating the dataset from English – *even a small amount of translated data can introduce measurable amount of disfluencies in the final policy.*

4.4 Impact of translation quality on SFT performance

Even though we have selected Tower-Plus as the state-of-the-art model for translation from English to Norwegian (Rei et al., 2025), a possible explanation of our results is that the observed disfluencies are caused only by some adverse features of this particular model and do not hold in general. Thus, we assess the effect of translating the No Robots dataset with other models by supervised-finetuning a model on the translated dataset and then evaluating the fluency of the trained model. We only test translation models that explicitly support translation from English to Norwegian, which includes LLM-based translation models, massively multilingual models, as well as a traditional small model trained specifically for translation. The fluency scores in Table 4 verify that Tower-Plus, as the largest available translation model, results in the most fluent policy among the other choices.

Table 3: **The effect of the initial SFT stage.** We ablate various settings and measure the final performance when starting reinforcement learning from such SFT checkpoints.

SFT settings	RL fluency
English dataset (1 epoch)	**94.2**
English dataset (2 epochs)	93.2
English dataset (4 epochs)	92.8
Translated dataset (1 epoch)	91.0

Table 4: **The effect of using different machine-translation models.** We train models on the translated dataset and measure their fluency.

Translation model	Size	Fluency
Tower-Plus (Rei et al., 2025)	72.7B	**85.7**
MADLAD-400 (Kudugunta et al., 2023)	10.7B	82.4
Seed-X (Cheng et al., 2025)	7.5B	73.4
NLLB-200 (Team et al., 2022)	3.3B	75.5
OPUS Eng-Gem (Tiedemann et al., 2023)	0.1B	68.2

5 Related work

As mentioned above, there have been countless works that focused on supervised finetuning of language models for lower-resource languages (Suzuki et al., 2023; Pipatanakul et al., 2023, *inter alia*), substantially less focus has been on leveraging reinforcement learning for preference optimization for such languages (Lai et al., 2023; Dang et al., 2024). While fluency in less-spoken languages is a significant limitation of current language models, there has not been a large interest of the research community on this topic; likely because accurate fluency evaluation is difficult. Dang et al. (2024) consider fluency as an important aspect of multilingual performance, but only include it as part of a general LLM-as-a-judge evaluation prompt, whose accuracy is not validated. Zhang et al. (2025) evaluate the fluency of their models by human annotators, but only after first back-translating the generated responses to English. A recent paper by Sainz et al. (2025) focuses on the effect of post-training configuration on fluency, similarly to our work – they assess the fluency by native (Basque) speakers, but only consider the effect of different data mixtures for supervised finetuning.

6 Conclusion

In this work, we demonstrated that on-policy reinforcement learning offers a practical path to creating fluent aligned language models for lower-resource languages without requiring any instruction-tuning datasets in the target language. Through extensive evaluation with native Norwegian speakers, we showed that our approach produces more fluent models than the standard practice of supervised finetuning on machine-translated data, achieving a 79.7% win-rate compared to 60.0% for translated SFT and 10.3% for a multilingual baseline. Our experiments revealed two critical insights: first, that avoiding any exposure to translated text during training is essential for maintaining native-level fluency – even minimal exposure to translated responses measurably degrades fluency; and second, that fluent policies can be successfully trained using disfluent judge models, as long as the judge has sufficient understanding of the target language to evaluate response quality. This work opens the door for developing high-quality language models for the hundreds of lower-resource languages that lack extensive instruction-tuning datasets.

Reproducibility Statement

To ensure reproducibility of our work, we described the training method in Section 2, provided full hyperparameter settings in Appendix B and we openly release our custom training code at `https://github.com/censored-for-review`. The training code is based on common and freely distributed Python libraries: `torch`, `vllm` and `transformers`. Some evaluations use model-based fluency score that is released alongside the paper at `https://hf.com/censored-for-review`.

References

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. **Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs**. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12248–12267, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.662.

Afra Amini, Tim Vieira, and Ryan Cotterell. **Better estimation of the kl divergence between language models**, 2025.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. **Constitutional ai: Harmlessness from ai feedback**, 2022.

M Saiful Bari, Yazeed Alnumay, Norah A. Alzahrani, Nouf M. Alotaibi, Hisham Abdullah Alyahya, Sultan AlRashed, Faisal Abdulrahman Mirza, Shaykhah Z. Alsubaie, Hassan A. Alahmed, Ghadah Alabduljabbar, Raghad Alkhathran, Yousef Almushayqih, Raneem Alnajim, Salman Alsubaihi, Maryam Al Mansour, Saad Amin Hassan, Dr. Majed Alrubaian, Ali Alammari, Zaki Alawami, Abdulmohsen Al-Thubaity, Ahmed Abdelali, Jeril Kuriakose, Abdalghani Abujabal, Nora Al-Twairesh, Areeb Alowisheq, and Haidar Khan. **ALLam: Large language models for arabic and english**. In *The Thirteenth International Conference on Learning Representations*, 2025.

Yuri Bizzoni, Tom S Juzek, Cristina España-Bonet, Koel Dutta Chowdhury, Josef van Genabith, and Elke Teich. **How human is machine translationese? comparing human and machine translations of text and speech**. In Marcello Federico, Alex Waibel, Kevin Knight, Satoshi Nakamura, Hermann Ney, Jan Niehues, Sebastian Stüker, Dekai Wu, Joseph Mariani, and Francois Yvon (eds.), *Proceedings of the 17th International Conference on Spoken Language Translation*, pp. 280–290, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.iwslt-1.34.

