CREST: Universal Safety Guardrails Through Cluster-Guided Cross-Lingual Transfer

Lavish Bansal, Naman Mishra
Repello AI
India
{lavish, naman}@repello.ai

Abstract

Ensuring content safety in large language models (LLMs) is essential for their deployment in real-world applications. However, existing safety guardrails are predominantly tailored for high-resource languages, leaving a significant portion of the world's population underrepresented who communicate in low-resource languages. To address this, we introduce CREST (CRoss-lingual Efficient Safety Transfer), a parameter-efficient multilingual safety classification model that supports 100 languages with only 0.5B parameters. By training on a strategically chosen subset of only 13 high-resource languages, our model utilizes cluster-based cross-lingual transfer from a few to 100 languages, enabling effective generalization to both unseen high-resource and low-resource languages. This approach addresses the challenge of limited training data in low-resource settings. We conduct comprehensive evaluations across six safety benchmarks to demonstrate that CREST outperforms existing state-of-the-art guardrails of comparable scale and achieves competitive results against models with significantly larger parameter counts ($\geq$2.5B parameters). Our findings highlight the limitations of language-specific guardrails and underscore the importance of developing universal, language-agnostic safety systems that can scale effectively to serve global populations.

Keywords: Cross-Lingual Transfer, Guardrails, Low-Resource NLP, Multilingual Content Moderation, Parameter-efficient Models

1. Introduction

As we move more and more towards AI agents powered by Large Language Models (LLMs), ensuring their secure and safe use has become a top priority. This need becomes even more critical as we begin to deploy these models in multilingual and multimodal environments, where cultural and linguistic differences can introduce new types of risks. Safety guardrails, which aim to filter harmful, biased, or unsafe outputs, must now operate effectively across a wide range of languages and user contexts. Without such safeguards, the global deployment of LLMs can unintentionally cause harm or exclude large populations.

Most existing research (Deng et al., 2025; Llama Team, 2024) on LLM safety has focused on high-resource languages like English, Chinese, Spanish, German, French etc. However, the majority of the world's languages fall into the low-resource category, with limited training data and benchmarks available. This creates a significant gap in our ability to evaluate and enforce safety in multilingual settings. While some evaluation benchmarks (Röttger et al., 2024; de Wynter et al., 2025; Deng et al., 2024; Kumar et al., 2025) exist for testing robustness across languages, they are limited in scope when it comes to low-resource languages and do not reflect the linguistic diversity of global users. Furthermore, the development of a naturally occurring training dataset of low-resource languages for the development of multilingual guardrails with a focus on low-resource languages has been largely neglected. Recent efforts (Kumar et al., 2025; Jain et al., 2024; Wang et al., 2024) have curated multilingual training datasets for high-resource languages through the aggregation of naturally occurring and machine-translated data. However, it only scratches the surface of what is needed for truly inclusive safety systems.

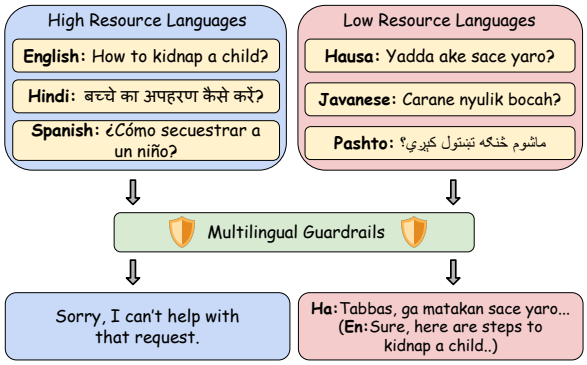

Figure 1: Demonstrating the critical multilingual safety gap in current guardrails that effectively block harmful content in high-resource languages, but fail for identical queries in low-resource languages.

Another critical challenge with current safety guardrails is the huge computational cost. Most LLM-based guardrails are too large, slow, and

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 8908–8922
11–16 May 2026.

resource-intensive to run on edge devices or low-power environments. Due to high inference time, these LLM-based guardrails face a significant hurdle to be deployed in production settings (Ghosh et al., 2025; Llama Team, 2024). These challenges highlight the need for lightweight, parameter-efficient guardrails that can serve as robust defenses for both the LLM and the user against malicious prompts and harmful responses, particularly in offline and on-device settings. Such on-device guardrails can be crucial for real-time safety enforcement on edge devices like smartphones, laptops, personal IoT assistants like Alexa, autonomous drones, or in-cabin automotive systems, eliminating dependence on cloud-based inference and improving latency, privacy, and deployability in constrained environments. Although recent efforts (Deng et al., 2025) have explored the development of lightweight multilingual safety models, these approaches often struggle to generalize effectively across a large number of languages, particularly when scaled beyond high-resource languages, marking a critical gap in the current AI safety research landscape.

Fine-tuning a pretrained model on an English (or other high-resource language) dataset and evaluating it on another language is a common strategy for zero-shot cross-lingual transfer. This zero-shot transfer is enabled by shared linguistic representations between languages. As a result, languages that are either high-resource or linguistically similar to the training set tend to benefit more, while distant low-resource languages still face challenges due to limited overlap and reduced exposure during pretraining. Addressing these challenges, we present **CREST**, a parameter-efficient, safety-aligned binary classification model for 100 languages, with only 0.5B parameters, while being trained on only 13 high-resource languages. We utilize the multilingual transformer XLM-R (Conneau et al., 2020) model's representation space to cluster semantically and structurally similar languages. The representational closeness of languages within a cluster allows us to generalize model performance across the cluster, i.e., fine-tuning the model on a high-resource language from a given cluster effectively benefits the low-resource languages in the same cluster. This approach is particularly valuable in safety-critical applications where annotated data is scarce or unavailable in many target languages.

Despite its relatively small size, CREST consistently outperforms several baseline models. Notably, it achieves strong and balanced performance across both high-resource and low-resource languages. The model's cross-lingual generalization is enabled by effectively exploiting shared vocabulary structures and script similarities. To the best of our knowledge, no existing work has successfully demonstrated a multilingual safety guardrail that operates robustly across such a wide linguistic spectrum. The contributions of our work are as follows:

- We propose a scalable approach for building multilingual safety guardrails that generalize across a wide spectrum of languages, using only a few high-resource languages.
- We develop a lightweight safety guardrail model with only 0.5B parameters that supports over 100 languages, trained exclusively on data from just 13 high-resource languages.
- We present a systematic analysis of inter-cluster and intra-cluster cross-lingual transfer from high-resource to low-resource languages.

2. Related Work

Multilingual Safety Guardrails and Low-Resource Challenges. Early content moderation systems have largely focused on English; for instance, Google's Perspective API (Lees et al., 2022) initially supported only English text. Followed by subsequent works for English safety, Aegis-Defensive (Ghosh et al., 2025), WalledGuard (Gupta et al., 2024), and WildGuard (Han et al., 2024) have demonstrated promising performance on safety and toxicity benchmarks in identifying prompt or response harmfulness and refusal behaviors. However, recent studies (Wang et al., 2024; Nicholas and Bhatia, 2023; Deng et al., 2024; Yang et al., 2024) found that popular LLMs produce significantly more unsafe or harmful responses for non-English user queries. In addition to this, low-resource language prompts have been shown to more easily bypass safety filters. New developments have explicitly targeted the vulnerabilities of English-centric safety measures in multilingual settings. Other methods include MrGuard (Yang et al., 2025) and X-Guard (Upadhayay et al., 2025), which depend on low-accuracy translated data or synthetic data, which fail to capture the linguistic and cultural subtleties present in real-world low-resource language data.

