Progressive Perturbation with KTO for Enhanced Machine Translation of Indian Languages

Yash Bhaskar[1], Ketaki Shetye[1], Vandan Mujadia[1], Dipti Misra Sharma[1], Parameswari Krishnamurthy[1]

[1]International Institute of Information Technology – Hyderabad
Correspondence: yash.bhaskar@research.iiit.ac.in, ketaki.shetye@research.iiit.ac.in, vandan.mu@research.iiit.ac.in, dipti@iiit.ac.in, param.krishna@iiit.ac.in

Abstract

This study addresses the critical challenge of data scarcity in machine translation for Indian languages, particularly given their morphological complexity and limited parallel data. We investigate an effective strategy to maximize the utility of existing data by generating negative samples from positive training instances using a progressive perturbation approach. This is used to align the model with preferential data using Kahneman-Tversky Optimization (KTO). Comparing it against traditional Supervised Fine-Tuning (SFT), we demonstrate how generating negative samples and leveraging KTO enhances data efficiency. By creating rejected samples through progressively perturbed translations from the available dataset, we fine-tune the Llama 3.1 Instruct 8B model using QLoRA across 16 language directions, including English, Hindi, Bangla, Tamil, Telugu, and Santali. Our results show that KTO-based preference alignment with progressive perturbation consistently outperforms SFT, achieving significant gains in translation quality with an average BLEU increase of 1.84 to 2.47 and CHRF increase of 2.85 to 4.01 compared to SFT for selected languages, while using the same positive training samples and under similar computational constraints. This highlights the potential of our negative sample generation strategy within KTO, especially in low-resource scenarios.

1 Introduction

Machine Translation (MT) has made remarkable progress in recent years, yet significant challenges persist, particularly for low-resource languages. This is evident in the diverse family of Indian languages, such as Tamil, with its agglutinative morphology (Sarveswaran et al., 2021) and complex suffixation, and Santali, which employs an Austroasiatic script (Choksi, 2018) and follows an SOV word order. These languages feature rich morphological systems that complicate tokenization and alignment in MT (Kumar et al., 2009) while also suffering from a scarcity of parallel corpora essential for training robust translation models.

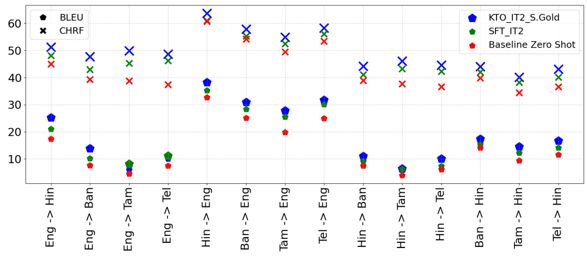

Figure 1: Performance Comparison of KTO (with Progressive Perturbation, using IndicTrans2 (IT2) output as positive samples and Perturbed Gold translations from BPCC Dataset (S.Gold) as negative samples) vs. SFT (using IT2 output as positive samples) and Zero-Shot on Llama 3.1 Instruct 8B.

The imbalance in training data between high-resource and low-resource languages has motivated the search for data-efficient techniques that maximize the utility of scarce resources. In this study, we tackle this challenge for Indian language machine translation by leveraging an approach based on preference alignment (Gisserot-Boukhlef et al., 2024). Rather than requiring extra positive training data, our method utilizes negative samples derived from existing high-quality translations. This enables the model to learn more effectively by distinguishing between subtle errors and accurate translations, thereby enhancing overall performance even in resource-scarce settings.

KTO (Ethayarajh et al., 2024) distinguishes itself from other preference-based methods by its flexibility in handling negative samples. Unlike Direct Preference Optimization (DPO) (Mecklenburg et al., 2024), which ideally requires rejected

Proceedings of Machine Translation Summit XX Volume 1, pages 344–352, June 23-27, 2025

completions for each positive example, and Proximal Policy Optimization (PPO) (Schulman et al., 2017), which necessitates the training of a separate and computationally intensive reward model, KTO allows for the utilization of negative samples without demanding a one-to-one pairing with every positive instance. This flexibility is particularly advantageous in low-resource settings, where generating a large number of diverse negative samples is challenging, and fine-tuning them increases computational cost.

