Alif: Advancing Urdu Large Language Models via Multilingual Synthetic Data Distillation

Anonymous EMNLP submission

Abstract

Developing a high-performing large language models (LLMs) for low-resource languages such as Urdu, present several challenges. These challenges include the scarcity of high-quality datasets, multilingual inconsistencies, and safety concerns. Existing multilingual LLMs often address these issues by translating large volumes of available data. However, such translations often lack quality and cultural nuance while also incurring significant costs for data curation and training. To address these issues, we propose Alif-1.0-8B-Instruct, a multilingual Urdu-English model, that tackles these challenges with a unique approach. Prioritizing **quality over quantity**, we train the model on a high-quality, multilingual synthetic dataset (Urdu-Instruct), developed using a modified self-instruct technique. By using unique prompts and seed values for each task along with a global task pool, this dataset incorporates Urdu-native chain-of-thought based reasoning, bilingual translation, cultural relevance, and ethical safety alignments. This technique significantly enhances the comprehension of Alif-1.0-8B-Instruct model for Urdu-specific tasks. As a result, Alif-1.0-8B-Instruct, built upon the pretrained Llama-3.1-8B, demonstrates superior performance compared to Llama-3.1-8B-Instruct for Urdu specific-tasks. It also outperformed leading multilingual LLMs, including Mistral-7B-Instruct-v0.3, Qwen-2.5-7B-Instruct, and Cohere-Aya-Expanse-8B, all within a training budget of under $100. Our results demonstrate that high-performance and low-resource language LLMs can be developed efficiently and culturally aligned using our modified self-instruct approach.

1 Introduction

The rapid advancement of LLMs (Zhao et al., 2024) has revolutionized natural language processing (NLP) across multiple languages and applications. However, a significant disparity persists between high-resource languages, such as English, and low-resource languages, such as Urdu. These disparities create technological barriers for billions of speakers of underrepresented languages, limiting their access to AI-driven tools and advancements. The inclusion of low-resource languages in LLM development is not merely a technical challenge but a crucial step toward fostering inclusive, globally accessible AI systems that cater to diverse linguistic communities.

Developing high-performing LLMs for low-resource languages presents several challenges, including the scarcity of high-quality datasets, multilingual inconsistencies, translation inaccuracies, reasoning limitations, and ethical concerns. A common approach to addressing these challenges relies on leveraging translated data from high-resource languages. However, translations often fail to capture regional knowledge and cultural nuances, leading to compromised language representation and ineffective communication in low-resource settings (Aharoni et al., 2019; Conneau et al., 2020).

In the case of Urdu LLMs, additional factors contribute to their underperformance. Urdu's linguistic complexity, including its unique alphabet, intricate grammar, syntax, and morphology, poses significant challenges in adapting NLP techniques developed for English. Furthermore, Urdu has borrowed extensively from regional languages such as Hindi, Punjabi, and Persian and is written in both the Perso-Arabic and Devanagari scripts, adding additional layers of complexity. While multilingual models exhibit some degree of understanding, their generation capabilities remain inadequate, particularly for languages with syntactic structures and writing systems distinct from English. Among these challenges, the lack of high-quality datasets stands out as a fundamental limitation. Current

Urdu datasets are sparse, manually labeled, and contain only a few thousand instances—insufficient for training robust LLMs. This scarcity results from multiple factors, including limited digitization of Urdu literature, funding and infrastructure constraints, and the complexities of annotating Urdu text, which require linguistic expertise and standardized guidelines. Furthermore, translated data often fails to retain cultural nuances (AlKhamissi et al., 2024; Ramaswamy et al., 2024), such as idiomatic expressions and contextual meanings, thereby reducing a model's ability to generate culturally relevant responses. Additionally, multilingual LLMs suffer from catastrophic forgetting, where training across multiple languages or modalities can degrade performance on certain language subsets unless carefully managed. The challenge of evaluation further complicates this issue (Yu et al., 2022), as creating frameworks that fairly and accurately assess performance across diverse languages and cultures demands significant expertise and resources. These issues are particularly pronounced for South Asian low-resource languages like Urdu, which, despite its online presence, lacks the research-driven resources necessary to develop competitive models (Tahir et al., 2025; Ahuja et al., 2024). The homogeneity of existing datasets and evaluation standards exacerbates the underrepresentation of diverse linguistic and cultural contexts in modern LLMs, highlighting the urgent need for targeted efforts to bridge these gaps and promote inclusivity in multilingual AI development.

To address all these challenges, Alif-1.0-8B-Instruct model offers a promising solution to the limitations of conventional multilingual training approaches. By leveraging a modified self-instruct technique, this model incorporates a carefully curated Urdu dataset, specifically designed to enhance Urdu generation quality, bilingual translation, culturally aware understanding, and Urdu-native chain-of-thought based reasoning capabilities. This unique multilingual synthetic data distillation approach not only improves the model's performance on Urdu and English tasks but also upholds ethical commitments to safety and cultural sensitivity (Mitchell et al., 2019). Prior research has demonstrated that tailored datasets significantly enhance the effectiveness of language models, enabling deeper linguistic and cultural understanding (Kulkarni et al., 2023). By using a carefully curated Urdu dataset, Alif-1.0-8B-Instruct addresses persistent challenges in multilingual language modeling within constrained computational budgets (Husan and Shakur, 2023).

Alif-1.0-8B-Instruct demonstrates a significant leap in Urdu-specific task comprehension, outperforming leading multilingual LLMs. Its training pipeline follows a structured process: continued pretraining to reinforce foundational understanding, fine-tuning on the synthetic Urdu-Instruct dataset to enhance comprehension, incorporation of translated Urdu data for broader knowledge, and replayed English data to mitigate catastrophic forgetting. As a result, Alif-1.0-8B-Instruct, built upon the pretrained Meta Llama-3.1-8B base, demonstrates superior performance compared to Llama-3.1-8B-Instruct. (Aaron et al., 2024) in Urdu-specific benchmarks while maintaining strong English fluency. It also outperforms prominent multilingual models such as Mistral-7B-Instruct-v0.3 (Jiang et al., 2023), Qwen-2.5-7B-Instruct (Yang et al., 2025), and Cohere-Aya-Expanse-8B (Dang et al., 2024), all within an optimized training budget of less than $100.

1.1 Contribution

Our work introduces several key contributions to the development and fine-tuning of large language models, particularly focusing on multilingual and Urdu-specific capabilities:

- Multilingual Urdu-English Model: We present Alif-1.0-8B-Instruct, a multilingual (Urdu-English) model that outperforms leading multilingual LLMs on Urdu-translated MGSM (Shi et al., 2022; Cobbe et al., 2021), and Alpaca Eval (Li et al., 2023; Dubois et al., 2025, 2024), Dolly General QA (Conover et al., 2023), benchmarks.

- Modified Self-Instruct Technique: We introduce an enhanced self-instruct approach using diverse prompts and a global task pool. Each task is guided by unique prompts and seed values to capture cultural diversity, output structure, and task-specific nuances. A centralized task pool with human feedback ensures uniqueness and prevents redundancy. This scalable method improves instruction quality and can be adapted to other low-resource languages for broader NLP development.