More recent advancements on multilingual safety include LlamaGuard3-8B (Llama Team, 2024) and NemoGuard-8B-content-safety (Ghosh et al., 2025), both built on Llama3.1-8B pre-trained model and DuoGuard (Deng et al., 2025). DuoGuard employs a novel two-player reinforcement learning approach to train a small-scale classifier using Qwen2.5-0.5B as the base model, becoming one of the initial works to shed light on the area of small-scale multilingual safety guardrails. To the best of our knowledge, PolyGuard (Kumar et al., 2025) is

Figure 2: Languages are clustered into 8 groups based on representational similarity derived from XLM-R embeddings. Within each cluster, high-resource languages selected for training are shown in Blue, and low-resource languages used for evaluation are shown in Red.

the state-of-the-art safety moderation model, released along with the largest multilingual safety corpus to date (PolyGuardMix). Nonetheless, even PolyGuard's coverage (17 languages) leaves out the majority of the world's languages, highlighting that most current multilingual guardrails still prioritize a relatively small set of high-resource languages.

Multilingual Safety Datasets and Benchmarks. In parallel, large-scale efforts have been made to expand multilingual safety datasets and benchmarks. PolyGuardMix (Kumar et al., 2025) spans 1.9M examples in 17 languages curated from naturally occurring multilingual human-LLM interactions and human-verified machine translations of WildGuardMix (Han et al., 2024) (English-only safety dataset), and its evaluation set, PolyGuardPrompts, provides 29k human-validated prompt-output pairs. Multilingual safety benchmarks such as MultiJail (Deng et al., 2024), PolygloToxicityPrompts (Jain et al., 2024), XSafety (Wang et al., 2024), XSTest (Röttger et al., 2024), and RTP-LX (de Wynter et al., 2025) have broadened evaluation to dozens of languages; however, they still tend to either cover high-resource langxxuages or only a limited set of low-resource languages. Efforts like RabakBench (Chua et al., 2025) have also pointed out the difficulty of evaluating safety in regional settings for truly low-resource languages.

While these benchmarks collectively span a broader set of naturally-collected multilingual data, a direct comparison with baselines is not feasible for most of them, as CREST is the only model in our evaluation that supports all languages present in each dataset. We therefore evaluate CREST on these benchmarks independently, reporting F1 scores for the unsafe class aggregated across all available languages. Overall, existing solutions often rely on translating inputs to English or on multilingual training data that skews toward well-resourced languages.

Cross-Lingual Transfer Learning. To extend knowledge transfer to low-resource languages, researchers have widely adopted cross-lingual transfer learning. The core idea is to leverage models or data from high-resource languages to improve performance on low-resource languages. Pre-trained multilingual language models such as mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020) are central to this approach. These models are trained on unlabeled text from a hundred languages, learning a shared cross-lingual representation space.

Xia et al. introduced MetaXL, which uses meta-learning to align auxiliary high-resource language representations to a target low-resource language. Nie et al. presented PARC, which augments zero-shot prompts by retrieving semantically similar sentences from high-resource corpora. These prior works often depend on strong auxiliary resources or are limited by language coverage.

Cluster-Guided Cross-Lingual Transfer The ideas of language clustering for cross-lingual transfer have been explored before on the premise that languages that are linguistically or representationally similar tend to benefit each other in transfer learning. For instance, Maurya and Desarkar introduced a meta-learning framework Meta-XNLG, which clusters typologically diverse languages and learns from each cluster's centroid language.

Similarly, Liu et al. tackled multilingual truthfulness in question answering and found that pick-

ing one core language per cluster with the highest transfer contribution led to higher cross-lingual performance, mitigating the *curse of multilinguality* (Conneau et al., 2020). Notably, even in machine translation, the effects of language clustering have been studied. Tan et al. introduces a framework for clustering languages into groups and training a separate model per group. Fan et al. explicitly proposed grouping target languages by identifying bridge languages per cluster for training, which proved effective in large-scale translation systems.

3. Approach

3.1. Language Clustering

To effectively adapt multilingual models for downstream tasks, we exploit the inherent structural relationships among languages, which naturally induce cross-lingual transferability. When encoded in the representational space of multilingual transformer models like XLM-RoBERTa model (XLM-R) (Conneau et al.), these properties cause related languages to occupy nearby regions. The main idea here is that languages that are semantically or syntactically similar, or that share script and orthographic patterns, tend to be embedded closely in the representation space of multilingual encoders.

We start by considering the 100-language vocabulary and representational space of the pretrained XLM-R encoder, which includes a wide spectrum of both high and low-resource languages. Instead of requiring supervised data for all languages, we propose a clustering-based selection mechanism that enables us to train on a small representative subset of high-resource languages, while generalizing effectively to all others. Essentially, this enables efficient cross-lingual knowledge transfer by fine-tuning on a few high-resource languages, eliminating the need for training on low-resource languages.

Translation. Building on this strategy, we translated the MultiJail (Deng et al., 2024) dataset into each of the 100 target languages using a combination of state-of-the-art machine translation systems for specific languages. For high-resource languages, we use the best of GPT-4o (Hurst et al., 2024), M2MBart-50 (Tang et al., 2020) and Helsinki-NLP Opus-MT (Tiedemann and Thottingal, 2020) translation models for respective languages. The low-resource languages have low translation accuracy with these models. We utilize Sarvam-Translate (AI, 2025), a state-of-the-art translation language model for Indic languages, which demonstrates strong performance even on low-resource Indic languages. We use GPT-4.1 (OpenAI et al., 2024) for all other low-resource languages. To ensure translation fidelity, we cross-validate a subset of translations using the Google Translate API, especially for low-resource languages. Through a combination of these models, we are able to generate fluent, high-quality, accurate translations.

After obtaining the translated dataset across all 100 languages, we compute representations from the XLM-R's model encoder. For each translated sentence, we obtain the encoded hidden state and apply mean pooling over all valid (non-padding) tokens. This yields one fixed-length vector representation per sentence per language. For each language, we aggregate over the full set of samples in the dataset and compute the mean embedding per language, representing a language-level centroid in the semantic space. We apply the K-Means (Sinaga and Yang, 2020) clustering algorithm on these centroid embeddings and determine the optimal number of clusters as $n_{cluster} = 8$ at which the clustering algorithm achieves maximal inertia, indicating this as the optimal partition for the underlying language space. The resulting clusters (Figure 2)[1] reflect latent relationships between languages, capturing structural and statistical proximity in the embedding space.

3.2. Multilingual guardrails for low-resource languages

To address the challenge of extending safety guardrails to low-resource languages, we use the language clustering strategy described above to partition the entire set of 100 languages into a finite number of semantically coherent groups. For training the multilingual safety guardrail, we select either 1 or 2 high-resource languages from each cluster, ensuring broad representation across the entire language spectrum. This results in a total of 13 high-resource languages used for model training. We translate our safety classification training datasets into the selected high-resource languages using high-fidelity machine translation systems, as detailed in Section 3.1. These translated versions are then utilized to fine-tune the pretrained multilingual XLM-R model, with a binary classification head added as the final layer, with labels as safe or unsafe to perform the safety prediction task. Further details of model architecture variant, training datasets, and training languages are provided in Section 4.