In this work, we propose a progressive perturbation strategy to generate negative samples by systematically adding controlled noise to positive translations. These rejected samples, along with the original positives, are then used with the KTO algorithm for preference alignment. This approach enhances translation quality without requiring additional parallel data, making it particularly effective in low-resource scenarios.

We validate our approach on the Llama 3.1 Instruct 8B model across 16 language directions involving English, Hindi, Bangla, Tamil, Telugu, and Santali. Experimental results demonstrate that our KTO-based preference alignment with progressive perturbation consistently outperforms traditional SFT (Ouyang et al., 2022), yielding significant improvements in both BLEU and CHRF scores.

2 Related Work

Low resource machine translation (MT) remains a persistent challenge, motivating a variety of strategies to maximize data efficiency. Early work demonstrated that careful structuring of training data can significantly impact convergence and overall translation quality. For example, (Platanios et al., 2019) introduced a competence-based curriculum that adapts the complexity of training examples to the model's evolving capabilities. In a similar vein, (Zhang et al., 2018) and (Liu et al., 2020) showed that progressively increasing data complexity by ordering training examples from simple to complex can lead to faster convergence and improved performance in MT.

In addition to curriculum learning, data augmentation techniques have been widely explored to overcome the scarcity of parallel corpora in low-resource settings. (Xia et al., 2019) augmented training data using monolingual corpora from related high-resource languages, thereby enriching the available signal without the need for additional bilingual data. Similarly, (Ramesh et al., 2021) proposed a method that leverages bilingual word embeddings and transformer-based representations (e.g., BERT (Devlin et al., 2019)) to introduce new words and increase the presence of rare vocabulary items in the training corpus. While effective, these approaches typically require access to supplementary resources or complex augmentation pipelines.

Data quality also plays a critical role in MT, particularly when dealing with automatically generated or noisy datasets. To address this, (Kowtal et al., 2024) developed a data selection method that uses cross-lingual sentence representations derived from a multilingual SBERT model (Reimers, 2019) to filter out semantically mismatched sentence pairs. This filtering enhances the reliability of the training data but does not directly tackle the challenge of making optimal use of the available examples.

Multilingual transfer learning offers another avenue for improving low-resource MT by exploiting the inherent relatedness between languages. (Goyal et al., 2020) combined techniques such as unified transliteration and shared subword segmentation with pre-training across multiple languages to enhance transfer learning capabilities. Although effective, such approaches generally require a joint training framework that spans multiple language pairs.

In contrast to these paradigms, our work adopts a preference-based optimization strategy that directly maximizes the utility of existing data. Instead of relying solely on positive examples or external augmentation, we generate informative negative samples through a progressive perturbation strategy. By systematically degrading high-quality translations, our approach creates rejected samples that force the model to learn fine-grained distinctions between accurate and flawed outputs.

3 Methodology

We opted to carry out our experiments across six distinct languages divided into three categories as listed below, originating from three to four different language families and varying in resource availability.

1. **English to Indian Languages**: Translations from English to Bangla, Hindi, Santali, Tamil, and Telugu.

2. **Indian Languages to English**: Translations

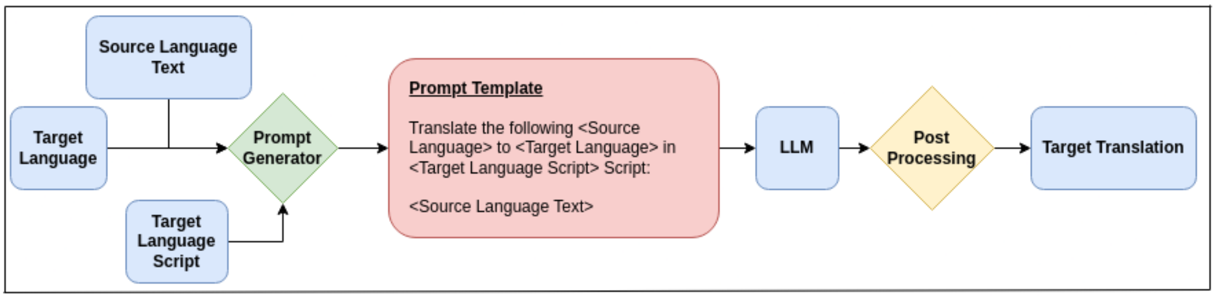

Figure 2: Prompting Mechanism for Translation

from Bangla, Hindi, Santali, Tamil, and Telugu to English.

