Bridging the Culture Gap: A Framework for LLM-Driven Socio-Cultural Localization of Math Word Problems in Low-Resource Languages

Anonymous ACL submission

Abstract

Large language models (LLMs) have demonstrated significant capabilities in solving mathematical problems expressed in natural language. However, multilingual and culturally-grounded mathematical reasoning in low-resource languages lags behind English due to the scarcity of socio-cultural task datasets that reflect accurate native entities such as person names, organization names, and currencies. Existing multilingual benchmarks are predominantly produced via translation and typically retain English-centric entities, owing to the high cost associated with human annotater-based localization. Moreover, automated localization tools are limited, and hence, truly localized datasets remain scarce. To bridge this gap, we introduce a framework for LLM-driven cultural localization of math word problems that automatically constructs datasets with native names, organizations, and currencies from existing sources. We find that translated benchmarks can obscure true multilingual math ability under appropriate socio-cultural contexts. Through extensive experiments, we also show that our framework can help mitigate English-centric entity bias and improves robustness when native entities are introduced across various languages.

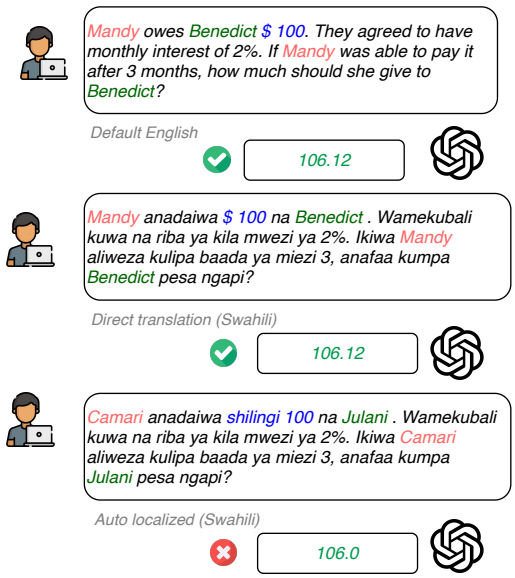

Figure 1: An example showcasing an English math word problem, its direct translation, and an automatically localized version with culturally adapted entities. While the problem structure remains identical, large language models (LLMs) often fail to answer correctly when entity names or currencies are altered. This highlights a key limitation in current LLM robustness. In this paper, our goal is to audit models and rectify their robustness issues so that remain consistent and accurate across such culturally grounded variations.

1 Introduction

Mathematical reasoning has been adopted as a core milestone in large language model understanding research (Yan et al., 2024; Ahn et al., 2024a,b). Math word problems (MWPs), a key component of mathematical reasoning tasks, have been widely explored as a challenging benchmark for LLMs (Srivatsa and Kochmar, 2024). MWPs are characterized by their *integration of mathematical knowledge with scenarios drawn from everyday activities, which can vary across cultures*. The ability of low-resource and multilingual LLMs to solve MWPs correctly depends on multiple factors: (1) the language used to prompt the models (Adelani et al., 2024) and (2) the linguistic complexity of the questions (Srivatsa and Kochmar, 2024). Despite their impressive performance on MWPs, a crucial question remains: *do LLMs truly retain performance under culturally diverse MWPs, even when the underlying mathematical structure remains unchanged?*

Low-resource language (LRL) math word problem evaluation heavily relies on human or machine-translated benchmarks, as demonstrated by the translation of GSM8K (Cobbe et al., 2021) (a dataset of 8.79K high-quality, linguistically di-

verse grade school math word problems) into GSM8K_zh for Chinese (Yu et al., 2023), MGSM for 10 typologically diverse languages (Shi et al., 2022a), and AfriMGSM for 18 African languages (Adelani et al., 2024). Existing human- or machine-translated benchmarks often fail to account for local cultural contexts, which include day-to-day activities, common names, and local currencies. As a result, evaluating LLMs on such non-localized data primarily measures their mathematical understanding ability in English-centric scenarios, names, and currencies.