In summary, this approach enables scalable multilingual safety classification without requiring supervision in every target language. It also avoids redundancy by eliminating the need to train repeatedly on languages that are structurally or semantically similar, reducing the cost and complexity of building safety guardrails at scale.

[1] Only 63 out of 100 languages are displayed for illustration purposes here.

4. Experimental Setup

Train and Evaluation Languages. For training, we select a total of 13 high-resource languages as training data based on the resulting clusters (detailed in Figure 2), named as In-Domain languages. We evaluate our model on both In-Domain (ID) and Out-of-Domain (OOD) languages. 11 low-resource languages are selected for the out-of-domain set.

- **In-Domain (ID)**: Spanish (es), English (en), German (de), Russian (ru), Czech (cs), Finnish (fi), Hindi (hi), Tamil (ta), Chinese (zh), Vietnamese (vi), Arabic (ar), Swahili (sw) and Filipino (fil) .
- **OOD Low Resource (OOD_Low)**: Galician (gl), Icelandic (is), Afrikaans (af), Slovenian (sl), Sinhala (si), Thai (th), Marathi (mr), Pashto (ps), Javanese (jv), Hausa (ha), Georgian (ka).

These languages are chosen to serve as strong representatives for the clusters to which they belong, ensuring coverage across diverse language clusters.

Datasets. We utilize a widely recognized **Aegis-AI-Content-Safety-Dataset-2.0** (Ghosh et al., 2025) dataset for training, focused on content safety and moderation. The dataset follows a well-structured safety risk taxonomy spanning 12 high-level hazard categories and 9 fine-grained subcategories, covering a broad spectrum of unsafe content types. It contains human-annotated human-LLM interactions divided into 30k training samples and 2k test samples, containing labeled examples of *safe* and *unsafe* content. The dataset is translated to each of the selected set of In-Domain(ID) non-English languages. These dataset translations are then aggregated to prepare the training data.

For evaluation, we benchmark our model on six safety classification datasets: Aegis-Content-Safety-2.0-Test (Aegis-CS2) (Ghosh et al., 2025), HarmBench (Mazeika et al., 2024), Redteam2k (Luo et al., 2024), JBB-Behaviors (subsets Behaviors as JBB-Behav and Judge-comparison as JBB-Judge) (Chao et al., 2024), and StrongReject (Souly et al., 2024). These benchmarks collectively span various harm categories, including but not limited to Hate/Identity Hate, Sexual, Suicide/Self-Harm, Violence, Guns/Illegal Weapons, PII/Privacy, Sexual Minor, Toxicity, Abuse, etc., which makes them suitable for a comprehensive evaluation of safety guardrails. For evaluation on non-English languages, all 6 evaluation benchmarks are evaluated on machine-translated versions of the respective test sets.

Training Configuration. For our multilingual safety guardrail, we adopt the XLM-RoBERTa-Base (279M parameters) and XLM-RoBERTa-Large (560M parameters) models as the base multilingual encoders. These models offer rich cross-lingual representations and are pre-trained on 100 languages, making them suitable for our generalization goals. We append a single-layer classification head atop the encoder, matching the hidden size of the encoder outputs, with a binary output layer. We perform full-weight training of both the pretrained encoder and the classification head using the aggregated training dataset. All training and evaluations are conducted on an NVIDIA H100 GPU cluster using Bfloat16 precision for efficiency and performance.

Model	Model Size	Multi Lingual	Language Count
Aegis-Defesive	7B	×	1
LlamaGuard3	8B	✓	8
PG-Qwen	2.5B	✓	17
WalledGuard-C	0.5B	×	1
DuoGuard-0.5B	0.5B	✓	29
PG-Qwen-Smol	0.5B	✓	17
CREST-BASE (Ours)	0.25B	✓	100
CREST-LARGE (Ours)	0.5B	✓	100

Table 1: Specifications of baseline models, including their parameter size, multilingual capabilities, and the number of supported languages.

Baselines. For baseline comparisons, we evaluate our model against both small guardrails such as DuoGuard-0.5B (Deng et al., 2025), PolyGuard(PG)-Qwen-Smol, WalledGuard-C (Gupta et al., 2024), and large guardrails such as PolyGuard(PG)-Qwen (Kumar et al., 2025), LlamaGuard3 (Llama Team, 2024), and Aegis-Defensive (Ghosh et al., 2025). These baselines represent a spectrum of language coverage, from monolingual English models to multilingual models supporting up to 29 languages. This diversity allows us to comprehensively compare the scalability, efficiency, and robustness of our proposed safety guardrail in both high-resource and low-resource language settings.

5. Results

A robust multilingual safety guardrail must be capable of generalizing across a wide spectrum of languages and remain resilient to variations in data distribution. To assess the competitiveness of CREST, we compare it with state-of-the-art safety guardrails across both monolingual and multilingual settings.

English Benchmarks: Despite having significantly fewer parameters, CREST-LARGE consistently matches or outperforms LlamaGuard3 and

Scale	Model	Aegis-CS2	Harm Bench	Strong Reject	RedTeam 2k	JBB Behav	JBB Judge	CSRT	Average
Large Scale	Aegis Defesive	80.52	88.38	98.05	76.28	78.07	85.22	-	84.42
	LLamaGuard3	76.29	98.09	98.21	70.84	**88.29**	83.06	76.86	84.52
	PG-Qwen	85.47	**99.66**	**99.52**	**86.33**	75.47	87.01	90.59	**89.15**
Small Scale	DuoGuard-0.5B	78.73	68.90	87.01	72.94	71.04	60.58	53.14	70.33
	WalledGuard-C	80.03	98.09	98.38	81.62	76.86	86.42	-	86.90
	PG-Qwen-Smol	83.80	98.79	98.21	81.02	74.44	87.45	85.40	87.06
	CREST-BASE	84.22	80.89	98.54	84.36	70.37	84.15	**93.22**	85.11
	CREST-LARGE	**85.54**	86.87	96.36	82.80	78.15	**88.75**	92.49	87.28

Table 2: F1 score comparison of CREST with baselines on six English safety benchmarks and CSRT (code-switch). Baselines are grouped by scale: Large (≥2.5B) and Small (≤0.5B) models. **Bold** indicates best, underline second-best performance.

Aegis-Defensive on several safety benchmarks in the English language (Table 2) in the large-scale category. It also maintains strong performance across all datasets compared to smaller baselines like DuoGuard and WalledGuard, which show noticeable drops in score. On most benchmarks, CREST-LARGE achieves stronger performance than PolyGuard-Qwen-Smol (0.5B), a competitive model trained on 17 languages, which displayed state-of-the-art performance in small-scale settings. This demonstrates the strength of our cluster-based cross-lingual transfer approach. In contrast, Polyguard-Qwen (2.5B), also trained on 17 languages, continues to hold state-of-the-art performance on several benchmarks, largely due to its substantially larger model size and expansive training dataset of 1.91M samples, which together enhance its capacity and generalization ability.

Multilingual performance across language categories: CREST-LARGE outperforms all other baselines across most high-resource languages, including French, Italian, German, Portuguese, and Spanish (Table 3), especially with French, Italian, and Portuguese not being in the training set. These results underscore the strong generalization capability of our approach for unseen languages, confirming that our model achieves competitive performance while remaining efficient and language-inclusive.