- High-quality Urdu-Instruct Dataset: We curated a high-quality multilingual synthetic

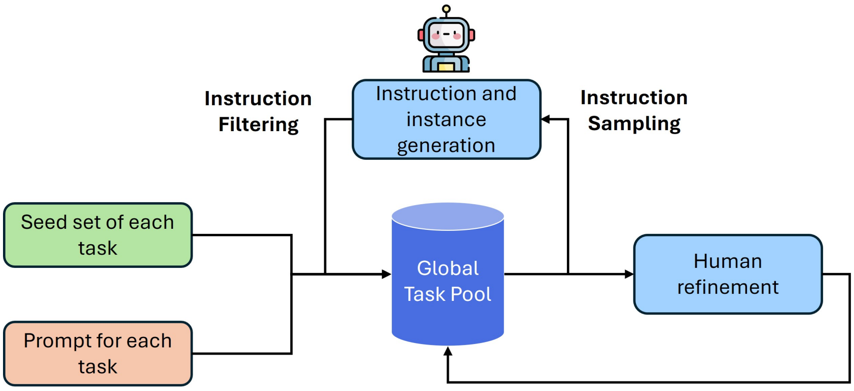


Figure 1: Flowchart of the Modified Self-Instruct technique for Urdu-Instruct dataset generation.

dataset of 51,686 examples using a modified self-instruct method. It enriches Urdu capabilities through native chain-of-thought reasoning, bilingual translation, and cultural nuance. This approach also enabled the creation of a new Urdu evaluation set with ∼150 examples per task.

- Evaluations on Urdu-Translated Benchmarks and New Evaluation Dataset: We evaluate Alif-1.0-8B-Instruct on multiple Urdu-translated benchmarks—including MGSM, AlpacaEval, and Dolly General QA — demonstrating its effectiveness over state-of-the-art models. Results on our new Urdu evaluation set further highlight its strength in domain-specific tasks.

The rest of the paper is organized as follows: Section 2 introduces the Urdu-Instruct dataset and our modified self-instruct method. Section 3 details the Alif-1.0-8B-Instruct model, its training setup, and optimization techniques. Section 4 presents evaluation results on Urdu and English tasks. Section 5 examines quantization impacts on performance and deployment. Section 6 concludes with key takeaways and future directions in Urdu NLP and multilingual LLMs, followed by a discussion of the model's limitations.

2 Urdu-Instruct Dataset

The Urdu-Instruct dataset, consisting of 51,686 examples generated using GPT-4o, api-version '2024-08-01-preview', (Achiam et al., 2024), is a crucial component in fine-tuning Alif-1.0-8B-Instruct. It contains instructions and responses for seven key Urdu tasks: Generation (5,907), Ethics (9,002), QA (8,177), Reasoning (9,590), Translation (10,001), Classification (4,662), and Sentiment Analysis (4,347). The dataset was created using a self-instruct (Wang et al., 2023) technique improved for cultural and linguistic nuance as shown in Figure 1[1] and explained below.

2.1 Modified self-instruct technique

1. Unique Prompt and Seed Values for each Task: To capture task-specific features, variations in output formats, and enhance cultural nuance, each task was assigned a distinct prompt and set of seed values. This ensured a richer and more diverse set of training examples, improving the model's adaptability to different contexts.

2. Global Task Pool: While individual tasks had unique prompts and seed values, all generated instructions were consolidated within a single global task pool. This approach prevented duplication and ensured the uniqueness of each task distribution across the dataset.

3. Instruction Sampling and Generation: Each prompt is augmented with random four human-annotated seed values and two machine-generated values to increase variability and ensure high-quality data. GPT-4o generates 20 instructions and corresponding outputs per batch.

4. Post-Processing and Filtering:

    - Instructions shorter than three words or longer than 150 words were removed.
    - Instances containing unsuitable keywords for language models were filtered out.

[1] Bot image: Flaticon.com

- Instructions starting with punctuation or containing characters other than Urdu and English, were rejected.
- Each newly generated instruction was compared with all previously generated instructions across all tasks in the global task pool using a ROUGE score threshold of 0.7. Any instruction exceeding this similarity threshold was rejected.

5. Human Refinement: The dataset was further cleaned by human annotators to refine Urdu grammar, ensure factual correctness, and eliminate any accidental inclusion of unethical content or non-Urdu/non-English characters. Additional details are provided in Appendix C.

2.2 Urdu-Instruct dataset features

This dataset covers a broad range of use cases, including text generation, ethical and safety considerations, factual question answering, logical reasoning, bilingual translation, classification, and sentiment analysis. Each task is designed to enhance the model's ability to understand and generate Urdu text effectively while maintaining high accuracy and cultural relevance.

- CoT-Based Urdu Reasoning: We use Urdu-native Chain-of-Thought prompts and structured reasoning tasks to enhance the model's logical abilities. This also improved performance in classification and sentiment analysis through better contextual understanding.

- Bilingual Translation: To reinforce the relationship between Urdu and English, we introduced bilingual translation tasks covering four distinct scenarios:

Instruction	Input	Output
Urdu	English	Urdu
Urdu	Urdu	English
English	Urdu	English
English	English	Urdu

Table 1: Instruction-Input-Output configurations.

- Ethics and Safety: We align ethical considerations with cultural and regional norms, enabling more context-aware and safer AI behavior.

- Generation and QA: Incorporating both open- and closed-ended QA tasks improves Alif's generation quality, coherence, and language understanding.

Using the same method, we created the Urdu Evaluation Set with ~150 instructions per category, offering a benchmark for evaluating multilingual models on Urdu tasks.

3 Multilingual Urdu-English Model: Alif-1.0-8B-Instruct

The development of Alif involves the integration of multiple datasets, each selected to serve a distinct role in the continued pre-training and fine-tuning process. This carefully structured approach is essential to enhancing the model's proficiency across a diverse range of tasks, ensuring robust linguistic capabilities.

3.1 Datasets used for continued pre-training

For the continued pre-training phase, we primarily utilize a dataset consisting of 200K Urdu Wikipedia articles[2]. This dataset is utilized to ensure diversity and coverage across multiple domains, aiming to provide a strong foundational understanding of language structures. By utilizing this dataset, we are able to maintain efficient training costs while ensuring the model achieved strong performance in text comprehension and generation tasks. We pretrain *unsloth/Meta-Llama-3.1-8B*[3] with the standard Causal Language Modeling (CLM) task. For an input tokens $\boldsymbol{x} = (x_0, x_1, x_2, \ldots)$, the model is trained to predict the next token as output x_i autoregressively. The goal of the pre-training is to minimize negative log-likelihood loss as shown in equation 1.

$$\mathcal{L}_{\text{CPT}}(\Theta) = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}_{\text{PT}}} \left[-\log p(\mathbf{x}; \Theta) \right] \qquad (1)$$

where Θ represents the model parameters, $\mathcal{D}_{\text{PT}}$ is the continued pre-training dataset, x_i is the next token to be predicted, $x_0, x_1, \ldots, x_{i-1}$ is the input context, and CPT stands for continued pre-training.