3. **Indian to Indian Languages**: Translations between Hindi and Bangla, Tamil, and Telugu (excluding Santali due to limited parallel data).

To address data scarcity in Indian language translation, we compare zero-shot inference (baseline), SFT, and KTO. All experiments use the llamaFactory toolkit [1] (Zheng et al., 2024).

3.1 Model Selection

For this study, we selected the Llama 3.1 Instruct 8B model [2] (Dubey et al., 2024) as the foundation for fine-tuning. This choice was made after conducting initial zero-shot experiments to assess the baseline translation performance of several models relevant to our tasks. Specifically, we evaluated the Llama 3.1 Instruct 8B, Llama 3.2 Instruct 3B, and Llama 3.2 Instruct 11B models in a zero-shot setting across the language directions.

Table 1 summarizes the average BLEU and CHRF scores for each model across the language directions, evaluated on the Flores-200 devtest set.

Table 1: Average Zero-Shot BLEU and Chrf Scores for Llama Models

Model	BLEU	Chrf
Llama3.1-8B	12.06	39.00
Llama3.2-3B	5.58	33.22
Llama3.2-11B	11.94	39.47

As evident from Table 1, the Llama 3.2 Instruct 3B model demonstrated significantly lower translation quality compared to both the 8B and 11B parameter versions. Notably, the zero-shot translation performance of the Llama 3.1 Instruct 8B and Llama 3.2 Instruct 11B models was remarkably similar. Given this performance parity, and considering computational resource constraints for extensive fine-tuning experiments, we opted to proceed with the Llama 3.1 Instruct 8B model.

3.2 Data and Preprocessing

To ensure diverse and representative training data, we utilized the Wiki and Massive datasets from the Bharat Parallel Corpus Collection (BPCC) (Gala et al., 2023), sampling data as detailed in Table 2. The languages involved are Bangla (Bengali script), Hindi (Devanagari script), Santali (Ol Chiki script), Telugu (Telugu script), English (Latin script), and Tamil (Tamil script).

Language Pairs	Sample Size
Eng ↔Hin	25,000
Eng ↔Ban	25,000
Eng ↔Tam	25,000
Eng ↔Tel	25,000
Eng ↔San	25,000
Hin ↔Ban	10,000
Hin ↔Tam	10,000
Hin ↔Tel	10,000

Table 2: Language Pairs and Sample Sizes. In this, **Eng** refers to **English**, **Hin** refers to **Hindi**, **Ban** refers to **Bangla**, **Tam** refers to **Tamil**, **Tel** refers to **Telugu**, **Sat** refers to **Santali**

Throughout our experiments, a consistent prompt format was maintained for all techniques to ensure comparability. This prompt structure, visualized in Figure 2, includes specifications for the source and target languages, the target script, and the source sentence for translation.

[1] https://github.com/hiyouga/LLaMA-Factory
[2] https://huggingface.co/meta-llama/Meta-Llama-3.1-8B-Instruct

Perturbation Level	Example Sentence
10%	**Original:** A person with proliferative retinopathy will always be at risk for complications from new bleeding as well as glaucoma, new blood vessels. **Perturbed:** A bleeding with proliferative retinopathy will always be at risk for eagle from new person as well as glaucoma, new blood vessels.
30%	**Original:** The confluence of the Mudirapuzha, Nallathani, and Kundala rivers takes place in the heart of the city. **Perturbed:** The confluence of nervosa Mudirapuzha, kiskindha Nallathani, and excreted Kundala takes place in the heart of the city.
50%	**Original:** Most of the street children in Bangalore have come in search of business and new beginnings. **Perturbed:** Most of the children Bangalore in street have come in search of business photos car new.

Table 3: Examples of english sentence perturbations at 10%, 30%, and 50% intensity levels.

3.3 Perturbation Strategy

To introduce controlled errors, we apply a set of text modification operations that simulate common translation errors:

- **Word Addition**: Randomly inserts a word from a predefined vocabulary, disrupting fluency and potential meaning.
- **Word Deletion**: Removes a random word, leading to grammatical errors and incomplete sentences.
- **Word Shuffling**: Swaps the position of two random words, disrupting word order and comprehensibility.
- **Word Replacement**: Replaces a random word with another vocabulary word, introducing semantic errors.