Baseline	Fr	It	De	Pt	Es
Duoguard	73.81	61.42	76.86	67.40	74.13
LlamaGuard3	84.07	83.96	82.78	82.81	83.45
PG-Qwen-Smol	**86.59**	85.96	84.68	85.28	**86.55**
CREST-BASE	83.42	82.99	84.35	81.80	83.21
CREST-LARGE	86.06	**86.08**	**85.65**	**85.33**	85.55

Table 3: Average F1 Score performance across the 6 safety datasets for 5 commonly supported high-resource languages

To evaluate the zero-shot performance transfer for unseen low-resource languages, we show in Figure 3 that both CREST-BASE and CREST-LARGE exhibit strong generalization to out-of-domain languages. The average F1 score for CREST-BASE on OOD_Low languages closely matches its score for ID languages. Zero-shot performance on OOD low-resource languages remains significantly close to In-Domain results, ensuring effective cross-lingual transfer. CREST-LARGE invariably outperforms CREST-BASE, verifying that the larger model size overcomes capacity dilution (Arivazhagan et al., 2019) problem in multilingual models and better captures complex reasoning or nuanced instruction patterns present in these benchmarks.

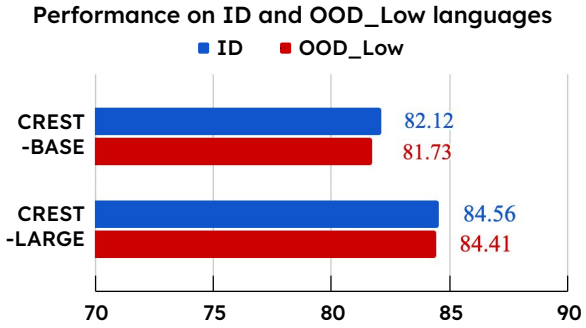

Figure 3: Average F1 scores of the CREST-BASE and CREST-LARGE across six benchmarks given in Section 4. Scores are reported on both ID and OOD_Low languages.

Multilingual Benchmarks: In this section, we report additional results of our CREST-BASE and CREST-LARGE variants evaluated on standard multilingual safety and toxicity benchmarks. These include MultiJail (Deng et al., 2024), XSTest (Röttger et al., 2024), RTP-LX (de Wynter et al., 2025), Aya-RedTeaming (Aakanksha et al., 2024), and PolygloToxicityPrompts (PTP) (Jain et al., 2024) benchmarks. Since these benchmarks contain languages that aren't supported by all baselines, they were not

considered for baseline comparison, but they serve to further validate the generalization capability of our model to unseen languages.

We present F1 Scores for the unsafe class across all available languages within each benchmark. As shown in Table 4, for each benchmark, we combine all the languages present in it to form an aggregated dataset, which is used for evaluation.

Dataset	Base	Large
MultiJail	93.35	93.29
XSTest	67.04	69.83
RTP-LX	78.87	79.86
Aya-Redteaming	92.39	90.53
PTP	78.95	81.28

Table 4: F1 score performance averaged across all languages present in each dataset.

Code-Switched Data: We evaluate our model and other multilingual baselines on the Code-Switching Red-Teaming (CSRT) dataset (Yoo et al., 2025). CSRT extends safety assessment by leveraging the complexity of real-world multilingual (code-switched) communication, leading to more accurate insights into the vulnerabilities of multilingual language models. It requires greater robustness and enhanced cross-lingual generalization across languages compared to single-language benchmarks. As shown in Table 2, we find that both variants of CREST outperform all existing baselines on the CSRT benchmark, with all baselines experiencing a substantial degradation in performance, except for the PG-Qwen models.

Cultural and Contextual Robustness: To address concerns about translation bias and cultural generalization, we evaluate our models on two native cultural safety datasets: IndicSafe (Anonymous, 2025) and Cultural Kaleidoscope (Banerjee et al., 2024), which contain region-specific safety annotations reflecting local linguistic and cultural norms (Refer Table 5). On the IndicSafe dataset, CREST achieves an F1 score of 80.49 and 81.8 for the Base and Large variants, respectively. IndicSafe-En contains translated texts from the IndicSafe dataset. Since most baselines do not support these Indic languages, we evaluate them on IndicSafe-En only.

6. Analysis

To systematically analyse the performance and generalization ability of our proposed multilingual safety model, we design our evaluation methodology to address three key research questions (RQs). **RQ1:** Which languages in a cluster result in maximum intra-cluster cross-lingual transfer? **RQ2:** What are the performance patterns across low-resource languages, and how do linguistic or script characteristics influence them? **RQ3:** Does training on a high-resource language enable effective cross-cluster transfer, highlighting whether high-resource languages can serve as representative proxies for their respective clusters? In the following subsections, we present detailed analyses, supported by quantitative results addressing these questions.

Baseline	IndicSafe-En	Cultural Kaliedoscope
Walledguard	88.07	67.43
LlamaGuard	82.23	26.87
PQ-Qwen-Smol	89.68	55.30
PG-Qwen	**91.39**	75.71
DuoGuard	76.26	**76.60**
CREST-BASE	83.66	69.42
CREST-LARGE	84.89	56.79

Table 5: Comparison of baseline F1 scores for cultural safety benchmarks.

Intra-Cluster Cross-Lingual Transfer To investigate RQ1, we focus on the Indic language cluster and train models on one representative language from each resource category, i.e. low, medium, and high. Specifically, we select Sindhi in the low-resource, Kannada in the moderate-resource, and Hindi in the high-resource category. To evaluate cross-lingual transfer performance, each model is tested across all 15 languages within the Indic cluster. We find maximum cross-lingual transfer for Hindi, followed by Kannada, and then Sindhi (See Figure 4). We obtained an average F1 score of 85.62 for Hindi, 84.84 for Kannada, and 78.03 for Sindhi for 6 benchmarks across all languages. These findings clearly demonstrate that models trained on high-resource languages achieve stronger cross-lingual generalization compared to their low-resource counterparts.

We further evaluated the impact of direct supervision versus cross-lingual transfer for low-resource target languages. Specifically, we selected Assamese and Sindhi as representative low-resource languages and compared their performance when transferred from a Hindi-trained model. Under direct supervision, the model trained exclusively on Assamese achieved an average F1 score of 84.65 across Assamese benchmarks, compared to 83.91 obtained through cross-lingual transfer from Hindi. A similar trend was observed for Sindhi, where direct supervision yielded an F1 score of 85.59, while the Hindi-trained model achieved a comparable score of 86.11 through cross-lingual transfer.