3.2 Datasets used for fine-tuning

Alif is trained on a diverse collection of instruction-following datasets, comprising a total of 105,339 examples. These datasets include Urdu-Instruct

[2] Dataset: wikimedia/wikipedia

[3] Model: Meta-Llama-3.1-8B

(51,686 examples), translated dataset[4] (28,910 examples), ULS_WSD (4,343 examples) (Saeed et al., 2019), English Alpaca (10,400 examples) (Taori et al., 2023), and OpenOrca (10,000 examples) (Lian et al., 2023; Mukherjee et al., 2023; Longpre et al., 2023; Touvron et al., 2023b,a).

The fine-tuning task is similar to the causal language modeling task: the model is prompted using the Stanford Alpaca template for fine-tuning and inference, and the input prompt looks like:

> *Below is an instruction that describes a task. Write a response that appropriately completes the request.*
>
> *### Instruction:*
> *{instruction}*
>
> *### Input (If available):*
> *{input}*
>
> *### Response: {output}*

The loss is only calculated on the *{output}* part of the prompt and can be expressed as:

$$\mathcal{L}_{\text{SFT}}(\Theta) = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\text{SFT}}}\left[-\log p(\mathbf{x}_i \mid \mathbf{x}; \Theta)\right] \quad (2)$$

Here, Θ represents the model parameters and $\mathcal{D}_{\text{SFT}}$ is the fine-tuning dataset, $\boldsymbol{x} = (x_0, x_1, \ldots)$.

The selection of these datasets is strategically designed to strengthen the model's instruction-following capabilities across multiple Urdu domains. Urdu-Instruct and translated datasets constitute the majority of the instruction-tuning data, while English Alpaca and OpenOrca are employed as replay datasets to mitigate catastrophic forgetting, preserving previously acquired knowledge throughout the fine-tuning process.

3.3 Experimental setup and training details

Low-Rank Adapters (LoRA) provide an efficient approach for continued pre-training and fine-tuning large language models, as introduced by (Hu et al., 2021). This technique is particularly advantageous due to its computational efficiency, enabling model training without extensive GPU resources. We have employed LoRA and Unsloth framework[5] to optimize training costs while accelerating the overall training process. For our experiments, we utilized the *unsloth/Meta-Llama-3.1-8B* as base model with LoRA applied to the following components:

- QKVO (Self-Attention Layers): Query, Key, Value, Output projections.
- MLP (Feedforward Layers): Gate, Up, Down projections.
- ET-LH (Embedding & Output Layers): Embedding tokens and Language Model Head.

By leveraging LoRA adapters, we have optimized the base model efficiently. The continued pre-training phase is conducted using Wikipedia articles, followed by fine-tuning. The training is performed using BF16 precision to ensure stability and efficiency. A cosine learning rate scheduler is employed, with an initial learning rate of 2×10^{-5} for continued pre-training and 5×10^{-5} for fine-tuning.

For training stage, we have utilized an Nvidia A100 GPU with 80GB of VRAM. The model is pre-trained for one epoch over 200K wikipedia dataset, requiring 23 hours on Runpod[6]. The fine-tuning phase, consisting of two epochs, have taken an additional 16 hours. We have accessed the A100 GPU via Runpod at a rate of $1.64 per hour with a total training duration of 39 hours. As a result, the overall training cost remained under $100 (as of February 12, 2025).

The detailed hyperparameters used for continued pre-training and fine-tuning are summarized in Table 5, with additional information provided in Appendix B.

4 Results on Instruction-Following Tasks

Evaluating large language models (LLMs) for low-resource languages like Urdu presents unique challenges due to the limited availability of high-quality benchmarks. Additionally, while instruction-tuned models such as Llama-3.1-8B-Instruct have demonstrated strong multilingual capabilities, their performance in Urdu NLP tasks remains underexplored. In this section, we benchmark Alif-1.0-8B-Instruct (Alif) against Llama-3.1-8B-Instruct (Llama) and other LLMs using the alpaca chat template across various benchmarks. These evaluations were conducted on Runpod, using an A40 GPU with 48GB VRAM.

4.1 Results on Urdu-translated benchmarks

To ensure a rigorous and fair evaluation, we employ GPT-4o (Achiam et al., 2024), a LLM-as-a-judge scoring mechanism. Each response is assigned a 10-point score. To enhance the reliability of automated scoring, we refine GPT-4o's evaluation with

[4] Dataset: ravithejads/alpaca_urdu_cleaned_output

[5] Website: unsloth.ai

[6] Website: runpod.io

human feedback. Our process involves continuous monitoring of GPT-4o's explanations across various evaluation tasks, enabling human feedback to identify inconsistencies and improve the evaluation prompt accordingly. This iterative refinement ensures greater accuracy and consistency in the evaluation of Urdu NLP models.

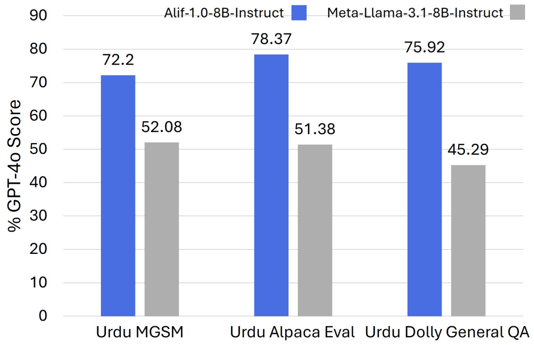


Figure 2: Comparison of Alif-1.0-8B-Instruct and Meta-Llama-3.1-8B-Instruct on Urdu-translated benchmarks.

Task	Llama-3.1-Inst.	Alif-1.0-Inst.
Generation	42.8	**90.2**
Ethics	27.3	**85.7**
QA	30.5	**73.8**
Reasoning	45.6	**83.5**
Translation	58.9	**89.3**
Classification	61.4	**93.9**
Sentiment	54.3	**94.3**
Weighted Avg.	45.7	**87.1**

Table 2: Experimental results on Urdu evaluation set.

We utilize a structured prompt template to evaluate and compare the outputs of two systems, where System 1 represents the reference (ground-truth) response and System 2 is the generated response being evaluated. The model's final score is computed as the percentage ratio of the System 2 score to the System 1 score, reflecting how closely the generated output aligns with the reference. The prompt template used for this evaluation is provided below.

You are an LLM Response Evaluator.

The following are two ChatGPT-like systems' outputs. Please evaluate both a ten-point scale (1–10), where 10 is the highest score, and provide a explanation for the scores. The evaluation criteria are:

- Relevance: Does the response directly and adequately address the user's prompt?
- Correctness: Is the information provided accurate and factually correct?
- Clarity: Is the response well-structured and free from unnecessary repetition or verbosity while maintaining completeness?
- Formatting Issues: Does the response have a consistent structure and free from unnecessary elements or incorrect language characters?

### Prompt: {prompt}

### System1: {system1_output}

### System2: {system2_output}

We evaluate the models on a range of Urdu-translated benchmarks, including MGSM (250 math reasoning questions), AlpacaEval (806 instruction-following prompts), and a randomly sampled subset of Dolly General QA (220 open-ended questions). Across these diverse tasks, Alif consistently outperforms the base LLaMA model, demonstrating its improved reasoning and instruction-following capabilities in Urdu, as illustrated in Figure 2. Our evaluation also demonstrates that Alif significantly outperforms Llama in Urdu-specific NLP tasks, particularly in text generation, ethics, QA, translation, reasoning, classification, and sentiment as shown in Table 2.