The number of modifications depends on the perturbation intensity level. For instance, at 30% perturbation, a 20-word sentence undergoes approximately six modifications. This progressive perturbation (50% $\rightarrow$ 30% $\rightarrow$ 10%) exposes the model to coarse-to-fine errors, aligning with its improving discrimination capability during training. Some of the examples depicting the different levels of intensity-perturbation can be seen in Table 3

We integrate a *progressive perturbation strategy* with KTO to enhance model training. This method systematically introduces controlled noise into gold-standard human translations and IndicTrans2 (IT2) outputs, generating rejected completions for preference alignment. Perturbations are applied at varying intensities (10%, 30%, 50%), beginning with highly degraded (50%) translations to establish clear negative examples, then progressively reducing perturbation levels (30%, 10%) to introduce more nuanced errors. This staged approach refines the model's ability to distinguish subtle translation flaws, improving overall translation quality.

4 Fine-tuning and Optimization

We compare SFT with KTO, both applied to the Llama 3.1 Instruct 8B model.

4.1 Supervised Fine-Tuning

SFT serves as our baseline, evaluating standard supervised learning with limited parallel data. We explore two variations:

- **SFT on Gold-Standard Translations**: Fine-tuning on a subset of the Massive and Wiki datasets from BPCC using human translations as ground truth, setting a benchmark for high-quality supervision.
- **SFT on IT2-Generated Translations**: Fine-tuning with IT2-generated translations (Gala et al., 2023) as targets, assessing whether synthetic data can supplement or replace human translations in low resource settings.

These variations help assess the impact of different supervision sources on translation performance.

4.2 Kahneman-Tversky Optimization

We evaluate KTO using four configurations to analyze how different data sources influence preference alignment:

- **KTO-Gold-S.IT2**: Gold-standard translations as preferred examples, with rejected samples from perturbed IT2 outputs.
- **KTO-Gold-S.Gold**: Both preferred and rejected examples from gold-standard translations, with perturbation applied for rejection samples.

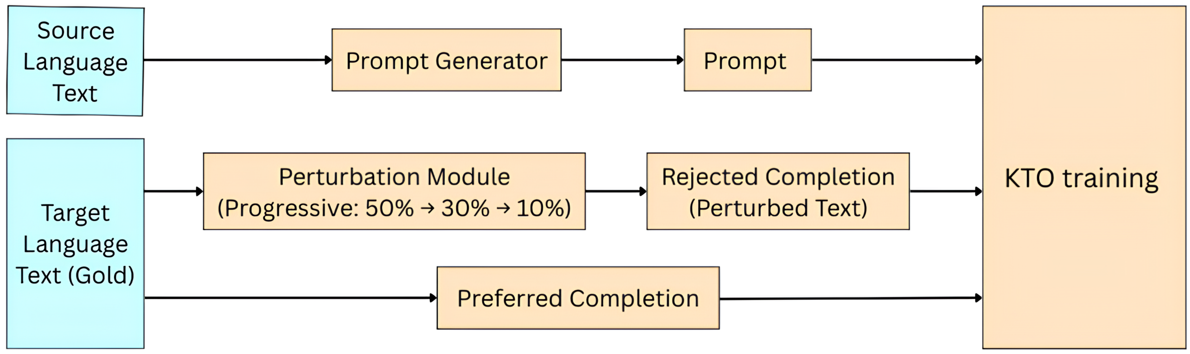

Figure 3: KTO training data workflow using gold text and progressive perturbation.

- **KTO-IT2-S.IT2**: IT2 generated translations as preferred examples, with perturbed versions as rejections.
- **KTO-IT2-S.Gold**: IT2 generated translations as preferred examples, with perturbed gold-standard translations as rejections.

These configurations systematically evaluate the effectiveness of KTO in low resource translation, demonstrating its potential to outperform SFT under identical data constraints.

4.3 Training Configuration

We fine-tune the Llama 3.1 Instruct 8B model for 1 epoch for both the SFT and KTO experiments. Due to computational constraints, further experimentation with additional epochs or technique combinations was not feasible.

To enhance computational efficiency, we employ 4-bit quantization (Kim et al., 2024) using QLoRA (Dettmers et al., 2023) for parameter-efficient fine-tuning. The specific hyperparameter configurations for LoRA are outlined in Table 5.