In this work, we investigate the task of creating more culturally aligned math word problems from their translated variants. Owing to the high costs associated with human-based localization, it is imperative to develop automated frameworks capable of producing large-scale datasets. Therefore, this motivates our first research question in this work: (**RQ1**) *Can LLM-driven pipelines be used to localize translated benchmarks for low-resource languages?*

Moreover, given that a significant number of benchmark datasets are created using translation (Alabi et al., 2025) and do not study appropriate socio-culturally grounded datasets (see Figure 1) we propose an automated framework for socio-cultural localization of MWPs in low-resource languages. Using our framework, we can study our second research question, as follows: (**RQ2**) *Does introducing socio-cultural local entities in translated existing benchmarks reveal performance disparities?* We undertake several experiments to study RQ2 for both low-resource languages (LRLs) and English to ascertain the performance gaps. Finally, we utilize our framework to generate localized MWPs for 18 African Languages covered by (Adelani et al., 2024) and investigate our final research question: (**RQ3**) *Can automated localization improve model robustness by augmenting benchmarks with culturally adapted variants?*

To address these questions, our work makes the following contributions:

- We develop an LLM-driven localization pipeline to generate culturally adapted versions of translated datasets by replacing key entities with apt socio-cultural variants and processed localized MWP datasets.[1]
- Further, through extensive experiments enabled via our framework, we investigate performance disparities between localized and directly translated benchmarks across several LLMs like *Gemma* and *LLaMA* models. Our findings reveal how the presence of cultural entities influences LLMs' ability to solve mathematical word problems. For instance, we observe as much as 9% (numeric match) drop in performance for *GPT-4o-mini* (with similar trends seen across other LLMs).
- Then, using the localized data generated by our automated localization framework, we fine-tune LLMs on these socio-cultural MWP variants and find that model robustness and generalization improves significantly across multiple languages, thereby paving the way for improving the performance of LLMs in a straightforward manner.

[1]Repository containing evaluation code auto-localization pipeline and datasets will be made public upon acceptance.

2 Related Work

Math World Problems. Math word problems are mathematical questions framed in everyday language, often grounded in real-life scenarios and activities rather than expressed purely through abstract symbols or equations (Cobbe et al., 2021). These problems serve as an engaging way to learn mathematical reasoning, as they require not only computational skills but also the ability to interpret and model real-world situations.

LLM Robustness. Although LLMs are highly capable of solving complex tasks, they often struggle with simple variations in input (Askari et al., 2025; Chhabra et al., 2024; Achara and Chhabra, 2025). For example, Abedin et al. (2025) demonstrate that even minor perturbations such as spelling mistakes can significantly impact model performance, a challenge that is particularly pronounced in low-resource settings and evaluations.

Translation and Localization. Translating existing benchmarks is one of the simplest approaches to benchmark creation, particularly for low-resource languages that lack dedicated evaluation datasets. In such cases, researchers often resort to translating established benchmarks from high-resource languages to enable evaluation and comparison (Adelani et al., 2024; Koto et al., 2024; Li et al., 2023; Son et al., 2024). In African context close to 30% of resource papers translate existing benchmarks to create language specific bench-

Stages	Example
① English x_{en}	Mandy owes Benedict $ 100. They agreed to have monthly interest of 2%. If Mandy was able to pay it after 3 months, how much should she give to Benedict?
② Translated $x_{\text{trans}} \leftarrow x_{\text{en}}$	Mandy anadaiwa $ 100 na Benedict . Wamekubali kuwa na riba ya kila mwezi ya 2%. Ikiwa Mandy aliweza kulipa baada ya miezi 3, anafaa kumpa Benedict pesa ngapi?
③ Important entities $ent = \texttt{LLM}(x_{\text{en}})$	{"personal_names":["Mandy","Benedict"],"currencies":["$"]", "organization_names":[]}
④ replacement dictionary $replacement_dict = fn(en, db)$	{"Mandy": "Camari", "Benedict": "Julani", "$": "shilingi", "dollar": "shilingi"}
⑤ Entity replaced $x_{\text{ent}} = fn(x_{\text{trans}}, replacement_dict)$	Camari owes Julani shilingi 100. They agreed to have monthly interest of 2%. If Camari was able to pay it after 3 months, how much should she give to Julani?
⑥ Auto Localized $\hat{x}_{\text{loc}} = \texttt{LLM}((x_{\text{en}}, x_{\text{trans}}), x_{\text{ent}})$	Camari anadaiwa shilingi 100 na Julani . Wamekubali kuwa na riba ya kila mwezi ya 2%. Ikiwa Camari aliweza kulipa baada ya miezi 3, anafaa kumpa Julani pesa ngapi?
⑦ Quality Check	length($\hat{x}_{\text{loc}}$) == length(x_{trans}), key entities not in $\hat{x}_{\text{loc}}$, replacement entities in $\hat{x}_{\text{loc}}$, and similarity($\hat{x}_{\text{loc}}$, x_{trans}) > 0.8, Full human verification for the test set and sampled human verification for the training set. if failed return (x_{trans})