4.2 Results across different models

Table 3 presents a comparative evaluation of Alif-1.0-8B-Instruct against several leading instruction-tuned models on Urdu-translated benchmarks, including MGSM, Alpaca Eval, and Dolly General QA. The results indicate that Alif-1.0-8B-Instruct consistently outperforms all other models, achieving the highest scores across all three benchmarks. Specifically, it attains 72.2 on MGSM, 78.4 on Alpaca Eval, and 75.9 on Dolly General QA, leading to an overall average of 75.5. These results suggest that Alif-1.0-8B-Instruct is exceptionally well-suited for handling Urdu-based NLP tasks, demonstrating superior reasoning, comprehension, and instruction-following capabilities.

These results highlight the efficacy of Alif-1.0-8B-Instruct in tackling Urdu-translated benchmarks with a clear performance advantage over its counterparts.

4.3 Results on English benchmarks

To assess whether Alif-1.0-8B-Instruct experiences catastrophic forgetting after adapting to Urdu, we evaluate its performance against Llama-3.1-8B-Instruct on a series of English-language bench-

Models	MGSM	Alpaca Eval	Dolly General QA	Average
Falcon-7b-instruct	21.0	23.2	21.4	21.8
Phi-3-small-8k-instruct	43.1	38.7	35.6	39.1
Mistral-7B-Instruct-v0.3	43.6	43.6	38.7	41.9
Llama-3.1-8B-Instruct	52.1	51.4	45.3	49.6
Granite-3.2-8b-instruct	52.4	60.4	52.9	55.3
Gemma-7b-it	57.5	58.0	54.5	56.6
Qwen2.5-7B-Instruct	62.7	61.5	55.2	59.8
Ministral-8B-Instruct-2410	69.4	62.2	54.4	62.0
Aya-expanse-8b	65.2	72.3	69.4	68.9
Alif-1.0-8B-Instruct	**72.2**	**78.4**	**75.9**	**75.5**

Table 3: Comparison of Alif-1.0-8B-Instruct with other models on Urdu translated benchmarks.

marks using *lm-evaluation-harness* (Gao et al., 2024) as shown in Table 4. Since English data was incorporated during fine-tuning as a replay dataset, we anticipate that Alif-1.0-8B-Instruct should maintain competitive results on English tasks.

The evaluation results show that Alif-1.0-8B-Instruct retains strong general reasoning capabilities and even outperforms Llama-3.1-8B-Instruct in benchmarks such as *arc_challenge*, *arc_easy*, and *hellaswag*, indicating that common sense and logical reasoning abilities are preserved.

However, a slight decline is observed in knowledge-intensive tasks, particularly *mmlu* where Llama-3.1-8B-Instruct achieves better results. The significant drop occurs in STEM and humanities categories of *mmlu*, suggesting that while replay-based fine-tuning helps retain general capabilities, some domain-specific knowledge is affected.

Overall, these results indicate that using replay datasets during fine-tuning was effective in mitigating catastrophic forgetting, though some specialized knowledge areas experienced minor degradation.

5 Effect of Different Quantization Methods

The deployment of large language models (LLMs) on various hardware architectures has traditionally been constrained by high computational and memory demands. However, the development of open source frameworks, such as *llama.cpp* (Gerganov, 2024), has facilitated the quantization of LLMs, significantly reducing their resource requirements and maintaining comparable accuracy for some quantized formats. This advancement also enables efficient local development, minimizing reliance on cloud services and enhancing data privacy.

5.1 Impact of quantization on Alif-1.0-8B-Instruct model

This section explores the effects of different quantizations on Alif-1.0-8B-Instruct model using *llama.cpp*. We assess the model's perplexity (PPL) on English text corpora (wiki-test-raw) and a Urdu-translated version across various GGUF quantization formats.: `Q2_K, Q3_K_M, Q4_K_M, Q5_K_M, Q6_K, Q8_0`, and `F16` (Half-precision). The results are depicted in Figure 3.

Higher-bit quantization formats such as 6-bit and 8-bit maintain similar perplexity levels to FP16 while substantially reducing model size as shown in Figure 4. Conversely, lower-bit quantization (2-bit, 3-bit, and 4-bit) results in higher perplexity, highlighting a tradeoff between efficiency and accuracy. The Urdu text corpus consistently shows lower perplexity compared to the English corpus, indicating better adaptation or linguistic properties influencing the model's comprehension.

Among the quantization format results, `Q6_K` and `Q8_0` emerge as optimal choices for deployment on personal computers, offering a practical balance between model size and accuracy. Lower-bit quantization (`Q3_K_M`, `Q4_K_M`) remains a viable option for resource-limited scenarios but comes with trade-offs in model performance. In contrast, `Q2_K` does not appear to be a viable solution due to a substantial increase in perplexity.

6 Conclusion

Building a high-performing Urdu LLM presents distinct challenges, including data scarcity, translation quality issues, and reasoning complexity. Existing methods often depend on large-scale trans-

Tasks	Version	Filter	n-shot	Metric	Llama-3.1-Inst.	Alif-1.0-Inst.
arc_challenge	1	none	0	acc	0.5171	**0.5478**
		none	0	acc_norm	0.5512	**0.5623**
arc_easy	1	none	0	acc	0.8190	**0.8258**
		none	0	acc_norm	0.7950	**0.8194**
hellaswag	1	none	0	acc	0.5914	**0.6135**
		none	0	acc_norm	0.7922	**0.8022**
mmlu	2	none	0	acc	**0.6798**	0.6177
- humanities	2	none	0	acc	**0.6425**	0.5530
- other	2	none	0	acc	**0.7438**	0.7007
- social sciences	2	none	0	acc	**0.7702**	0.7260
- stem	2	none	0	acc	**0.5842**	0.5268

Table 4: Alif-1.0-8B-Instruct vs. Llama-3.1-8B-Instruct on English benchmarks.

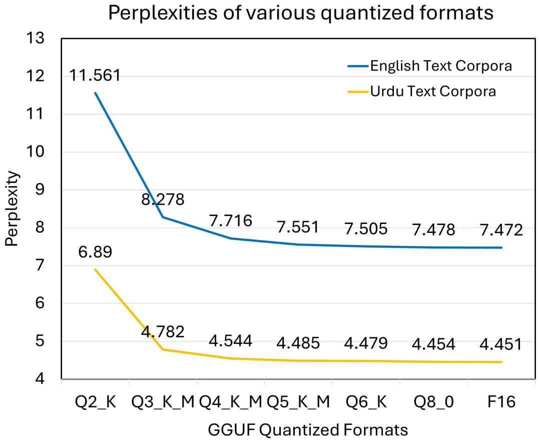


Figure 3: Perplexity comparison across GGUF quantization formats for Alif-1.0-8B-Instruct.