5 Evaluation

We evaluated the translation quality of the fine-tuned models using BLEU[3] (Post, 2018) and CHRF[4] (Popović, 2015) on the `dev` and `devtest` splits of the Flores-200 Benchmark Dataset[5] (Costa-jussà et al., 2022). We used the sacreBLEU library for BLEU and chrF calculation. The Flores-200 dataset provided a comprehensive benchmark for evaluating machine translation across various language pairs.

[3] https://github.com/mjpost/BLEU
[4] https://github.com/marian-nmt/chrf
[5] https://github.com/facebookresearch/flores

6 Results and Discussion

6.1 Overall Performance Comparison (SFT vs. KTO)

Our experiments demonstrate the effectiveness of KTO-based preference alignment with progressive perturbation for low-resource Indian language translation. As shown in Figure 1, KTO consistently outperforms SFT in selected languages. In the Flores-200 devtest set, we observed an average BLEU improvement from 1.84 to 2.47 and CHRF from 2.85 to 4.01 compared to SFT. These gains, achieved with the same positive training data and computational constraints, highlight the data efficiency of our approach.

6.2 KTO Configuration Analysis

Among KTO variants, KTO_IT2_S.Gold achieved the highest scores, while KTO_Gold_S.IT2 performed the lowest.

Using IT2-generated translations as the preferred completion consistently outperformed gold-standard human translations, aligning with trends observed in SFT. This suggests that IT2 translations may provide a more effective learning signal than gold translations in our setup. Additionally, using perturbed gold translations (S.Gold) as rejected examples generally resulted in better model alignment than perturbed IT2 translations (S.IT2), likely due to the higher intrinsic quality of gold translations.

6.3 Language-Specific Observations

A notable exception was Santali, where SFT outperformed all KTO variants. This outcome is likely due to the model's limited initial proficiency in Santali. Since KTO relies on negative examples, it may amplify noise when the baseline quality is extremely low. In such cases, the model might learn

Model	Metric	English→XX					XX→English					Hin→XX			XX→Hin		
		Hin	Ban	Tam	Tel	Sat	Hin	Ban	Tam	Tel	Sat	Ban	Tam	Tel	Ban	Tam	Tel
Llama3.1-Instruct-8B	BLEU	17.43	7.64	4.54	7.43	0.03	32.71	25.14	19.79	24.92	0.63	7.51	4.03	6.12	14.16	9.36	11.52
	CHRF	45.04	39.33	38.74	37.44	2.40	60.74	54.30	49.44	53.41	18.88	38.88	37.68	36.66	39.88	34.50	36.64
Llama3.2-Instruct-3B	BLEU	7.97	4.04	4.10	6.01	0.01	10.14	9.07	8.38	10.56	0.03	6.11	4.14	5.33	4.61	3.45	5.30
	CHRF	36.75	35.35	39.88	38.66	2.06	47.49	44.56	42.27	46.16	3.47	37.25	37.90	35.99	27.39	25.75	30.66
Llama3.2-Instruct-11B	BLEU	17.06	7.08	4.46	6.60	0.01	30.94	25.29	19.95	26.24	1.63	7.31	3.95	5.71	13.87	8.69	12.22
	CHRF	44.68	38.23	40.22	37.76	2.65	59.85	54.62	50.19	54.59	23.41	38.70	38.01	36.19	40.51	33.42	38.41
SFT_Gold	BLEU	20.74	8.81	6.75	10.38	2.24	34.73	27.37	24.64	28.95	7.54	8.36	5.68	7.70	14.73	12.33	13.95
	CHRF	46.98	40.23	44.30	44.09	27.74	61.17	54.98	51.93	56.13	30.21	39.48	41.34	40.39	40.59	37.00	40.04
SFT_IT2	BLEU	21.05	10.15	8.20	11.10	**2.51**	35.25	28.30	25.45	30.15	**7.8**	9.10	6.25	7.35	15.30	12.20	14.15
	CHRF	48.20	43.05	45.35	46.25	**31.61**	61.05	55.30	52.45	56.15	**30.56**	41.15	43.20	42.35	42.10	38.25	40.10
KTO_Gold_S.IT2	BLEU	20.88	9.64	7.25	9.57	0.78	34.68	27.27	23.95	28.17	4.63	8.84	5.21	7.35	15.41	11.94	14.21
	CHRF	47.48	41.96	45.10	43.81	19.34	61.26	54.89	51.63	55.15	25.14	40.65	41.91	40.47	41.40	37.00	40.01
KTO_Gold_S.Gold	BLEU	21.15	9.44	7.04	10.10	0.70	34.31	27.57	24.28	28.19	4.50	8.50	5.67	7.18	14.74	12.31	13.89
	CHRF	48.04	42.05	44.82	44.08	19.22	61.13	55.40	51.76	55.49	25.18	40.69	41.50	40.30	40.94	37.33	39.96
KTO_IT2_S.IT2	BLEU	21.99	11.15	**8.71**	**11.49**	0.15	36.68	30.09	27.11	31.17	5.66	10.09	**7.32**	9.13	16.99	13.68	15.30
	CHRF	50.02	45.01	48.40	47.30	13.42	62.64	57.20	53.95	57.52	26.17	43.00	45.98	43.71	43.60	39.40	41.69
KTO_IT2_S.Gold	BLEU	**25.26**	**13.81**	6.00	10.09	0.43	**38.26**	**30.96**	**27.78**	**31.68**	7.46	**10.96**	6.35	**10.08**	**17.33**	**14.55**	**16.64**
	CHRF	**51.21**	**47.74**	**49.93**	**48.63**	14.56	**63.72**	**57.94**	**54.93**	**58.27**	28.18	**44.27**	**46.17**	**44.67**	**44.10**	**40.17**	**43.07**