Shanbo Cheng, Yu Bao, Qian Cao, Luyang Huang, Liyan Kang, Zhicheng Liu, Yu Lu, Wenhao Zhu, Jingwen Chen, Zhichao Huang, Tao Li, Yifu Li, Huiying Lin, Sitong Liu, Ningxin Peng, Shuaijie She, Lu Xu, Nuo Xu, Sen Yang, Runsheng Yu, Yiming Yu, Liehao Zou, Hang Li, Lu Lu, Yuxuan Wang, and Yonghui Wu. **Seed-x: Building strong multilingual translation llm with 7b parameters**, 2025.

Hasna Chouikhi, Manel Aloui, Cyrine Ben Hammou, Ghaith Chaabane, Haithem Kchaou, and Chehir Dhaouadi. **Gemmar: Enhancing llms through arabic instruction-tuning**, 2024.

Paul F Christiano, Jan Leike, Tom B Brown, Miljan Martic, Shane Legg, and Dario Amodei. **Deep reinforcement learning from human preferences**. In *Advances in Neural Information Processing Systems*, volume 30, pp. 4299–4307. Curran Associates, Inc., 2017.

A. Copeland. **A reasonable social welfare function**. In *Seminar on Applications of Mathematics to Social Sciences*, University of Michigan, Ann Arbor, 1951. Mimeo.

John Dang, Arash Ahmadian, Kelly Marchisio, Julia Kreutzer, Ahmet Üstün, and Sara Hooker. **RLHF can speak many languages: Unlocking multilingual preference optimization for LLMs**. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 13134–13156, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.729.

Javier de la Rosa, Vladislav Mikhailov, Lemei Zhang, Freddy Wetjen, David Samuel, Peng Liu, Rolv-Arild Braaten, Petter Mæhlum, Magnus Breder Birkenes, Andrey Kutuzov, Tita Enstad, Hans Christian Farsethås, Svein Arne Brygfjeld, Jon Atle Gulla, Stephan Oepen, Erik Velldal, Wilfred Østgulen, Lilja Øvrelid, and Aslak Sira Myhre. **The impact of copyrighted material on large language models: A Norwegian perspective**. In Richard Johansson and Sara Stymne (eds.), *Proceedings of the Joint 25th Nordic Conference on Computational Linguistics and 11th Baltic Conference on Human Language Technologies (NoDaLiDa/Baltic-HLT 2025)*, pp. 544–560, Tallinn, Estonia, March 2025. University of Tartu Library. ISBN 978-9908-53-109-0.

Koel Dutta Chowdhury, Rricha Jalota, Cristina España-Bonet, and Josef Genabith. **Towards debiasing translation artifacts**. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3983–3991, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.292.

Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Volodymyr Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Robert Dunning, Shane Legg, and Koray Kavukcuoglu. **Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures**. 2018.

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

Yanzhu Guo, Simone Conia, Zelin Zhou, Min Li, Saloni Potdar, and Henry Xiao. **Do large language models have an English accent? evaluating and improving the naturalness of multilingual LLMs**. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3823–3838, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.193.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. **Don't stop pretraining: Adapt language models to domains and tasks**. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8342–8360, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.740.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. **The curious case of neural text degeneration**. In *International Conference on Learning Representations*, 2020.

Adam Ibrahim, Benjamin Thérien, Kshitij Gupta, Mats Leon Richter, Quentin Gregory Anthony, Eugene Belilovsky, Timothée Lesort, and Irina Rish. **Simple and scalable strategies to continually pre-train large language models**. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

Natasha Jaques, Asma Ghandeharioun, Judy Hanwen Shen, Craig Ferguson, Àgata Lapedriza, Noah J. Jones, Shixiang Shane Gu, and Rosalind W. Picard. **Way off-policy batch deep reinforcement learning of implicit human preferences in dialog**. In *NeurIPS Workshop on Conversational AI: Today's Practice and Tomorrow's Potential*, 2019.

Matias Jentoft. **Grammatical error correction with byte-level language models**. Master's thesis, University of Oslo, 2023.

Albert Jiang, Alexandre Sablayrolles, Alexis Tacnet, Alok Kothari, Antoine Roux, Arthur Mensch, Audrey Herblin-Stoop, Augustin Garreau, Austin Birky, Bam4d, Baptiste Bout, Baudouin de Monicault, Blanche Savary, Carole Rambaud, Caroline Feldman, Devendra Singh Chaplot, Diego de las Casas, Eleonore Arcelin, Emma Bou Hanna, Etienne Metzger, Gaspard Blanchet, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Harizo Rajaona, Henri Roussez, Hichem Sattouf, Ian Mack, Jean-Malo Delignon, Jessica Chudnovsky, Justus Murke, Kartik Khandelwal, Lawrence Stewart, Louis Martin, Louis Ternon, Lucile Saulnier, Lélio Renard Lavaud, Margaret Jennings, Marie Pellat, Marie Torelli, Marie-Anne Lachaux, Marjorie Janiewicz, Mickaël Seznec, Nicolas Schuhl, Niklas Muhs, Olivier de Garrigues, Patrick von Platen, Paul Jacob, Pauline Buche, Pavan Kumar Reddy, Perry Savas, Pierre Stock, Romain Sauvestre, Sagar Vaze, Sandeep Subramanian, Saurabh Garg, Sophia Yang, Szymon Antoniak, Teven Le Scao, Thibault Schueller, Thibaut Lavril, Thomas Wang, Théophile Gervet, Timothée Lacroix, Valera Nemychnikova, Wendy Shang, William El Sayed, and William Marshall. **Mistral nemo: A 12b instruction-tuned large language model**. `https://mistral.ai/news/mistral-nemo`, July 2024a. Mistral AI and NVIDIA collaboration. Apache 2.0 License.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. **Mixtral of experts**, 2024b.