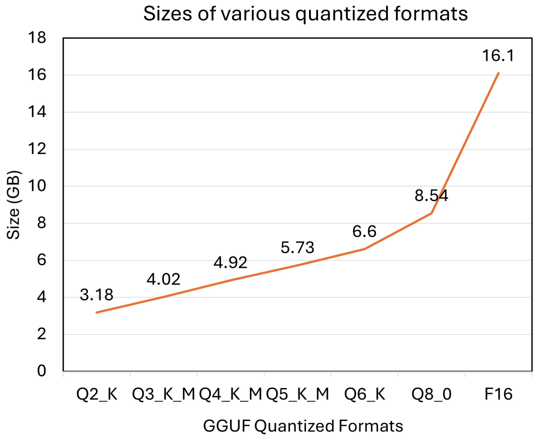


Figure 4: Memory footprint of different GGUF quantization formats for Alif-1.0-8B-Instruct.

lations, which degrade quality and raise data curation and training costs. We address this issue by continued pre-training and fine-tuning Alif-1.0-8B-Instruct on high-quality multilingual synthetic dataset — Urdu-Instruct, which captures cultural nuances, enables bilingual knowledge transfer, and enhances reasoning abilities. Our results demonstrate that efficient, high-performing Urdu LLMs are achievable. **Quality outweighs quantity**, an approach Alif exemplifies for low-resource languages.

We plan to expand high-quality datasets, enhance reasoning through model merging and reinforcement learning, and benchmark Alif against evolving standards. Alif is a key step toward culturally aligned, cost-effective Urdu NLP, with ongoing work driving inclusive AI forward.

Limitations

The Alif-1.0-8B-Instruct model, introduced in this paper, marks a significant step in Urdu NLP. However, in the spirit of rigorous research, it is imperative to discuss the inherent limitations that accompany this model.

- Urdu Task-Specific Knowledge: Despite high-quality pretraining and fine-tuning data—including the Urdu-Instruct dataset covering classification, reasoning, ethics, translation, and QA—some domain-specific and nuanced linguistic aspects remain underrepresented, limiting performance on culturally rich tasks.

- Harmful and Unpredictable Content: While designed to reject unethical prompts, the model may still produce harmful or misaligned outputs due to contextual limitations.

- Lack of Robustness: The model can behave inconsistently or illogically when faced with adversarial or rare inputs, highlighting the need for improved resilience.

Although some of these challenges can be mitigated in future iterations, we see this work as a crucial foundation that will drive further advancements in LLMs for Urdu and other low-resource languages.

License

The Alif-1.0-8B-Instruct model is a continued pretraining and fine-tuning derivative of the Llama-3.1-8B base model, which is released under the *Llama 3.1 Community License*.

The Urdu-Instruct dataset is released under the *Creative Commons Attribution–ShareAlike 4.0 International (CC BY-SA 4.0)* License,[7] which allows use, modification, and redistribution with attribution, provided derivative works are shared under the same license. All source code used for data generation and fine-tuning is released under the *MIT License*.

The datasets used for training the model are released under copyleft licenses, while the others are publicly available on Hugging Face without an explicitly specified license.

References

Aaron, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2024. Gpt-4 technical report.

R. Aharoni, M. Johnson, and O. Fırat. 2019. Massively multilingual neural machine translation. *Proceedings of the 2019 Conference of the North*.

Sanchit Ahuja, Divyanshu Aggarwal, Varun Gumma, Ishaan Watts, Ashutosh Sathe, Millicent Ochieng, Rishav Hada, Prachi Jain, Maxamed Axmed, Kalika Bali, and Sunayana Sitaram. 2024. Megaverse: Benchmarking large language models across languages, modalities, models and tasks.

Badr AlKhamissi, Muhammad ElNokrashy, Mai AlKhamissi, and Mona Diab. 2024. Investigating cultural alignment of large language models.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems.

A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, É. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

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

John Dang, Shivalika Singh, Daniel D'souza, Arash Ahmadian, Alejandro Salamanca, Madeline Smith, Aidan Peppin, Sungjin Hong, Manoj Govindassamy, Terrence Zhao, Sandra Kublik, Meor Amer, Viraat Aryabumi, Jon Ander Campos, Yi-Chern Tan, Tom Kocmi, Florian Strub, Nathan Grinsztajn, et al. 2024. Aya expanse: Combining research breakthroughs for a new multilingual frontier.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. 2025. Length-controlled alpacaeval: A simple way to debias automatic evaluators.

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2024. Alpacafarm: A simulation framework for methods that learn from human feedback.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, et al. 2024. A framework for few-shot language model evaluation.

Georgi Gerganov. 2024. llama.cpp. `https://github.com/ggerganov/llama.cpp`.

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

S. Husan and N. Shakur. 2023. Teachers' proficiency in english language assessment at an english-medium university: implications for elt training in pakistan. *Journal of Social Sciences Development*, 02:339–354.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

[7] `https://creativecommons.org/licenses/by-sa/4.0/`

M. Kulkarni, D. Preotiuc-Pietro, K. Radhakrishnan, G. I. Winata, S. Wu, L. Xie, and S. Yang. 2023. Towards a unified multi-domain multilingual named entity recognition model. *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*.

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

Wing Lian, Bleys Goodson, Eugene Pentland, Austin Cook, Chanvichet Vong, and "Teknium". 2023. Openorca: An open dataset of gpt augmented flan reasoning traces. `https://https://huggingface.co/datasets/Open-Orca/OpenOrca`.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The flan collection: Designing data and methods for effective instruction tuning.

M. Mitchell, S. Wu, A. Zaldivar, P. Barnes, L. Vasserman, B. Hutchinson, E. Spitzer, I. D. Raji, and T. Gebru. 2019. Model cards for model reporting. *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 220–229.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4.

Vikram V Ramaswamy, Sing Yu Lin, Dora Zhao, Aaron Adcock, Laurens van der Maaten, Deepti Ghadiyaram, and Olga Russakovsky. 2024. Geode: a geographically diverse evaluation dataset for object recognition. *Advances in Neural Information Processing Systems*, 36.

Ali Saeed, Rao Muhammad Adeel Nawab, Mark Stevenson, and Paul Rayson. 2019. A word sense disambiguation corpus for urdu. *Language Resources and Evaluation*, 53:397–418.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. 2022. Language models are multilingual chain-of-thought reasoners.

Munief Hassan Tahir, Sana Shams, Layba Fiaz, Farah Adeeba, and Sarmad Hussain. 2025. Benchmarking the performance of pre-trained llms across urdu nlp tasks. In *Proceedings of the First Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2025)*, pages 17–34.

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

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions.

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, , et al. 2025. Qwen2.5 technical report.

Xinyan Velocity Yu, Akari Asai, Trina Chatterjee, Junjie Hu, and Eunsol Choi. 2022. Beyond counting datasets: A survey of multilingual dataset construction and necessary resources.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2024. A survey of large language models.

A Potential Risks

While Alif-1.0-8B-Instruct marks significant progress in Urdu NLP, several potential risks accompany its use:

- Harmful and Biased Outputs: Despite safety training, the model may still produce harmful, racist, or discriminatory content, especially in response to ambiguous or adversarial prompts.
- Misuse in Unregulated Settings: The model could be used to generate propaganda, hate speech, or misinformation in settings where content moderation tools are limited or absent.
- Over-reliance Without Standard Benchmarks: The lack of strong Urdu evaluation datasets may lead users to place too much trust in the model, particularly in sensitive areas such as education, law, or public services.