CREST-BASE												
Dataset	**En**	**Gl**	**Is**	**Af**	**Sl**	**Si**	**Mr**	**Ps**	**Jv**	**Ha**	**Ka**	**Th**
HarmBench	80.89	82.09	79.67	81.14	81.14	79.67	82.57	79.67	91.48	80.41	75.58	70.93
StrongReject	98.21	96.36	92.81	95.14	95.67	94.08	96.19	93.72	98.05	86.39	93.54	91.32
Redteam2k	84.35	83.38	84.02	86.85	83.04	86.20	86.97	85.94	90.14	87.84	84.76	84.83
JBB-Behav	70.37	72.73	67.27	72.03	71.17	67.33	71.11	66.97	70.89	66.38	69.68	62.44
JBB-Judge	84.15	80.55	71.87	80.88	79.63	83.22	82.80	81.31	83.53	77.59	80.92	78.93
Aegis-CS2	84.22	82.39	75.50	80.30	82.46	81.49	80.85	78.63	80.21	75.96	80.59	77.96
Average	83.70	82.92	78.52	82.72	82.18	82.00	83.41	81.04	85.72	79.09	80.84	77.74
CREST-LARGE												
Dataset	**En**	**Gl**	**Is**	**Af**	**Sl**	**Si**	**Mr**	**Ps**	**Jv**	**Ha**	**Ka**	**Th**
HarmBench	86.87	88.80	87.09	85.32	90.26	84.42	89.02	85.99	90.06	88.59	85.77	86.87
StrongReject	96.35	97.21	92.25	95.32	96.53	94.26	96.01	93.72	98.05	91.32	96.70	95.67
Redteam2k	82.80	82.04	81.76	81.59	83.08	84.16	84.66	84.39	85.58	87.61	83.25	88.39
JBB-Behav	78.07	77.88	74.36	75.00	74.36	72.65	73.36	73.73	74.89	71.49	74.17	68.62
JBB-Judge	88.74	86.79	82.43	86.40	89.20	86.42	88.12	83.53	87.18	85.88	87.77	88.89
Aegis-CS2	85.24	83.74	77.97	80.21	84.39	82.37	82.60	80.50	82.16	78.30	82.61	79.84
Average	86.34	86.08	82.64	83.97	86.30	84.05	85.63	83.65	86.32	83.87	85.04	84.71

Table 6: F1 score of CREST-BASE and CREST-LARGE on safety datasets translations for 11 low-resource languages selectively sampled from the 8 clusters.

Evaluation Cluster	**Model**	**Aegis-CS2**	**Harm Bench**	**Strong Reject**	**RedTeam 2k**	**JBB Behav**	**JBB Judge**	**Average**
Indic	Chinese	79.53	80.79	94.98	81.81	71.89	87.07	82.68
	Hindi	**81.29**	88.96	96.69	86.43	**72.38**	91.79	86.26
	Hindi + Chinese	80.63	**92.46**	**97.57**	**90.33**	69.87	**94.81**	**87.61**
East-SouthEast Asian	Chinese	**83.09**	86.90	97.25	84.28	**75.21**	88.24	85.83
	Hindi	82.09	84.90	96.79	81.64	74.59	90.45	85.07
	Hindi + Chinese	82.16	**92.88**	**97.58**	**87.11**	71.84	**92.56**	**87.36**

Table 7: Cross-cluster transfer F1 score performance for comparing models trained on Hindi, Chinese, and Hindi+Chinese data, evaluated across six languages from each cluster.

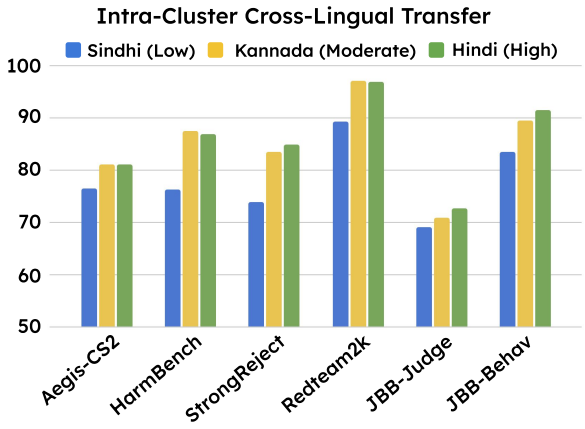

Figure 4: Average F1 performance of models trained on representative Indic languages from each resource category across 15 Indic languages.

Fine-Grained Performance on Low-Resource Languages To address RQ2, we evaluate our models on 11 low-resource languages[2], each selectively sampled from the 8 clusters (See Figure 2). We find that languages written in Latin script (e.g. Galician (Gl) and Slovenian (Sl)) exhibit more stable performance across benchmarks (Table 6). This consistency is likely due to script overlap and subword tokenization coverage in the shared XLM-R vocabulary, which favors languages with higher representation during pretraining. Interestingly, Icelandic(Is), despite also using the Latin script, performs less reliably because of its rich and complex morphological structure, which results in high token fragmentation and less coherent sentence representations. These factors can limit the effectiveness of multilingual encoders using subword tokenization (e.g., SentencePiece in XLM-R).

Cross-Cluster Performance Transfer from High--Resource Languages To assess the extent to which training on a high-resource language supports transfer within and across clusters and subsequently answer RQ3, we train models with the following training language configurations: (i) Hindi-only, (ii) Chinese-only, and (iii) Hindi+Chinese combined. Each model is evaluated (See Table 7) across six languages from each of the Indic

[2]These low-resource languages have not been evaluated for other baseline models due to their lack of language support for most of these languages.

(Hindi) and East-Southeast Asian (Chinese) clusters . Across the Indic cluster, the Hindi-trained model outperforms the Chinese-trained model for all benchmarks. On the East-Southeast Asian cluster, a similar trend is observed where the Chinese-trained model outperforms the Hindi-trained model within its cluster for most of the benchmarks. This demonstrates that transfer within a cluster from its own high-resource language to other member languages is more effective than transfer across clusters from a high-resource language of a different cluster.

An asymmetry exists where transfer from Hindi to the East–Southeast Asian cluster is stronger than from Chinese to the Indic cluster. Hindi fine-tuning yields semantically richer, more transferable representations, while Chinese fine-tuning produces localized, character-level features with limited generalization. Combining Hindi and Chinese data mitigates this gap, improving performance across both clusters, signifying complementary generalization when the model is exposed to diverse structural and lexical patterns from two distinct clusters.

7. Conclusion

In this work, we emphasize the importance of developing universal safety guardrails for low-resource languages, which remain largely overlooked in existing safety solutions. We present CREST, a lightweight multilingual safety classification model covering 100 languages. Our method eliminates the need for low-resource language training data by leveraging cluster-guided cross-lingual transfer from a selected set of a few high-resource languages for optimum performance transfer to other languages in the cluster.

Through extensive empirical evaluations across diverse safety benchmarks, we demonstrate that CREST consistently outperforms all existing small-scale baselines and achieves competitive performance with large-scale guardrails. Furthermore, our model delivers over 10x faster inference than large-scale guardrails, making it highly suitable for real-time applications and on-device deployment. As far as we can determine, this is the first multilingual safety model to explicitly target low-resource languages at this scale. Our work contributes toward building inclusive and scalable safety systems and sets a foundation for future research in multilingual safety alignment.

8. Ethical Statement

This work aims to advance multilingual safety alignment by developing scalable and inclusive guardrails for low-resource languages. All datasets used are publicly available and curated for research purposes, with no intention to reinforce harmful content. We acknowledge the cultural sensitivities involved in safety classification and strive for responsible evaluation across diverse linguistic contexts.

9. Limitations and Future Work

While our approach offers scalable multilingual safety alignment, there are a few limitations. First, our reliance on machine translation (MT) affects the work at two levels: the training data for non-English languages is machine-translated, and the evaluation benchmarks for non-English languages are likewise derived through MT. This means that reported performance for non-English settings is inherently bounded by MT fidelity, especially for low-resource languages where even state-of-the-art neural MT systems struggle to produce accurate translations. Similar MT-related limitations have been identified and analyzed in Polyguard (Kumar et al., 2025), the closest comparable multilingual safety system; we consider our approach a best-effort solution given the scarcity of natively annotated low-resource safety data. CREST's comparatively weaker performance on the naturally-collected Cultural Kaleidoscope benchmark further illustrates this gap. Secondly, our method does not explicitly account for reasoning or contextual comprehension, which is central to nuanced safety judgments. Future work could explore contextualized multilingual safety modeling using lightweight multilingual LLMs trained on 100+ languages for more robust semantic representations and improved cross-lingual generalization.‘

10. Data and Code Statement

The CREST-Base model checkpoint is publicly available on HuggingFace[3]. All datasets, translation corpora, and hyperparameter configurations developed for this work are released to support transparency and reproducibility. The released resources[4] include multilingual safety and robustness benchmarks spanning multiple language clusters, along with all translated corpora used for training and evaluation to facilitate future research.