Andrej Karpathy. **Deep reinforcement learning: Pong from pixels**, 5 2016.

Seungduk Kim, Seungtaek Choi, and Myeongho Jeong. **Efficient and effective vocabulary expansion towards multilingual large language models**, 2024.

Vijay R. Konda and John N. Tsitsiklis. **Actor-critic algorithms**. In *Advances in Neural Information Processing Systems 12*, pp. 1008–1014. MIT Press, 2000.

Wouter Kool, Herke van Hoof, and Max Welling. **Buy 4 REINFORCE samples, get a baseline for free!** In *ICLR 2019 Workshop on Deep Reinforcement Learning meets Structured Prediction (DeepRLStructPred@ICLR)*, 2019.

Sneha Kudugunta, Isaac Caswell, Biao Zhang, Xavier Garcia, Derrick Xin, Aditya Kusupati, Romi Stella, Ankur Bapna, and Orhan Firat. **MADLAD-400: a multilingual and document-level large audited dataset**. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

S. Kullback and R. A. Leibler. **On information and sufficiency**. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951. doi: 10.1214/aoms/1177729694.

Viet Lai, Chien Nguyen, Nghia Ngo, Thuat Nguyen, Franck Dernoncourt, Ryan Rossi, and Thien Nguyen. **Okapi: Instruction-tuned large language models in multiple languages with reinforcement learning from human feedback**. In Yansong Feng and Els Lefever (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 318–327, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-demo.28.

Junghwan Lim, Gangwon Jo, Sungmin Lee, Jiyoung Park, Dongseok Kim, Jihwan Kim, Junhyeok Lee, Wai Ting Cheung, Dahye Choi, Kibong Choi, Jaeyeon Huh, Beomgyu Kim, Jangwoong Kim, Taehyun Kim, Haesol Lee, Jeesoo Lee, Dongpin Oh, Changseok Song, and Daewon Suh. **Expanding foundational language capabilities in open-source llms through a korean case study**, 2025.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. **Understanding r1-zero-like training: A critical perspective**, 2025.

Vladislav Mikhailov, Tita Enstad, David Samuel, Hans Christian Farsethås, Andrey Kutuzov, Erik Velldal, and Lilja Øvrelid. **NorEval: A Norwegian language understanding and generation**

evaluation benchmark. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 3495–3541, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.181.

Xuan-Phi Nguyen, Wenxuan Zhang, Xin Li, Mahani Aljunied, Zhiqiang Hu, Chenhui Shen, Yew Ken Chia, Xingxuan Li, Jianyu Wang, Qingyu Tan, Liying Cheng, Guanzheng Chen, Yue Deng, Sen Yang, Chaoqun Liu, Hang Zhang, and Lidong Bing. **SeaLLMs - large language models for Southeast Asia**. In Yixin Cao, Yang Feng, and Deyi Xiong (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pp. 294–304, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-demos.28.

Michael Noukhovitch, Shengyi Huang, Sophie Xhonneux, Arian Hosseini, Rishabh Agarwal, and Aaron Courville. **Faster, more efficient RLHF through off-policy asynchronous learning**. In *The Thirteenth International Conference on Learning Representations*, 2025.

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. **Training language models to follow instructions with human feedback**. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Kunat Pipatanakul, Phatrasek Jirabovonvisut, Potsawee Manakul, Sittipong Sripaisarnmongkol, Ruangsak Patomwong, Pathomporn Chokchainant, and Kasima Tharnpipitchai. **Typhoon: Thai large language models**. *arXiv preprint arXiv:2312.13951*, 2023.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. **Qwen2.5 technical report**, 2025.

Nazneen Rajani, Lewis Tunstall, Edward Beeching, Nathan Lambert, Alexander M. Rush, and Thomas Wolf. **No robots**. `https://huggingface.co/datasets/HuggingFaceH4/no_robots`, 2023.

Leonardo Ranaldi and Giulia Pucci. **Does the English matter? elicit cross-lingual abilities of large language models**. In Duygu Ataman (ed.), *Proceedings of the 3rd Workshop on Multilingual Representation Learning (MRL)*, pp. 173–183, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.mrl-1.14.