Table 4: Performance comparison of Zero-Shot Llama models vs. SFT & KTO fine-tuned Llama 3.1 Instruct-8B on Flores DevTest. SFT models use supervised fine-tuning with either gold-standard human translations (SFT_Gold) or IndicTrans2-generated translations (SFT_IT2). KTO models apply Kahneman-Tversky Optimization with different preference and rejection criteria: gold-standard translations with perturbed IT2 (KTO_Gold_S.IT2), gold-standard translations with perturbed gold-standard translations (KTO_Gold_S.Gold), IT2 translations with perturbed IT2 (KTO_IT2_S.IT2), and IT2 translations with perturbed gold-standard translations (KTO_IT2_S.Gold). All SFT and KTO models are fine-tuned versions of Llama 3.1 Instruct-8B.

Method	**Value**
LoRA modules	PEFT
Rank	8
Alpha	8
Dropout	0
Learning rate	5e-5
Effective batch size	64
Epochs	1

Table 5: Hyper-parameter configurations for LoRA

to avoid all translation choices from the Santali data, including those that are correct.

7 Conclusion

This study explores KTO with progressive perturbation for Indian language translation, demonstrating its superiority over SFT in most cases and highlighting its potential to maximize the utility of existing data in resource-scarce scenarios. Our method systematically degrades high-quality translations through controlled perturbations, generating a spectrum of negative examples ranging from overtly erroneous to subtly flawed outputs. These negative samples provide a rich training signal, helping the model distinguish between accurate and error-prone translations, thereby enabling efficient learning from limited data.

Notably, IT2-generated translations were more effective than gold-standard translations as preferred completions, raising questions about the reliability of the gold data in the BPCC Dataset. However, KTO was less effective in extremely low-resource cases like Santali, where SFT outperformed it, suggesting that KTO's effectiveness depends on the model's initial proficiency in a given language.

8 Limitations

In conducting our experiments, we relied on high-performance GPUs, specifically RTXA6000. However, we acknowledge that not everyone may have access to such powerful computing resources, which could present challenges in reproducing our experiments and achieving identical results. Despite these computing limitations, we were still able to carry out meaningful experiments, although we were unable to conduct more comprehensive analyses.

9 Future Work

Future work could explore several directions, including experimenting with different perturbation schedules for performance improvements. Additionally, addressing the challenges of applying KTO with progressive perturbation to low-resource languages like Santali is crucial, possibly by adapting the strategy or exploring alternative training objectives. Finally, applying this approach to other low-resource machine translation tasks across language families and domains could help assess its generalizability.

10 CO_2 Emission Related to Experiments

Experiments were conducted using a private infrastructure, which has a carbon efficiency of 0.813 $kgCO_2eq/kWh$. A cumulative of 648 hours of computation was performed on hardware of type RTX A6000 (TDP of 300W).

Total emissions are estimated to be 158.05 $kgCO_2eq$ of which 0 percent were directly offset.