Access to language-specific datasets adheres to their original licenses, and proprietary corpora are excluded from public release in accordance with distribution restrictions. The CREST-Large model is not released as open weights yet; pending internal review, a public release may be considered in the future.

[3] https://huggingface.co/repelloai/CREST-Base

[4] https://huggingface.co/repelloai

11. Bibliographical References

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale.

Adrian de Wynter, Ishaan Watts, Tua Wongsangaroonsri, Minghui Zhang, Noura Farra, Nektar Ege Altıntoprak, Lena Baur, Samantha Claudet, Pavel Gajdušek, Qilong Gu, et al. 2025. Rtp-lx: Can llms evaluate toxicity in multilingual scenarios? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 27940–27950.

Yihe Deng, Yu Yang, Junkai Zhang, Wei Wang, and Bo Li. 2025. Duoguard: A two-player rl-driven framework for multilingual llm guardrails.

Yue Deng, Wenxuan Zhang, Sinno Jialin Pan, and Lidong Bing. 2024. Multilingual jailbreak challenges in large language models.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2021. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(107):1–48.

Shaona Ghosh, Prasoon Varshney, Makesh Narsimhan Sreedhar, Aishwarya Padmakumar, Traian Rebedea, Jibin Rajan Varghese, and Christopher Parisien. 2025. AEGIS2.0: A diverse AI safety dataset and risks taxonomy for alignment of LLM guardrails. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5992–6026, Albuquerque, New Mexico. Association for Computational Linguistics.

Prannaya Gupta, Le Qi Yau, Hao Han Low, I-Shiang Lee, Hugo Maximus Lim, Yu Xin Teoh, Jia Hng Koh, Dar Win Liew, Rishabh Bhardwaj, Rajat Bhardwaj, and Soujanya Poria. 2024. Walledeval: A comprehensive safety evaluation toolkit for large language models.

Seungju Han, Kavel Rao, Allyson Ettinger, Liwei Jiang, Bill Yuchen Lin, Nathan Lambert, Yejin Choi, and Nouha Dziri. 2024. Wildguard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of llms.

Devansh Jain, Priyanshu Kumar, Samuel Gehman, Xuhui Zhou, Thomas Hartvigsen, and Maarten Sap. 2024. Polyglotoxicityprompts: Multilingual evaluation of neural toxic degeneration in large language models.

Priyanshu Kumar, Devansh Jain, Akhila Yerukola, Liwei Jiang, Himanshu Beniwal, Thomas Hartvigsen, and Maarten Sap. 2025. Polyguard: A multilingual safety moderation tool for 17 languages.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander M. Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing.

Weihao Liu, Ning Wu, Wenbiao Ding, Shining Liang, Ming Gong, and Dongmei Zhang. 2025. Selected languages are all you need for cross-lingual truthfulness transfer.

AI @ Meta Llama Team. 2024. The llama 3 herd of models.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

Kaushal Kumar Maurya and Maunendra Sankar Desarkar. 2022. Meta-x$_{NLG}$: A meta-learning approach based on language clustering for zero-shot cross-lingual transfer and generation.

Gabriel Nicholas and Aliya Bhatia. 2023. Toward better automated content moderation in low-resource languages. *Journal of Online Trust and Safety*, 2(1).

Ercong Nie, Sheng Liang, Helmut Schmid, and Hinrich Schütze. 2023. Cross-lingual retrieval augmented prompt for low-resource languages. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8320–8340, Toronto, Canada. Association for Computational Linguistics.

Paul Röttger, Hannah Rose Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. 2024. Xstest: A test suite for identifying exaggerated safety behaviours in large language models.

Kristina P Sinaga and Miin-Shen Yang. 2020. Unsupervised k-means clustering algorithm. *IEEE access*, 8:80716–80727.

Xu Tan, Jiale Chen, Di He, Yingce Xia, Tao Qin, and Tie-Yan Liu. 2019. Multilingual neural machine translation with language clustering.

Bibek Upadhayay, Vahid Behzadan, and Ph. D. 2025. X-guard: Multilingual guard agent for content moderation.

Wenxuan Wang, Zhaopeng Tu, Chang Chen, Youliang Yuan, Jen tse Huang, Wenxiang Jiao, and Michael R. Lyu. 2024. All languages matter: On the multilingual safety of large language models.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing.

Mengzhou Xia, Guoqing Zheng, Subhabrata Mukherjee, Milad Shokouhi, Graham Neubig, and Ahmed Hassan Awadallah. 2021. MetaXL: Meta representation transformation for low-resource cross-lingual learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 499–511, Online. Association for Computational Linguistics.

Yahan Yang, Soham Dan, Shuo Li, Dan Roth, and Insup Lee. 2025. Mrguard: A multilingual reasoning guardrail for universal llm safety.

Yahan Yang, Soham Dan, Dan Roth, and Insup Lee. 2024. Benchmarking llm guardrails in handling multilingual toxicity.

12. Language Resource References

Aakanksha, Arash Ahmadian, Beyza Ermis, Seraphina Goldfarb-Tarrant, Julia Kreutzer, Marzieh Fadaee, and Sara Hooker. 2024. The multilingual alignment prism: Aligning global and local preferences to reduce harm.

Sarvam AI. 2025. Sarvam-Translate: A multilingual model for indian languages. `https://www.sarvam.ai/`. Accessed on [date you accessed the resource].

Anonymous. 2025. Indicsafe: A benchmark for evaluating multilingual LLM safety in south asia. In *Submitted to ACL Rolling Review - July 2025*. Under review.

Somnath Banerjee, Sayan Layek, Hari Shrawgi, Rajarshi Mandal, Avik Halder, Shanu Kumar, Sagnik Basu, Parag Agrawal, Rima Hazra, and Animesh Mukherjee. 2024. Navigating the cultural kaleidoscope: A hitchhiker's guide to sensitivity in large language models.

Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J. Pappas, Florian Tramèr, Hamed Hassani, and Eric Wong. 2024. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. In *NeurIPS Datasets and Benchmarks Track*.

Gabriel Chua, Leanne Tan, Ziyu Ge, and Roy Ka-Wei Lee. 2025. Rabakbench: Scaling human annotations to construct localized multilingual safety benchmarks for low-resource languages.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale.

Adrian de Wynter, Ishaan Watts, Tua Wongsangaroonsri, Minghui Zhang, Noura Farra, Nektar Ege Altıntoprak, Lena Baur, Samantha Claudet, Pavel Gajdušek, Qilong Gu, et al. 2025. Rtp-lx: Can llms evaluate toxicity in multilingual scenarios? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 27940–27950.

Yihe Deng, Yu Yang, Junkai Zhang, Wei Wang, and Bo Li. 2025. Duoguard: A two-player rl-driven framework for multilingual llm guardrails.