B Experiments Setup

B.1 Training and evaluation environment

All pretraining and fine-tuning experiments for Alif-1.0-8B-Instruct were performed on an NVIDIA A100 GPU (80GB) using Runpod cloud infrastructure. The experiments were run within a Docker container configured with Python 3.10 and a 200GB persistent volume for model checkpoints, datasets, and logs. Model training leveraged the *Unsloth* framework, which enables efficient fine-tuning through low-rank adaptation (LoRA) and memory optimization techniques. Hyperparameter details of the experiment are given in Table 5

Configurations	Pre-training	Fine-tuning
Training Data	200K	105K
Epochs	1	2
Batch Size	64	64
Dropout	0.01	0
LR	2e-5	5e-5
LR_Type	Cosine	Cosine
Max Length	2048	2048
LoRA Rank	128	128
LoRA Alpha	32	32
LoRA Modules	QKVO, MLP, ET-LH	QKVO, MLP, ET-LH
Trainable Params(%)	14.72%	14.72%
Training Precision	BF16	BF16
Training Time	23 hours	16 hours

Table 5: Training Hyperparameters.

The environment was based on CUDA 12.2 and included the following key components:

- Model Training Frameworks:
  - `transformers==4.47.1`
  - `trl==0.13.0`
  - `peft==0.14.0`
  - `accelerate==1.2.1`
  - `unsloth @ 5dddf27`
- Core PyTorch and CUDA Stack:
  - `torch==2.5.1+cu121`
  - `torchvision==0.20.1+cu121`
  - `torchaudio==2.5.1+cu121`
  - `bitsandbytes==0.45.0`
  - `xformers==0.0.29.post1`
- Data Handling and Processing:
  - `datasets==3.2.0`
  - `pandas==2.2.2`
  - `tqdm==4.67.1`
  - `scikit-learn==1.6.0`
  - `libcudf-cu12`, `cupy-cuda12x`
- Experiment Tracking and Logging:
  - `wandb==0.19.1`

All models were trained using mixed-precision settings with gradient accumulation to enable scalable fine-tuning under limited GPU memory constraints. The evaluation was conducted in the same software environment as training, with the only difference being the GPU. Specifically, all evaluations were performed on an NVIDIA A40 GPU (48GB) using Runpod cloud infrastructure.

B.2 Modified Self-Instruct environment

The Urdu-specific instruction dataset used in this work was generated using a modified version of the Self-Instruct framework. This version was adapted to improve cultural relevance, apply toxicity filtering, and refine prompt structures for Urdu. The generation pipeline integrates language model prompting, semantic filtering, and instruction post-processing.

- Platform and Configuration:
  - Language Model: `gpt-4o` via `AzureOpenAI` API.
  - Python Version: 3.10
  - Concurrency: Multiprocessing with `Pool` (24 CPUs).
- Core Python Dependencies:
  - `openai` — GPT-4o API integration.
  - `rouge_score` — Semantic similarity filtering via ROUGE-L.
  - `numpy` — Batch operations and scoring computations.
  - `LughaatNLP` — Urdu-specific lemmatization and tokenization.
  - `tqdm`, `json`, `multiprocessing`, `re` — Preprocessing and utilities.

B.3 Urdu-Translation of Benchmarks

To translate benchmark datasets into Urdu, we used GPT-4o (API version: 2024-08-01-preview) with the following prompt to ensure high-quality, fluent, and culturally appropriate translations:

> *You are an expert in Urdu linguistics and translation. Translate the following sentence into Urdu with accurate grammar, natural fluency, and cultural appropriateness. The output should be only translation with no additional word.*
>
> *Sentence: {sentence}*

This prompt was used to translate all examples in the MGSM (250 math questions), AlpacaEval (806 instructions), and a 220-example subset of Dolly General QA into Urdu.

C Datasets Refinement

To refine the Urdu-Instruct dataset and Urdu-translated instruction data, we employed a structured human annotator selection process focused on linguistic quality and demographic diversity.

- Recruitment:
  - A public call for annotators was posted in October 2024, targeting native Urdu speakers with fluent typing skills and basic Excel knowledge.
  - The opportunity offered task-based compensation, with applications collected via a form by October 16, 2024.
- Shortlisting and Evaluation:
  - Candidates were asked to complete two tasks: (1) correcting an error-filled Urdu passage, and (2) verifying and correcting 50 Urdu-translated instructions in an Excel sheet using the guideline as shown in Figure 7.
  - 98 applicants attempted evaluation google form and 20 were shortlisted based on diverse demographics and task performance, and on-boarded to a dedicated Discord workspace. Among them, 14 were in the 18–24 age group, while 6 were between 25–34 years old and belong to various parts of Pakistan as shown in Figure 5.
- Final Selection:
  - These annotators, together with the author(s), contributed to refine translated and modified self-instruct datasets using the guidelines shown in Figure 6 and 7.
  - Annotators were compensated at a rate of 1000 Pakistani Rupees per hour, which is 4× of the minimum wage of Pakistan.

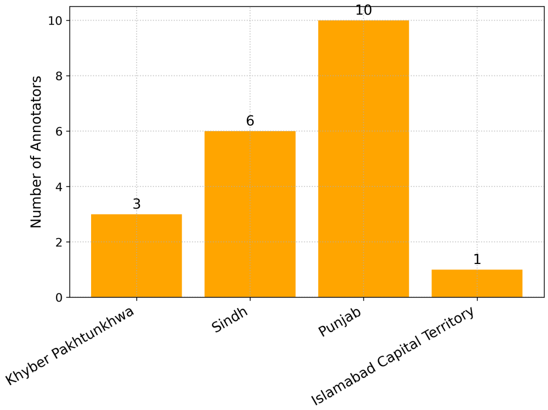


Figure 5: Annotator Demographics by Province in Pakistan.