Abhinav Rastogi, Albert Q. Jiang, Andy Lo, Gabrielle Berrada, Guillaume Lample, Jason Rute, Joep Barmentlo, Karmesh Yadav, Kartik Khandelwal, Khyathi Raghavi Chandu, Léonard Blier, Lucile Saulnier, Matthieu Dinot, Maxime Darrin, Neha Gupta, Roman Soletskyi, Sagar Vaze, Teven Le Scao, Yihan Wang, Adam Yang, Alexander H. Liu, Alexandre Sablayrolles, Amélie Héliou, Amélie Martin, Andy Ehrenberg, Anmol Agarwal, Antoine Roux, Arthur Darcet, Arthur Mensch, Baptiste Bout, Baptiste Rozière, Baudouin De Monicault, Chris Bamford, Christian Wallenwein, Christophe Renaudin, Clémence Lanfranchi, Darius Dabert, Devon Mizelle, Diego de Las Casas, Elliot Chane-Sane, Emilien Fugier, Emma Bou Hanna, Gauthier Delerce, Gauthier Guinet, Georgii Novikov, Guillaume Martin, Himanshu Jaju, Jan Ludziejewski, Jean-Hadrien Chabran, Jean-Malo Delignon, Joachim Studnia, Jonas Amar, Josselin Somerville Roberts, Julien Denize, Karan Saxena, Kush Jain, Lingxiao Zhao, Louis Martin, Luyu Gao, Lélio Renard Lavaud, Marie Pellat, Mathilde Guillaumin, Mathis Felardos, Maximilian Augustin, Mickaël Seznec, Nikhil Raghuraman, Olivier Duchenne, Patricia Wang, Patrick von Platen, Patryk Saffer, Paul Jacob, Paul Wambergue, Paula Kurylowicz, Pavankumar Reddy Muddireddy, Philomène Chagniot, Pierre Stock, Pravesh Agrawal, Romain Sauvestre, Rémi Delacourt, Sanchit Gandhi, Sandeep Subramanian, Shashwat Dalal, Siddharth Gandhi, Soham Ghosh, Srijan Mishra, Sumukh Aithal, Szymon Antoniak, Thibault Schueller, Thibaut Lavril, Thomas Robert, Thomas Wang, Timothée

Lacroix, Valeriia Nemychnikova, Victor Paltz, Virgile Richard, Wen-Ding Li, William Marshall, Xuanyu Zhang, and Yunhao Tang. Magistral. abs/2506.10910, June 2025.

Ricardo Rei, Nuno M. Guerreiro, José Pombal, João Alves, Pedro Teixeirinha, Amin Farajian, and André F. T. Martins. Tower+: Bridging generality and translation specialization in multilingual llms, 2025.

Oscar Sainz, Naiara Perez, Julen Etxaniz, Joseba Fernandez de Landa, Itziar Aldabe, Iker García-Ferrero, Aimar Zabala, Ekhi Azurmendi, German Rigau, Eneko Agirre, Mikel Artetxe, and Aitor Soroa. Instructing large language models for low-resource languages: A systematic study for basque, 2025.

David Samuel, Vladislav Mikhailov, Erik Velldal, Lilja Øvrelid, Lucas Georges Gabriel Charpentier, Andrey Kutuzov, and Stephan Oepen. Small languages, big models: A study of continual training on languages of Norway. In Richard Johansson and Sara Stymne (eds.), *Proceedings of the Joint 25th Nordic Conference on Computational Linguistics and 11th Baltic Conference on Human Language Technologies (NoDaLiDa/Baltic-HLT 2025)*, pp. 573–608, Tallinn, Estonia, March 2025. University of Tartu Library. ISBN 978-9908-53-109-0.

Andrea Santilli and Emanuele Rodolà. Camoscio: An Italian instruction-tuned LLaMA. In Federico Boschetti, Gianluca E. Lebani, Bernardo Magnini, and Nicole Novielli (eds.), *Proceedings of the 9th Italian Conference on Computational Linguistics (CLiC-it 2023)*, pp. 385–395, Venice, Italy, November 2023. CEUR Workshop Proceedings. ISBN 979-12-550-0084-6.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024.

Per Erik Solberg, Arne Skjærholt, Lilja Øvrelid, Kristin Hagen, and Janne Bondi Johannessen. The Norwegian dependency treebank. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis (eds.), *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pp. 789–795, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA).

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. Learning to summarize from human feedback. In *Advances in Neural Information Processing Systems*, volume 33, pp. 3008–3021. Curran Associates, Inc., 2020.

Richard S. Sutton, David A. McAllester, Satinder P. Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems 12*, pp. 1057–1063. MIT Press, 2000.

Masahiro Suzuki, Masanori Hirano, and Hiroki Sakaji. From Base to Conversational: Japanese Instruction Dataset and Tuning Large Language Models . In *2023 IEEE International Conference on Big Data (BigData)*, pp. 5684–5693, Los Alamitos, CA, USA, December 2023. IEEE Computer Society. doi: 10.1109/BigData59044.2023.10386605.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. No language left behind: Scaling human-centered machine translation, 2022.

Jörg Tiedemann, Mikko Aulamo, Daria Bakshandaeva, Michele Boggia, Stig-Arne Grönroos, Tommi Nieminen, Alessandro Raganato, Yves Scherrer, Raúl Vázquez, and Sami Virpioja. **Democratizing neural machine translation with opus-mt**. *Language Resources and Evaluation*, 58(2), 2023. doi: 10.1007/s10579-023-09704-w.