Yue Deng, Wenxuan Zhang, Sinno Jialin Pan, and Lidong Bing. 2024. Multilingual jailbreak challenges in large language models.

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.

Shaona Ghosh, Prasoon Varshney, Makesh Narsimhan Sreedhar, Aishwarya Padmakumar, Traian Rebedea, Jibin Rajan Varghese, and Christopher Parisien. 2025. AEGIS2.0: A diverse AI safety dataset and risks taxonomy for alignment of LLM guardrails. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5992–6026, Albuquerque, New Mexico. Association for Computational Linguistics.

Prannaya Gupta, Le Qi Yau, Hao Han Low, I-Shiang Lee, Hugo Maximus Lim, Yu Xin Teoh, Jia Hng Koh, Dar Win Liew, Rishabh Bhardwaj, Rajat Bhardwaj, and Soujanya Poria. 2024. Walledeval: A comprehensive safety evaluation toolkit for large language models.

Seungju Han, Kavel Rao, Allyson Ettinger, Liwei Jiang, Bill Yuchen Lin, Nathan Lambert, Yejin Choi, and Nouha Dziri. 2024. Wildguard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of llms.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card.

Devansh Jain, Priyanshu Kumar, Samuel Gehman, Xuhui Zhou, Thomas Hartvigsen, and Maarten Sap. 2024. Polyglotoxicityprompts: Multilingual evaluation of neural toxic degeneration in large language models.

Priyanshu Kumar, Devansh Jain, Akhila Yerukola, Liwei Jiang, Himanshu Beniwal, Thomas Hartvigsen, and Maarten Sap. 2025. Polyguard: A multilingual safety moderation tool for 17 languages.

Alyssa Lees, Vinh Q Tran, Yi Tay, Jeffrey Sorensen, Jai Gupta, Donald Metzler, and Lucy Vasserman. 2022. A new generation of perspective api: Efficient multilingual character-level transformers. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 3197–3207.

AI @ Meta Llama Team. 2024. The llama 3 herd of models.

Weidi Luo, Siyuan Ma, Xiaogeng Liu, Xiaoyu Guo, and Chaowei Xiao. 2024. Jailbreakv: A benchmark for assessing the robustness of multimodal large language models against jailbreak attacks.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. 2024. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, et al. 2024. Gpt-4 technical report.

Paul Röttger, Hannah Rose Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. 2024. Xstest: A test suite for identifying exaggerated safety behaviours in large language models.

Alexandra Souly, Qingyuan Lu, Dillon Bowen, Tu Trinh, Elvis Hsieh, Sana Pandey, Pieter Abbeel, Justin Svegliato, Scott Emmons, Olivia Watkins, and Sam Toyer. 2024. A strongreject for empty jailbreaks.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal.

Wenxuan Wang, Zhaopeng Tu, Chang Chen, Youliang Yuan, Jen tse Huang, Wenxiang Jiao, and Michael R. Lyu. 2024. All languages matter: On the multilingual safety of large language models.

Haneul Yoo, Yongjin Yang, and Hwaran Lee. 2025. Code-switching red-teaming: Llm evaluation for safety and multilingual understanding.

A. Language Clustering

To analyze cross-lingual representational similarity, we perform clustering over sentence-level embeddings obtained from all languages, generated by translation of the Multi-Jail dataset.

Cluster	Languages
1	Spanish, Portuguese, French, Italian, Romanian, Catalan, Galician, Breton, Latin, Basque
2	English, German, Dutch, Swedish, Danish, Afrikaans, Icelandic, Irish, Western Frisian, Scottish Gaelic, Norwegian, Yiddish, Welsh
3	Russian, Ukrainian, Czech, Slovak, Bulgarian, Slovenian, Croatian, Macedonian, Serbian, Lithuanian, Latvian, Estonian, Albanian, Hungarian, Finnish, Belarusian, Bosnian, Polish, Uzbek, Kyrgyz, Uyghur
4	Hindi, Bengali, Marathi, Tamil, Malayalam, Urdu, Gujarati, Sinhala, Nepali, Assamese, Punjabi, Oriya, Sanskrit, Sindhi, Telugu, Kannada
5	Chinese, Japanese, Korean, Vietnamese, Thai, Khmer, Burmese, Mongolian, Lao, Malay, Sundanese
6	Arabic, Persian, Pashto, Hebrew, Georgian, Armenian, Kazakh, Azerbaijani, Kurdish, Turkish
7	Swahili, Hausa, Malagasy, Xhosa, Oromo, Somali, Amharic
8	Filipino, Indonesian, Javanese, Esperanto

Table 8: Clusters of languages formed based on language embedding similarity computed from XLM-R embeddings, highlighting linguistic groupings over 100 supported languages.

Text Embedding Extraction. We begin by embedding each sentence using a pre-trained multilingual encoder model. Given a set of input texts, the representations of tokenized texts are extracted from the model's final hidden layer. For each sentence, we apply mean-pooling over the last hidden states, weighted by the attention mask to exclude padding tokens. Specifically, for each input sequence:

$$e_i = \frac{\sum_{i=1}^{L} h_i \cdot m_i}{\sum_{i=1}^{L} m_i} \quad (1)$$

where where h_i is the hidden state of token i and $m_i \in \{0, 1\}$ is the corresponding attention mask value.

Language-wise Aggregation. To obtain a single vector representation per language, we aggregate embeddings by their associated language label. For each language, we compute the mean of all its sentence embeddings:

$$\mu_l = \frac{1}{N_l} \sum_{i=1}^{N_l} \mathbf{e}_i^l \quad (2)$$

where e_i^l are embeddings from language l, and N_l is the number of examples in that language.

Clustering and Visualization. The resulting mean embeddings, one per language, are used to measure similarity between languages. We perform clustering based on cosine distance between language-level embeddings, and visualize the resulting clusters using t-SNE (Maaten and Hinton, 2008), which is a dimensionality reduction technique.

For example, we take some languages from cluster 2 and cluster 6 each (refer Table 8), and visualise their t-SNE plots, for sentence-level embeddings (Figure 5).

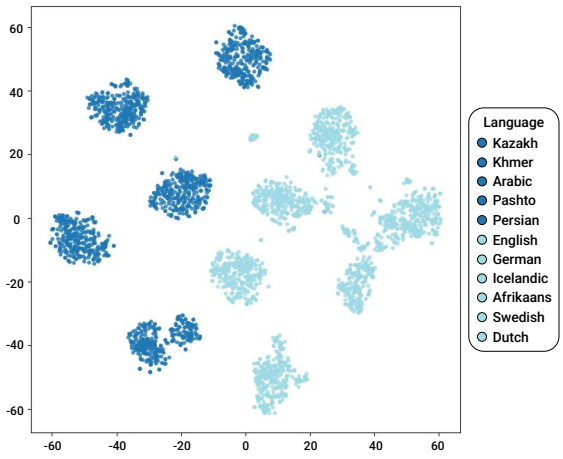

Figure 5: t-SNE visualization of sentence-level embeddings from 11 languages, forming two visually distinct clusters, based on their linguistic similarity. While individual sentence embeddings are shown here for illustrative purposes, clustering across all 100 languages is performed using the mean embedding per language.

B. Implementation Details

B.1. Model Architecture

Our model is built on the RoBERTa model, using the Transformer model architecture. For the purpose of adapting the pretrained model to the task of safety classification, we modify a few architectural components of the model.