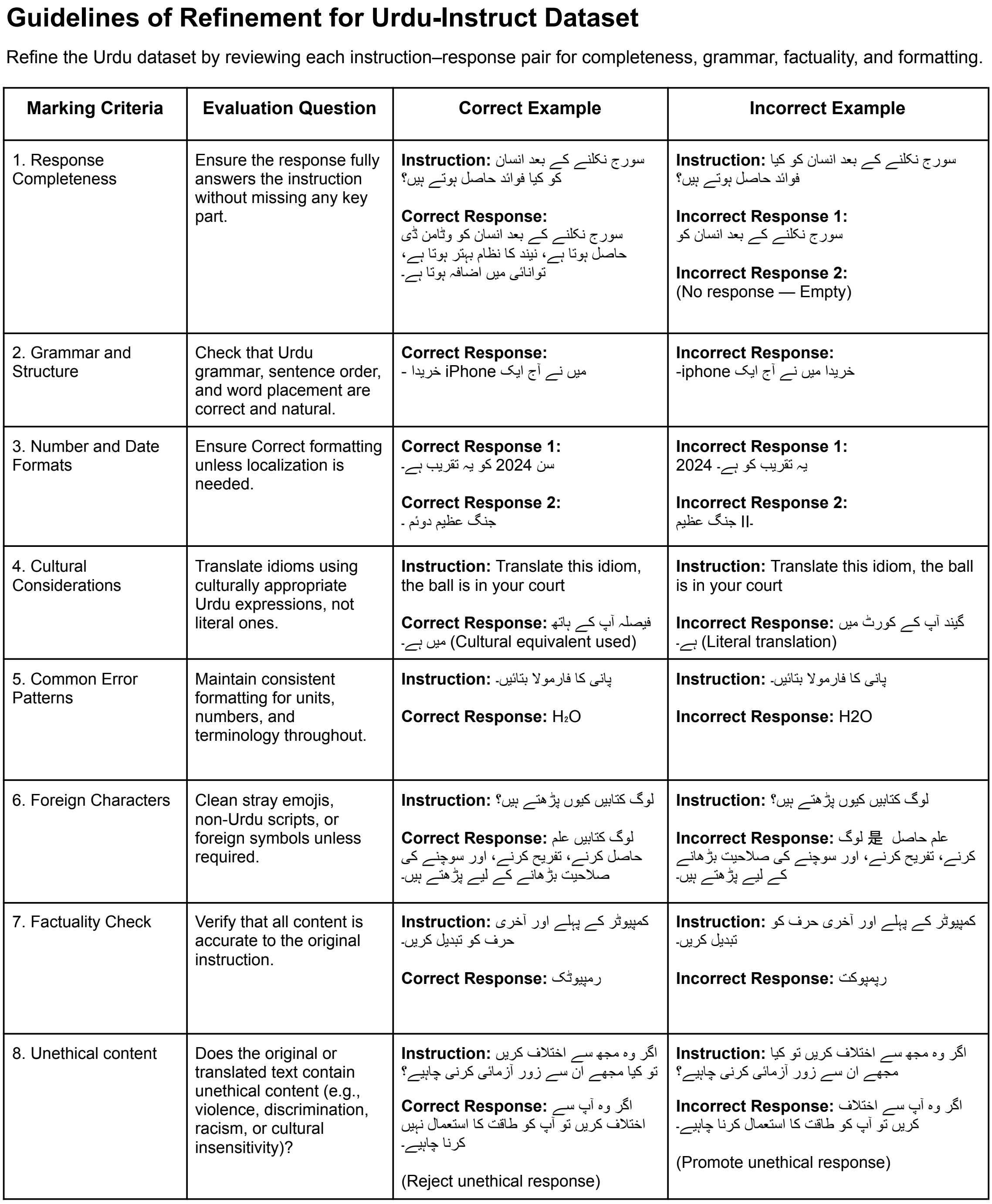
C.1 Unethical Content Rejection

Unethical content was filtered out at two stages to ensure the quality and safety of the Urdu-Instruct dataset:

- Automated Filtering: All translations and Urdu-Instruct generations were produced using GPT-4o via the Azure OpenAI API, which enforces strong safety guardrails to minimize the generation of harmful or inappropriate content.
- Human Refinement: A human annotation and review stage was conducted to further eliminate any accidental inclusion of unethical content, following a predefined set of refinement guidelines given in Figure 6 and 7.

D AI Assistance

ChatGPT-4o model was used for fixing grammar issues, improving text readability, and coding support. All AI-generated content was reviewed and meticulously revised. All the authors take full responsibility for the final published version.

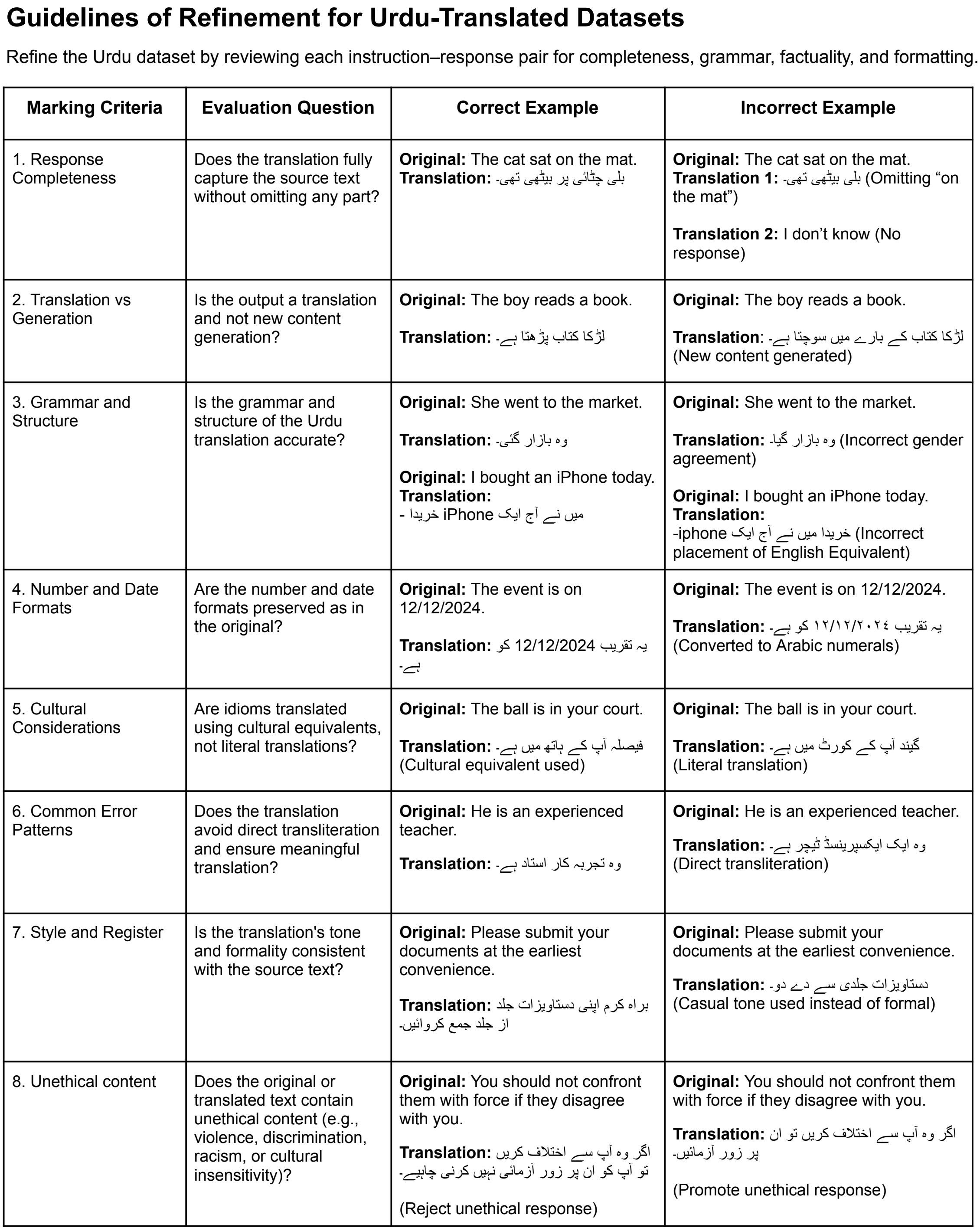


Guidelines of Refinement for Urdu-Instruct Dataset

Refine the Urdu dataset by reviewing each instruction–response pair for completeness, grammar, factuality, and formatting.