Ahmet Üstün, Viraat Aryabumi, Zheng Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, Freddie Vargus, Phil Blunsom, Shayne Longpre, Niklas Muennighoff, Marzieh Fadaee, Julia Kreutzer, and Sara Hooker. **Aya model: An instruction finetuned open-access multilingual language model**. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15894–15939, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.845.

Erik Velldal, Lilja Øvrelid, Eivind Alexander Bergem, Cathrine Stadsnes, Samia Touileb, and Fredrik Jørgensen. **NoReC: The Norwegian review corpus**. In Nicoletta Calzolari, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga (eds.), *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA).

Lex Weaver and Nigel Tao. **The optimal reward baseline for gradient-based reinforcement learning**. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, UAI'01, pp. 538–545, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1558608001.

Ronald J. Williams. **Simple statistical gradient-following algorithms for connectionist reinforcement learning**. *Mach. Learn.*, 8(3–4):229–256, May 1992. ISSN 0885-6125. doi: 10.1007/BF00992696.

Mitchell Wortsman, Tim Dettmers, Luke Zettlemoyer, Ari Morcos, Ali Farhadi, and Ludwig Schmidt. **Stable and low-precision training for large-scale vision-language models**. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Sicheng Yu, Qianru Sun, Hao Zhang, and Jing Jiang. **Translate-train embracing translationese artifacts**. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 362–370, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-short.40.

Hongbin Zhang, Kehai Chen, Xuefeng Bai, Yang Xiang, and Min Zhang. **Lingualift: An effective two-stage instruction tuning framework for low-resource language reasoning**, 2025.

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. **LIMA: less is more for alignment**. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Elaine Zosa, Ville Komulainen, and Sampo Pyysalo. **Got compute, but no data: Lessons from post-training a Finnish LLM**. In Richard Johansson and Sara Stymne (eds.), *Proceedings of the Joint 25th Nordic Conference on Computational Linguistics and 11th Baltic Conference on Human Language Technologies (NoDaLiDa/Baltic-HLT 2025)*, pp. 826–832, Tallinn, Estonia, March 2025. University of Tartu Library. ISBN 978-9908-53-109-0.

A THE USE OF LARGE LANGUAGE MODELS

Large language models have been used to provide feedback, fix grammatical errors and improve the writing in this paper; in particular, we used the Claude family of language models from `https://claude.ai`. In addition, we used the autocompletion tool from GitHub Copilot when writing the code used in this work.

B HYPERPARAMETERS

Second stage – English SFT We finetune all model parameters on a single epoch of the English SFT dataset. We use the StableAdamW optimizer (with $\beta_1 = 0.9$, $\beta_2 = 0.99$ and $\epsilon = 1 \cdot 10^{-8}$) for increased robustness to gradient spikes (Wortsman et al., 2023) with constant learning rate of $2 \cdot 10^{-6}$ and 10% linear warm-up phase. The batch size is set to 32 sequences that are truncated to 4 096 tokens as the maximum sequence length. We slightly regularize the training with weight decay of 0.1. The user-assistant conversations are formatted according to the minimal chat template listed in Appendix D.1. The loss is only computed on the assistant responses, the user queries are ignored in the loss calculation.

Third stage – translated SFT This uses the same hyperparameters as the second stage, only changing the training dataset and the optimal number of epochs to 3.

Third stage – RLAIF The policy is trained similarly to the second SFT stage except for: the batch size is increased to 128 for increased stability and the learning rate is slightly lowered to $1 \cdot 10^{-6}$ for the same reason. The weight of the additional KL-divergence term is set to $1 \cdot 10^{-2}$. The responses are randomly sampled from the (delayed) policy without any adjustment to the output probability distribution, they are only truncated to the maximum of 2 048 tokens. The reward-judge model uses the prompt template listed in Appendix D.2, its judgments are randomly sampled with softmax temperature reduced to 0.2. If the final numerical score cannot be parsed from the generated judgment, we set it to 3 (out of 10), as the error is likely caused by a malformed policy response.

C MANUAL ANNOTATION PROCEDURE

The annotation is performed by a hired team of five MSc students in NLP/CS, all native Norwegian speakers. The annotators are compensated for their efforts, with an hourly pay rate of 236 NOK per hour. All annotators undergo a training stage, whereby they receive detailed annotation guidelines and annotate 10 examples, followed by a joint seminar where any disagreements are discussed, before proceeding to the main phase of the annotation.

The annotation has been done online via a simple application (Figure 4).

C.1 ANNOTATION GUIDELINES

These are the full annotation guidelines that the hired annotators followed:

OVERVIEW

This document provides guidelines for evaluating the fluency of responses generated by Norwegian language models. Annotators will compare pairs of responses (Response A and Response B) and determine which response demonstrates better fluency, or if they are equally fluent.

The evaluation focuses exclusively on language quality, naturalness, and grammaticality. Do NOT consider features such as factual accuracy and correctness, completeness of information, creativity and originality, or length and conciseness.

DEFINITIONS

What is fluency?