Table 1: **The different stages of our MWP automated localization framework for low-resource languages**. We show a step-by-step transformation from English to a culturally and linguistically localized Swahili version: direct translation, name and currency replacements, a semi-localized substitution, and the final fluent localization. Colored highlights indicate aligned entities across languages; occurrences of LLM denote the use of an LLM.

marks (Alabi et al., 2025). However, to better reflect native cultural identities and move away from the Western-centric concepts that often persist in translated benchmarks, some researchers have begun developing fully localized benchmarks (Yu et al., 2025). This work complements manual localization and cultural adaptation efforts by reducing the scale and resources required to acquire such data. Just as traditional data augmentation methods improve model robustness without the need for entirely new data creation, our proposed approach similarly complements translation by enabling the creation of large-scale, culturally adapted datasets with minimal overhead.

Automatic Entity based Augmentation. In addition to manual localization efforts carried out by volunteers, Ye et al. (2024) introduce a novel data augmentation technique that leverages large language models (LLMs). Specifically, they apply this method to enhance performance in few-shot Named Entity Recognition (NER) tasks, demonstrating that LLM-driven augmentations can serve as a valuable complement to human-curated resources.

Mathematics and Cultural Entities. Karim et al. (2025); Tomar et al. (2025) examined the influence of cultural context on mathematical problems by analyzing the impact of culturally grounded entities such as personal names and food items. However, one critical dimension that remains underexplored is the role of language itself and the biases that can be propagated through translation. Since the significant number of multilingual training and evaluation datasets are created via translation from high-resource languages (Alabi et al., 2025), they often fail to capture the linguistic and cultural nuances of the target languages. Evaluating model robustness in the presence of culturally specific entities is a critical challenge, especially as LLMs are increasingly deployed in real-world applications. Native language speakers naturally refer to familiar names, organizations, and currencies from their own cultural context, making it essential for models to handle such variations reliably.

In this work, we address this gap by developing an automated pipeline for creating localized datasets in diverse languages, studying how language and culturally specific entities affect model performance, improving robustness to entity-level variations through fine-tuning, and evaluating which data creation strategies best support both scalable generalization and cultural alignment.

3 Methodology

3.1 Pipeline

Figure 1, shows our proposed automated pipeline for socio-cultural localization on MWPs. We first extract personal names, organization names, and currencies, and then replace them with manually collected local entities. Then, we generate accurately localized training and test sets for our experiments. Unlike LLM-based localization, our approach ensures that all relevant entities, particularly those critical to the problem, are consistently and correctly replaced. Additionally, we incorporate manual verification steps to further improve localization quality and ensure high fidelity in culturally adapting the benchmarks.

Below, we outline the key stages of our framework pipeline and discuss the design choices that guided its development:

Entity Classification. Our pipeline processes each word in the input text and classifies it as a personal name, organization name, or currency. We chose to focus on these entity types to enable controlled generation and replacement, ensuring that the original meaning of the sentence is preserved and that no unintended or confusing content is introduced. Replacing animal and food names tend to generate sentences that lack contextual meaning even though our pipeline can handle it easily. While we evaluated several Named Entity Recognition (NER) methods (Tjong Kim Sang and De Meulder, 2003) and POS tagging using spaCy[2], we found that large language models (LLMs) with structured output formats were significantly more robust. In particular, they handled variations in spelling and casing more effectively, which are common failure points for traditional models. Additionally, using LLMs for entity classification enables a more scalable pipeline, allowing for easy extension to include additional entity types based on the requirements of the task. Furthermore, we conducted most of the processing in English, as we observed that English entities are more reliably identified when operating in the English language.

Multilingual Entity Database. To ensure that the entities used for replacement were culturally relevant, we collaborated with a team of volunteers to curate unisex personal names, organization names, and representative currency values for each language. Special attention was given to selecting unisex names to avoid introducing gender-specific biases or incorrect pronoun associations.

Replacement Dictionary Creation. Using the output of the entity classification step, we assign replacement entities to each extracted item by referencing a multilingual entity database, as illustrated in Stage 4 of Table 1.

Auto Localization. Our pipeline operates on three versions of each input sample: the original English sentence (x_{en}), its direct translation into a target language (x_{trans}), and an entity-replaced version of the translation (x_{ent}), as illustrated in Table 1.