Estimations were conducted using the MachineLearning Impact calculator presented in (Lacoste et al., 2019).

References

Nishaant Choksi. 2018. Script as constellation among munda speakers: the case of santali. *South Asian History and Culture*, 9(1):92–115.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Preprint*, arXiv:2305.14314.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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*.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*.

Jay Gala, Pranjal A Chitale, Raghavan AK, Varun Gumma, Sumanth Doddapaneni, Aswanth Kumar, Janki Nawale, Anupama Sujatha, Ratish Puduppully, Vivek Raghavan, et al. 2023. Indictrans2: Towards high-quality and accessible machine translation models for all 22 scheduled indian languages. *arXiv preprint arXiv:2305.16307*.

Hippolyte Gisserot-Boukhlef, Ricardo Rei, Emmanuel Malherbe, Céline Hudelot, Pierre Colombo, and Nuno M Guerreiro. 2024. Is preference alignment always the best option to enhance llm-based translation? an empirical analysis. *arXiv preprint arXiv:2409.20059*.

Vikrant Goyal, Sourav Kumar, and Dipti Misra Sharma. 2020. Efficient neural machine translation for low-resource languages via exploiting related languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 162–168, Online. Association for Computational Linguistics.

Jeonghoon Kim, Jung Hyun Lee, Sungdong Kim, Joonsuk Park, Kang Min Yoo, Se Jung Kwon, and Dongsoo Lee. 2024. Memory-efficient fine-tuning of compressed large language models via sub-4-bit integer quantization. *Advances in Neural Information Processing Systems*, 36.

Nidhi Kowtal, Tejas Deshpande, and Raviraj Joshi. 2024. A data selection approach for enhancing low resource machine translation using cross lingual sentence representations. In *2024 IEEE 9th International Conference for Convergence in Technology (I2CT)*, pages 1–7. IEEE.

Arun Kumar, V Dhanalakshmi, RU Rekha, KP Soman, S Rajendran, et al. 2009. Morphological analyzer for agglutinative languages using machine learning approaches. In *2009 International Conference on Advances in Recent Technologies in Communication and Computing*, pages 433–435. IEEE.

Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. 2019. Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*.

Xuebo Liu, Houtim Lai, Derek F. Wong, and Lidia S. Chao. 2020. Norm-based curriculum learning for neural machine translation. *Preprint*, arXiv:2006.02014.

Nick Mecklenburg, Yiyou Lin, Xiaoxiao Li, Daniel Holstein, Leonardo Nunes, Sara Malvar, Bruno Silva, Ranveer Chandra, Vijay Aski, Pavan Kumar Reddy Yannam, et al. 2024. Injecting new knowledge into large language models via supervised fine-tuning. *arXiv preprint arXiv:2404.00213*.

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.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom M. Mitchell. 2019. Competence-based curriculum learning for neural machine translation. *Preprint*, arXiv:1903.09848.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Akshai Ramesh, Haque Usuf Uhana, Venkatesh Balavadhani Parthasarathy, Rejwanul Haque, and Andy Way. 2021. Augmenting training data for low-resource neural machine translation via bilingual word embeddings and bert language modelling. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

N Reimers. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Kengatharaiyer Sarveswaran, Gihan Dias, and Miriam Butt. 2021. Thamizhi morph: A morphological parser for the tamil language. *Machine Translation*, 35(1):37–70.

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

Mengzhou Xia, Xiang Kong, Antonios Anastasopoulos, and Graham Neubig. 2019. Generalized data augmentation for low-resource translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5786–5796, Florence, Italy. Association for Computational Linguistics.

Xuan Zhang, Gaurav Kumar, Huda Khayrallah, Kenton Murray, Jeremy Gwinnup, Marianna J Martindale, Paul McNamee, Kevin Duh, and Marine Carpuat. 2018. An empirical exploration of curriculum learning for neural machine translation. *Preprint*, arXiv:1811.00739.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. *arXiv preprint arXiv:2403.13372*.