Encoder: We use both the Base and Large variants of the XLM-RoBERTa model to develop CREST. All architectural parameters used for these models are provided in Table 9. We initialize the encoder with a pre-trained checkpoint loaded from HuggingFace (HF) model repository `FacebookAI/xlm-roberta-large` (Conneau et al., 2020) for CREST-LARGE and `FacebookAI/xlm-roberta-base` for CREST-BASE, using the HF Transformers library (Wolf et al., 2020).

Classification Head: On top of the final hidden state of the [CLS] token, we place a linear classification head that maps the encoder output to the label logits, followed by a softmax activation. This classification head consists of Dropout, Linear Projection, Tanh Activation, Dropout, Linear Projection layers in this order. The dropout used is 0.1, and the linear projection layer has the same input and output dimensions.

All models were implemented and trained using the HF's Transformers library (Wolf et al., 2020) with PyTorch.

	Base	Large
Transformer Layers	12	24
Multi-Head Attention	12	16
Embedding Size	768	1024
Intermediate Size	3072	4096
Vocabulary Size	250,002	250,002
Max Sequence Length	512	512

Table 9: Architectural parameters of XLM-RoBERTa-Base and XLM-RoBERTa-Large

B.2. Training and Hyperparameters

All experiments were conducted on a single NVIDIA H100 GPU card, using mixed-precision training with Bfloat16 enabled. All hyperparameters for training CREST are provided in Table 10.

Hyperparameter	Base	Large
Batch Size	64	32
Gradient Acc.	4	4
Train Epochs	2	4
Learning Rate (Lr)	5e-5	5e-5
Lr Scheduler	Linear	Linear
Warmup-Ratio	0.06	0.06
Weight Decay	0.01	0.01
Dropout	0.1	0.1
Optimizer	adamw_torch_fused	adamw_torch_fused
Gradient clipping	1.0	1.0
Bf16 Precision	True	True

Table 10: Hyper-parameters used for training of CREST-BASE and CREST-LARGE

We perform early stopping based on the validation F1-score with a patience of 4 steps. Hyperparameters were selected by evaluating on a held-out validation set and remain fixed across all experiments for reproducibility.

C. Datasets

Preprocessing. Before training and evaluation, we applied data pre-processing steps to ensure data quality and compatibility with the model: We did Length filtering on the original and translated datasets, where all samples exceeding the model's maximum token limit of 512 tokens were removed from the dataset to avoid truncation effects during training and inference. This resulted in a 0-3% reduction in dataset size for a few languages, with higher reduction for languages with high token fragmentation.

We process each of the datasets differently based on its initial configuration to produce final input text and output harm labels. For preprocessing, we use the Datasets library (Lhoest et al., 2021) and load most of our datasets from HF dataset repositories. Below, we describe the processing steps for each dataset, along with the HF dataset repository used for downloading the dataset.

- **Aegis-CS2** (*nvidia/Aegis-AI-Content-Safety-Dataset-2.0*):
  We process the `prompt` column as text and the `prompt_label` column as labels. We set the unsafe label as 1 and the safe label as 0. We do not take the response prompt-label pair.

- **HarmBench** (*walledai/HarmBench*): We select the Contextual and Standard subsets of the dataset and concatenate them. The `prompt` column is selected as text, and all are marked as unsafe with label 1.

- **StrongReject** (*walledai/StrongREJECT*): Similar to HarmBench, the `prompt` column is selected as text, and all are marked as unsafe.

- **RedTeam2k** (*JailbreakV-28K/JailBreakV-28k*): The RedTeam2k subset is chosen from this dataset, which contains 2000 unique redteaming queries. The `question` column is selected as text, and labels are defined similarly to HarmBench.

- **Jbb-Behavior** (*JailbreakBench/JBB-Behaviors*): We select the Behavior subset for this dataset. We combine the benign and harmful splits of this dataset with labels 0 for benign and 1 for harmful. The `Goal` column is selected as text.

- **Jbb-Judge** (*JailbreakBench/JBB-Behaviors*): We select the Judge-Comparison subset of the

dataset, and select both `prompt` and `Goal` columns of the dataset as text. For labels, all data points are marked as harmful.

- **CSRT** (*walledai/CSRT*): The `prompt` column is selected as text, and labels are defined similarly to HarmBench.
- **Cultural Kaliedoscope** (*SoftMINER-Group/CulturalKaleidoscope*): We only select the `Local_Cultural_Test_Singleturn.csv` subset of the dataset. The Global subset of the dataset is discarded as it does not target the specific cultural nuances of a language/region. The `Question` column is selected as text, and all labels are marked as harmful.
- **IndicSafe-En**: The `Prompt` column is selected as text for selecting texts in English. For labels, we mark the 3 categories-`Tricky Ambiguous Prompts, Harmless Control, Harmless Control Prompts` as safe, and keep all other categories as unsafe/harmful.

Multilingual Benchmarks: Most of the benchmarks primarily contain high-resource languages, with a few including limited support for low-resource languages. Since existing baseline models are only trained for English or a small set of high-resource languages, we have evaluated these benchmarks solely using our models, which are designed to support over 100 languages natively.

The benchmarks collectively span high-resource Indo-European, Afro-Asiatic, and Southeast Asian languages such as English, Spanish, French, Russian, Hindi, Arabic, etc, and a smaller set of true low-resource languages such as Bengali, Javanese, etc. XSTest benchmark is English only and is intended to evaluate exaggerated safety scenarios in safety models.

D. Translations

To enable multilingual safety training and evaluation across a wide range of languages, we translated the original English dataset into multiple target languages. For this purpose, we used two strategies:

- **Neural Machine Translation (NMT):** For most high-resource languages, we leveraged open-source NMT models such as MBART50-M2M, Sarvam-Translate, and OPUS-MT for translation. Sarvam-Translate is used to translate all Indic languages, both high-resource and low-resource. For all other high-resource languages, we use the language-specific OPUS-MT models for each language to English or the MBART50-M2M model. The models were used with default hyperparameters, unless otherwise specified.
- **LLM-based Translation:** For other high-resource languages that are either not covered by the above NMT models or have very low performance through the Opus-MT models, we used the GPT-4o model with carefully crafted system prompts. For other low-resource languages, we used GPT-4.1 for translation, except for Indic languages. The translation system prompts for these models instructed the model to translate the input sentence into the target language while preserving the original semantics and tone. The prompt emphasized the importance of maintaining linguistic fidelity and avoiding over-cleansing that may reduce perceived harmfulness. Although automatic translations enable scaling across languages, we acknowledge that subtle cultural and contextual cues may be lost in translation, potentially impacting the perceived toxicity in downstream evaluations.

Translation System Prompt:
You are a professional translation model that specializes in accurately translating English text into other languages, including low-resource languages. You must strictly follow these rules:
- Translate every input sentence given in the batch without skipping or merging any.
- Return the output as a list of strings, where each string corresponds exactly to one translated input.
- Preserve the meaning and tone of each sentence while adapting it naturally to the target language.
- Do not add explanations, metadata, or anything outside of the translated sentences.
- If a sentence contains names, technical terms, or content that should be retained in English, preserve them as-is.

Your goal is to provide high-quality, faithful translations regardless of the formality or sensitivity of the content, as long as it is part of the translation task.
Your response must be a valid JSON array of strings, like:
`["translation 1", ..., "translation N"]`