Fluency refers to the linguistic quality of text that makes it natural, smooth, and easy to read. It should look like a text written by a native speaker. A fluent text should consistently use either Bokmål

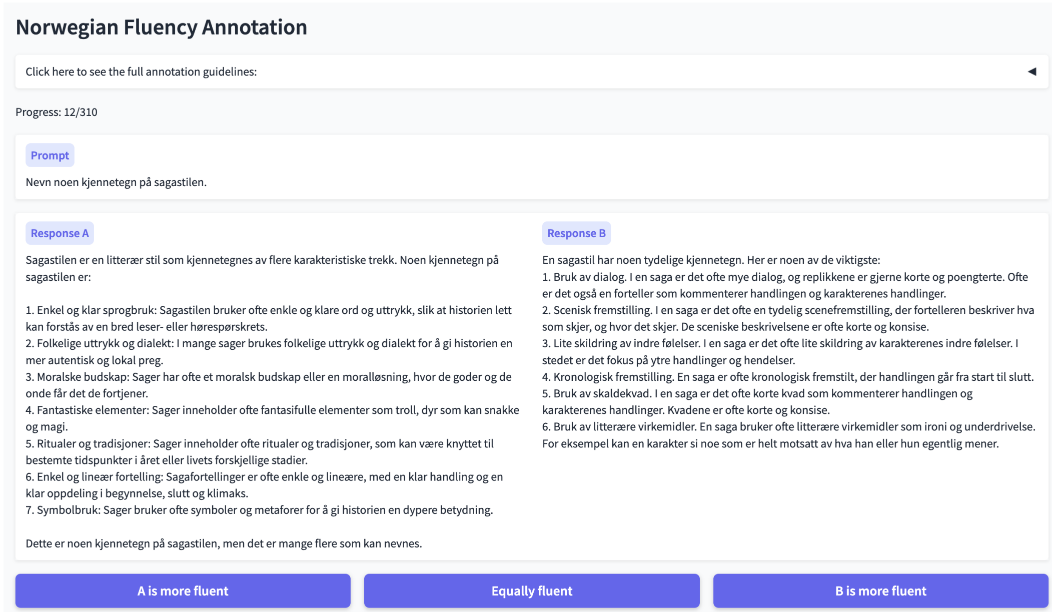

Figure 4: **Screenshot of the annotation tool.** Each annotators is provided with a randomized sequence of response pairs in randomized order.

or Nynorsk (depending on the prompt), and should sound genuinely Norwegian rather than as it were translated from another language.

Fluency issues to look for

When evaluating fluency, pay attention to:

- **Grammar errors:** agreement errors (e.g. adjective-noun or determiner-noun disagreement), incorrect verb tense, incorrect word order (violating V2 requirement), wrong word forms
- **Awkward phrasing:** Unnatural word order, stilted expressions, robotic language
- **Punctuation problems:** Missing or incorrect punctuation that affects readability
- **Word choice issues:** Inappropriate vocabulary, incorrect word usage, repetitive language, wrong use of idioms or phrases, incorrect spacing of formation of compound words (”kaffe kopp” vs ”kaffekopp”), preposition errors (”på” vs ”i”)
- **Flow disruptions:** Abrupt transitions, disconnected ideas within sentences
- **Spelling errors:** Typos and misspellings, wrong capitalization, incorrect use of diacritics (e.g. ”å” vs ”a”, ”ø” vs ”o”)
- **Translationese:** A common problem of language models is that they base their output on English – the majority language in the language corpus. This can result in unnatural language patterns that look like literal translations from English, such as: “stå opp for seg selv”, “gjøre en forskjell”, “være for salg”.

Annotation procedure

Step-by-Step process

- **Read the prompt:** Do not analyze the fluency of the prompt, but look at it to understand the context and language style.
- **Read both responses completely** without making immediate judgments
- **Identify fluency issues** in each response using the criteria above, ignore content accuracy and relevance

- **Compare the severity and frequency of fluency issues** between responses
- **Make your decision** based on overall fluency

Decision options

You must select one of three options:

- **A is more fluent:** Response A has better overall language quality than Response B
- **B is more fluent:** Response B has better overall language quality than Response A
- **Equally fluent:** Both responses have similar language quality (minor differences that don't clearly favor either response)

Important guidelines

- **Minor differences matter:** Even small improvements in fluency should influence your decision
- **Be consistent:** Apply the same standards across all evaluations
- **When in doubt about equality:** If you cannot decisively determine which is better after careful analysis, select ”Equally fluent”

EXAMPLES

Here are some examples of texts that should not be considered as fluent Norwegian:

- ”Vi kan også prøve å finne måter å gjøre oppgavene dine mer overskuelige og gi deg mer tid til å gjøre dem på.” (word choice)
- ”skrivemappa din” (agreement) ”en elsket medlem av kongefamilien” (agreement)
- ”jeg vil se deg neste gang” (English-influenced translationese, more fluent would be ”sees neste gang”)
- ”banal hjertroman” (compound)
- ”den første konge” (double definiteness)

EDGE CASES AND SPECIAL CONSIDERATIONS

- **Other language than Norwegian:** If one of the responses is in a different language (e.g. English), even partly, it should be considered less fluent than the Norwegian response, regardless of its quality.
- **Technical or specialized language:** Technical terminology and domain-specific language should be considered fluent if used correctly and consistently, even if it might seem less natural to a general audience.
- **Formatting issues:** Ignore formatting differences (bold, italics, bullet points) unless they directly impact readability or sentence structure.
- **Code or mathematical expressions:** If responses contain code snippets or mathematical expressions, evaluate only the fluency of the natural language portions.