A Appendix

Model	Metric	English→XX					XX→English					Hin→XX			XX→Hin		
		Hin	Ban	Tam	Tel	Sat	Hin	Ban	Tam	Tel	Sat	Ban	Tam	Tel	Ban	Tam	Tel
Llama3.1-Instruct-8B	BLEU	17.37	7.45	4.63	7.21	0.01	31.06	26.31	20.85	26.60	0.53	7.56	4.32	6.00	14.18	10.15	12.01
	CHRF	45.76	39.54	39.05	37.89	2.16	60.23	55.12	50.37	54.82	18.35	39.33	38.29	36.75	39.95	35.39	37.20
Llama3.2-Instruct-3B	BLEU	7.61	2.93	3.64	5.97	0.01	8.79	8.11	7.99	10.90	0.02	7.24	4.13	5.45	4.29	3.02	4.98
	CHRF	36.64	34.11	38.49	39.31	1.96	45.34	43.49	41.42	46.84	3.53	39.12	37.79	36.14	27.30	24.26	30.97
Llama3.2-Instruct-11B	BLEU	17.30	8.04	4.68	6.14	0.04	31.36	25.49	21.24	27.17	1.42	6.83	3.31	5.76	14.09	9.66	12.65
	CHRF	45.15	40.39	40.91	36.78	3.05	60.03	55.00	50.70	55.36	22.83	38.87	34.70	36.69	41.20	34.72	39.24
SFT_Gold	BLEU	21.79	8.50	7.10	10.23	2.31	35.26	28.69	25.79	30.90	**7.62**	7.61	6.27	7.80	15.56	12.65	15.06
	CHRF	48.30	40.98	44.58	44.17	27.78	61.63	55.87	52.63	57.62	**30.18**	39.80	41.96	40.92	41.84	37.60	41.17
SFT_IT2	BLEU	23.45	12.34	**10.05**	12.30	**3.10**	36.15	30.25	27.40	32.10	6.05	9.15	7.25	8.40	17.20	14.35	16.50
	CHRF	50.12	45.67	48.24	47.10	**29.62**	62.34	57.12	54.30	58.45	25.10	43.15	45.30	43.20	44.05	41.25	43.40
KTO_Gold_S.IT2	BLEU	19.52	7.67	5.43	7.73	0.57	33.98	28.55	24.17	29.53	5.13	6.68	4.37	6.76	13.34	11.02	12.37
	CHRF	46.55	40.12	41.72	40.44	19.00	61.02	55.68	51.20	56.52	25.12	38.61	39.04	38.66	39.69	35.65	38.40
KTO_Gold_S.Gold	BLEU	22.05	9.71	7.69	10.35	0.96	33.98	28.55	24.91	29.60	5.15	8.18	6.02	7.36	16.00	12.62	14.83
	CHRF	48.87	42.81	45.38	44.20	19.23	61.02	55.68	51.73	56.61	25.26	41.10	42.04	40.56	42.62	37.51	41.15
KTO_IT2_S.IT2	BLEU	23.90	11.79	10.04	**12.33**	0.15	36.67	30.57	27.54	32.31	6.13	9.70	**7.83**	8.82	17.12	14.66	16.53
	CHRF	51.16	45.98	49.32	47.13	12.87	62.59	57.86	54.56	58.53	26.21	43.49	46.16	43.60	44.24	40.56	42.85
KTO_IT2_S.Gold	BLEU	**26.37**	**14.01**	5.60	12.01	0.34	**38.78**	**32.14**	**28.56**	**33.90**	7.33	**10.93**	6.77	**10.30**	**18.24**	**15.60**	**16.98**
	CHRF	**52.08**	**48.55**	**49.44**	**49.15**	14.02	**63.84**	**59.07**	**55.26**	**60.00**	28.18	**44.69**	**46.57**	**45.12**	**45.07**	**40.96**	**43.53**

Table 6: Performance comparison of Zero-Shot Llama models vs. SFT & KTO fine-tuned Llama 3.1 Instruct-8B on Flores Dev. SFT models use supervised fine-tuning with either gold-standard human translations (SFT_Gold) or IndicTrans2-generated translations (SFT_IT2). KTO models apply Kahneman-Tversky Optimization with different preference and rejection criteria: gold-standard translations with perturbed IT2 (KTO_Gold_S.IT2), gold-standard translations with perturbed gold-standard translations (KTO_Gold_S.Gold), IT2 translations with perturbed IT2 (KTO_IT2_S.IT2), and IT2 translations with perturbed gold-standard translations (KTO_IT2_S.Gold). All SFT and KTO models are fine-tuned versions of Llama 3.1 Instruct-8B.