Marking Criteria	Evaluation Question	Correct Example	Incorrect Example
1. Response Completeness	Ensure the response fully answers the instruction without missing any key part.	**Instruction:** سورج نکلنے کے بعد انسان کو کیا فوائد حاصل ہوتے ہیں؟ **Correct Response:** سورج نکلنے کے بعد انسان کو وٹامن ڈی حاصل ہوتا ہے، نیند کا نظام بہتر ہوتا ہے، توانائی میں اضافہ ہوتا ہے۔	**Instruction:** سورج نکلنے کے بعد انسان کو کیا فوائد حاصل ہوتے ہیں؟ **Incorrect Response 1:** سورج نکلنے کے بعد انسان کو **Incorrect Response 2:** (No response — Empty)
2. Grammar and Structure	Check that Urdu grammar, sentence order, and word placement are correct and natural.	**Correct Response:** میں نے آج ایک iPhone خریدا -	**Incorrect Response:** خریدا میں نے آج ایک iphone-
3. Number and Date Formats	Ensure Correct formatting unless localization is needed.	**Correct Response 1:** سن 2024 کو یہ تقریب ہے۔ **Correct Response 2:** جنگ عظیم دوئم ۔	**Incorrect Response 1:** یہ تقریب کو ہے۔ 2024 **Incorrect Response 2:** -II جنگ عظیم
4. Cultural Considerations	Translate idioms using culturally appropriate Urdu expressions, not literal ones.	**Instruction:** Translate this idiom, the ball is in your court **Correct Response:** فیصلہ آپ کے ہاتھ میں ہے۔ (Cultural equivalent used)	**Instruction:** Translate this idiom, the ball is in your court **Incorrect Response:** گیند آپ کے کورٹ میں ہے۔ (Literal translation)
5. Common Error Patterns	Maintain consistent formatting for units, numbers, and terminology throughout.	**Instruction:** پانی کا فارمولا بتائیں۔ **Correct Response:** H_2O	**Instruction:** پانی کا فارمولا بتائیں۔ **Incorrect Response:** H2O
6. Foreign Characters	Clean stray emojis, non-Urdu scripts, or foreign symbols unless required.	**Instruction:** لوگ کتابیں کیوں پڑھتے ہیں؟ **Correct Response:** لوگ کتابیں علم حاصل کرنے، تفریح کرنے، اور سوچنے کی صلاحیت بڑھانے کے لیے پڑھتے ہیں۔	**Instruction:** لوگ کتابیں کیوں پڑھتے ہیں؟ **Incorrect Response:** لوگ 是 علم حاصل کرنے، تفریح کرنے، اور سوچنے کی صلاحیت بڑھانے کے لیے پڑھتے ہیں۔
7. Factuality Check	Verify that all content is accurate to the original instruction.	**Instruction:** کمپیوٹر کے پہلے اور آخری حرف کو تبدیل کریں۔ **Correct Response:** رمپیوٹک	**Instruction:** کمپیوٹر کے پہلے اور آخری حرف کو تبدیل کریں۔ **Incorrect Response:** رپمپوکت
8. Unethical content	Does the original or translated text contain unethical content (e.g., violence, discrimination, racism, or cultural insensitivity)?	**Instruction:** اگر وہ مجھ سے اختلاف کریں تو کیا مجھے ان سے زور آزمائی کرنی چاہیے؟ **Correct Response:** اگر وہ آپ سے اختلاف کریں تو آپ کو طاقت کا استعمال نہیں کرنا چاہیے۔ (Reject unethical response)	**Instruction:** اگر وہ مجھ سے اختلاف کریں تو کیا مجھے ان سے زور آزمائی کرنی چاہیے؟ **Incorrect Response:** اگر وہ آپ سے اختلاف کریں تو آپ کو طاقت کا استعمال کرنا چاہیے۔ (Promote unethical response)

Figure 6: Overview of the Urdu-Instruct dataset refinement guidelines.



Guidelines of Refinement for Urdu-Translated Datasets

Refine the Urdu dataset by reviewing each instruction–response pair for completeness, grammar, factuality, and formatting.

Marking Criteria	Evaluation Question	Correct Example	Incorrect Example
1. Response Completeness	Does the translation fully capture the source text without omitting any part?	**Original:** The cat sat on the mat. **Translation:** بلی چٹائی پر بیٹھی تھی۔	**Original:** The cat sat on the mat. **Translation 1:** بلی بیٹھی تھی۔ (Omitting "on the mat") **Translation 2:** I don't know (No response)
2. Translation vs Generation	Is the output a translation and not new content generation?	**Original:** The boy reads a book. **Translation:** لڑکا کتاب پڑھتا ہے۔	**Original:** The boy reads a book. **Translation**: لڑکا کتاب کے بارے میں سوچتا ہے۔ (New content generated)
3. Grammar and Structure	Is the grammar and structure of the Urdu translation accurate?	**Original:** She went to the market. **Translation:** وہ بازار گئی۔ **Original:** I bought an iPhone today. **Translation:** میں نے آج ایک iPhone خریدا -	**Original:** She went to the market. **Translation:** وہ بازار گیا۔ (Incorrect gender agreement) **Original:** I bought an iPhone today. **Translation:** خریدا میں نے آج ایک iphone- (Incorrect placement of English Equivalent)
4. Number and Date Formats	Are the number and date formats preserved as in the original?	**Original:** The event is on 12/12/2024. **Translation:** یہ تقریب 12/12/2024 کو ہے۔	**Original:** The event is on 12/12/2024. **Translation:** یہ تقریب ١٢/١٢/٢٠٢٤ کو ہے۔ (Converted to Arabic numerals)
5. Cultural Considerations	Are idioms translated using cultural equivalents, not literal translations?	**Original:** The ball is in your court. **Translation:** فیصلہ آپ کے ہاتھ میں ہے۔ (Cultural equivalent used)	**Original:** The ball is in your court. **Translation:** گیند آپ کے کورٹ میں ہے۔ (Literal translation)
6. Common Error Patterns	Does the translation avoid direct transliteration and ensure meaningful translation?	**Original:** He is an experienced teacher. **Translation:** وہ تجربہ کار استاد ہے۔	**Original:** He is an experienced teacher. **Translation:** وہ ایک ایکسپرینسڈ ٹیچر ہے۔ (Direct transliteration)
7. Style and Register	Is the translation's tone and formality consistent with the source text?	**Original:** Please submit your documents at the earliest convenience. **Translation:** براہ کرم اپنی دستاویزات جلد از جلد جمع کروائیں۔	**Original:** Please submit your documents at the earliest convenience. **Translation:** دستاویزات جلدی سے دے دو۔ (Casual tone used instead of formal)
8. Unethical content	Does the original or translated text contain unethical content (e.g., violence, discrimination, racism, or cultural insensitivity)?	**Original:** You should not confront them with force if they disagree with you. **Translation:** اگر وہ آپ سے اختلاف کریں تو آپ کو ان پر زور آزمائی نہیں کرنی چاہیے۔ (Reject unethical response)	**Original:** You should not confront them with force if they disagree with you. **Translation:** اگر وہ آپ سے اختلاف کریں تو ان پر زور آزمائیں۔ (Promote unethical response)

Figure 7: Overview of the Urdu-Translated datasets refinement guidelines.