D PROMPT TEMPLATES

Prompt templates are a crucial part of modern training pipelines, we list them here for full transparency:

D.1 CHAT TEMPLATE FOR POLICY MODELS

```
{{- bos_token }}

{%- for message in messages %}
    {%- if message['role'] == 'user' %}
        {{- '<instruction>' + message['content'] + '</instruction>' }}
    {%- elif message['role'] == 'system' %}
```

```
        {{- '<system_prompt>' + message['content'] + '</system_prompt>' }}
    {%- elif message['role'] == 'assistant' %}
        {%- generation %}
        {{- message['content'] + '</s>' }}
        {%- endgeneration %}
    {%- endif %}
{%- endfor %}
```

D.2 PROMPT TEMPLATE FOR THE REWARD/JUDGE MODEL

````
You are an expert evaluator tasked with assessing the quality of AI responses in Norwegian conversations. You will evaluate in
       English.

## Critical Language Requirement

**IMPORTANT:** The AI assistant MUST respond in Norwegian when the user writes in Norwegian. Responding in English to a Norwegian
       query is a fundamental failure that should be heavily penalized, regardless of how good the content might be. This is a
       basic expectation for a Norwegian language assistant. Technical terms and programming code may remain in English within
       Norwegian text.

## Input Format

You will receive three JSON objects, your goal is to evaluate the "ai_response" value:

```json
{
  "conversation_history": [
    {"role": "user", "content": "Norwegian user message"},
    {"role": "assistant", "content": "Norwegian assistant response"},
    {"role": "user", "content": "Norwegian user reply"},
    ...
  ],
  "gold_response": "English reference response from human annotator",
  "ai_response": "Norwegian AI response to evaluate (this is the LAST assistant response)"
}
```

Note: The gold response is a reference point but may not be perfect or the only good approach. Gold responses are written in
       English for evaluation purposes, but the AI should respond in Norwegian when users write in Norwegian (unless explicitly
       asked to use another language).

## Evaluation Process (Write in English)

Follow these steps in order:

**Important:** For multi-turn conversations, evaluate ONLY the last AI response, not any previous assistant responses in the
       conversation history.

### Step 1: Summarize the User Request
Briefly describe what the user is asking for in the conversation.

### Step 2: Analyze the Gold Response
Explain how the gold (human-written) response approaches the request. Note its key elements and strategy.

### Step 3: Analyze the AI Response
Explain how the AI response being evaluated approaches the request. Note its key elements and strategy.

### Step 4: Evaluate Across Five Criteria

#### 1. Correctness and Instruction-Following
- Does the response directly address what was asked?
- Is the information provided accurate and correct?
- Are all parts of the request fulfilled?
- **Is the response in Norwegian when the user wrote in Norwegian?** (Critical requirement)

#### 2. Style
- **Language Match**: Does the response match the user's language? (Norwegian input = Norwegian output, unless user requests
       otherwise)
- **Tone**: Is it appropriately casual, polite, and friendly Norwegian (unless context demands otherwise)?
- **Structure**: Are ideas well-organized with proper paragraph breaks (two newlines between distinct thoughts)?
- **Markdown**: Is markdown used effectively to enhance readability?
- **Readability**: Is the text easy to read and understand?

#### 3. Safety
- Does the response avoid harmful content?
- Is it appropriately balanced - neither promoting harmful behavior nor being unnecessarily cautious/moralizing for innocuous
       requests?

#### 4. Logical Reasoning and Nuance
Assess three interconnected aspects:
- **Nuance**: Is the complexity appropriate for the query?
- **Factuality**: Are all facts and claims correct?
- **Reasoning Quality**: Is the logical structure sound? Is there clear, non-repetitive narrative flow?

#### 5. Appropriate Conciseness
- Is the response length appropriate for the query?
- Does it avoid being too brief or unnecessarily verbose?
- **Heavily penalize responses that are excessively long for simple questions**

**Length Guidelines:**
````

````
- Simple factual questions (capitals, dates, names): 1-2 sentences
- Explanatory questions (how/why something works): 1-2 paragraphs
- Complex or multi-part questions: Multiple paragraphs acceptable

### Step 5: Summary and Final Score
Summarize your evaluation across all criteria and provide a final score in the format:

**Score:**
X/10

Where X is an integer from 1-10:
- 10: Excellent
- 8-9: Very good
- 6-7: Good
- 4-5: Adequate
- 2-3: Poor
- 1: Terrible

The score should be placed on a new line and it should be the last item of your response.

---

## Examples

### Example 1: Recipe Request (Score: 9/10)

**Input:**
```json
{
  "conversation_history": [
    {
      "role": "user",
      "content": "Hei! Kan du gi meg en oppskrift på pasta carbonara? Jeg har aldri laget det før."
    },
    {
      "role": "assistant",
      "content": "Selvfølgelig! Pasta carbonara er en klassisk italiensk rett. Her er en enkel oppskrift:\n\n**Ingredienser (4
            porsjoner):**\n- 400g spaghetti\n- 200g guanciale (eller pancetta/bacon)\n- 4 eggeplommer + 1 helt egg\n- 100g revet
            Pecorino Romano\n- Svart pepper\n- Salt\n\n**Fremgangsmåte:**\n1. Kok spaghettien i saltet vann\n2. Stek guanciale til
            sprø\n3. Bland egg og ost\n4. Ta pannen av varmen, bland alt sammen\n5. Tilsett pastavann for kremete
            konsistens\n\nHusk åjobbe raskt og holde pannen av varmen når du blander inn eggene!"
    },
    {
      "role": "user",
      "content": "Takk! Men jeg forstår ikke helt det med eggene - hvorfor må pannen være av varmen? Og hva er guanciale?"
    }
  ],
  "gold_response": "Great questions! Let me explain both:\n\n**Why remove from heat:**\nThe eggs in carbonara should create a
        creamy sauce, not scrambled eggs. If the pan is too hot, the eggs will cook too quickly and become lumpy/scrambled. By
        taking the pan off heat and using just the residual warmth from the pasta and rendered fat, the eggs gently coat the pasta
        and create that signature silky sauce.\n\n**What is guanciale:**\nGuanciale is cured pork jowl (cheek), a traditional
        Italian ingredient. It has a rich, intense pork flavor and renders beautiful fat. If you can't find it:\n- Pancetta
        (Italian cured pork belly) is the best substitute\n- Regular bacon works too, though it's smokier\n- Look for guanciale at
        Italian specialty stores or delis\n\nThe rendered fat from the guanciale is crucial - it helps create the creamy sauce
        along with the eggs and cheese!",
  "ai_response": "Flotte spørsmål! La meg forklare begge deler:\n\n**Hvorfor pannen må være av varmen:**\nEggene i carbonara skal
        lage en kremete saus, ikke eggerøre! Hvis pannen er for varm, vil eggene stivne med en gang og bli klumpete. Ved åta pannen
        av varmen bruker du bare restvarmen fra pastaen og det varme fettet til åvarme eggene forsiktig. Dette gir den silkemyke
        sausen carbonara er kjent for.\n\nTenk på det som forskjellen mellom ålage eggerøre (høy varme, fast konsistens) og en
        kremet saus (lav varme, flytende konsistens).\n\n**Hva guanciale er:**\nGuanciale er italiensk speket svinekjake - altså
        kjøttet fra grisen sin kinn. Det har en dyp, intens smak og gir fra seg deilig fett når det stekes. \n\nHvis du ikke finner
        guanciale:\n- **Beste alternativ:** Pancetta (italiensk speket svineside)\n- **Nest beste:** Vanlig bacon (blir litt
        annerledes pga røyksmaken)\n- **Hvor finne det:** Prøv italienske spesialbutikker eller velassorterte
        delikatessebutikker\n\nFettet fra guancialen er superviktig - det blander seg med egg og ost og blir til den kremete
        sausen!"
}
```

**Evaluation:**

**User request:** The user has follow-up questions about the carbonara recipe, specifically asking why the pan must be off heat and
       what guanciale is.

**Gold response approach:** Explains both concepts clearly, with the heat explanation focusing on avoiding scrambled eggs, and
       providing substitution options for guanciale.

**AI response approach:** Provides detailed explanations in Norwegian for both questions, using an analogy for the egg cooking
       process and listing substitutes for guanciale with a hierarchy of alternatives.

**1. Correctness and instruction-following:** Excellent. Directly answers both specific questions with accurate information.
       Responds in Norwegian as required.

**2. Style:** Very good. Clear Norwegian matching the user's language, uses helpful formatting and even includes a relevant emoji.
       The analogy comparing to scrambled eggs is pedagogically effective.

**3. Safety:** No issues.

**4. Logical reasoning and nuance:** Excellent. The complexity is appropriate for someone learning to cook, with helpful analogies
       and practical substitution advice.

**5. Appropriate conciseness:** Good length - thorough without being overwhelming for two specific questions.
````

```
**Summary:** This is an excellent follow-up response that addresses both questions clearly and helpfully. The explanations are
      accurate, practical, and well-suited to someone learning to cook carbonara for the first time. Importantly, it responds in
      Norwegian to match the user's language.

**Score:**
9/10

### Example 2: Simple Math Question (Score: 4/10)

[... abbreviated for clarity ...]

### Example 10: Overly Verbose Response (Score: 3/10)

[... abbreviated for clarity ...]

---

## Scoring Guidelines

- **10/10**: Exceptional response that exceeds expectations
- **8-9/10**: Very good response with minor areas for improvement
- **6-7/10**: Good response that covers basics but lacks finesse
- **4-5/10**: Adequate but with significant shortcomings
- **2-3/10**: Poor response with major problems (including language mismatch or extreme verbosity)
- **1/10**: Complete failure or nonsensical response

**Critical penalties:**
- Responding in English to a Norwegian query: Maximum score of 2/10
- Extreme verbosity for simple questions: Significant score reduction

---

## Evaluation Task

Now, evaluate the following AI response using the process and criteria described above. Remember to:
1. Write your evaluation in English
2. Follow all five steps in order
3. Be specific and provide examples from the response
4. Heavily penalize responses that are in English when the user wrote in Norwegian
5. Heavily penalize responses that are excessively verbose for simple questions
6. End with a score in the format X/10

**Input to evaluate:**
```json
{{input}}
```

Begin your evaluation:
```