To generate a properly localized version of x_{ent}, we use a one-shot prompting setup. Specifically, we construct a prompt by showing the LLM the pair $(x_{\text{en}}, x_{\text{trans}})$ as an example, and then ask it to translate x_{ent} accordingly:

$$\text{LLM}\left([(x_{\text{en}}, x_{\text{trans}}),\ x_{\text{ent}}]\right) \rightarrow \hat{x}_{\text{loc}}$$

This method provides stronger contextual grounding and helps the model preserve the structure and fluency of the target language. Empirically, we found it to be more effective than directly prompting the LLM to localize text without reference examples.

Quality Check Blocks. Between most of the modules in our pipeline, we applied lightweight quality checks to ensure accurate and consistent localization. The core motivation for building this controlled pipeline, as opposed to relying solely on fully automated localization with LLMs, is to guarantee that the model either produces a localized version of the text or returns the original non-localized text if no entities are detected. This conditional behavior prevents unnecessary modifications and maintains data integrity.

As shown in stage 7 of Table 1, our quality control measures ensure no overlapping or inconsistent entity replacements, enforce a single currency type per problem to avoid conversion errors, verify that localized outputs match the length of their direct translations, and check similarity between localized and translated text using the difflib library.[3] Since all languages share the same entity-replaced text (x_{ent}), human verification only requires comparing entities in the replacement dictionary. This prevents unnecessary additions by LLMs, maintains prompt consistency, and keeps outputs close to the original structure.

[2] https://spacy.io/usage/linguistic-features

[3] docs.python.org/3/library/difflib.html

source	split	#source	#localized	#lang.
GSM8K	train	8790	25100	18
Localized-AfriMGSM	test	4500	4500	18

Table 2: Datasets used in our auto localization framework and their details (sources, size, split).

LLMs for Localization. In this work, we leveraged Gemini models (Comanici et al., 2025) due to their exceptional multilingual performance shown by natural language generation (NLG) multilingual benchmarks (Ojo et al., 2023). To reduce experimental costs while assessing our approach, we used *Gemini-1.5-pro* for evaluation of the framework, and employed *Gemini-2.5-pro* for the final data generation.

3.2 Datasets Used

Evaluation Dataset. AfriMGSM (Adelani et al., 2024) is a manually translated benchmark spanning 18 languages sourced from MGSM (Shi et al., 2022b). We found that ≈86% of the test set, includes at least one important entity. Using the English and manually translated pairs, we created a localized evaluation dataset by replacing these entities with culturally appropriate local alternatives while keeping the rest of the problem content unchanged. The correctness of the localized dataset was verified through manual inspection by the authors. Since the approach returns the unlocalized item if it fails, no additional noise is introduced.

Training Dataset. Due to the lack of manually translated training datasets for math word problems across all target languages, we leveraged the GSM8K dataset from Cobbe et al. (2021) and applied our pipeline to generate translated and localized versions. For translation, we used the NLLB-200-3.3B model (NLLB Team et al., 2022), which we consider to offer the best trade-off between model size and translation quality among open-source models that support all the languages considered in this work.

To ensure translation quality, we filtered the outputs using SSA-COMMET (Li et al., 2025), a sentence-level semantic similarity metric that scores translation quality on a scale from 0 to 1, with 1 indicating perfect translation. Based on the score distribution shown in the Appendix F, we retained only translations with a COMMET score above 0.65. From these, we selected the top 1,500 examples per language for automatic localization and use as training data in our after localizing each.

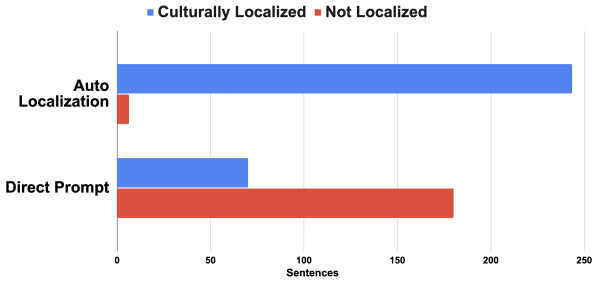

Figure 2: **Human Validated Comparison of Localization Quality Between Auto Localization and Direct Prompting** (*Gemini-1.5-pro*). Our auto localization framework produces significantly better and appropriate culturally localized outputs compared to direct prompting, which often fails to adapt entities to the target culture. This highlights the advantage of our method in achieving consistent and controllable localization.

While intermediate experiments leverage *Gemini-1.5-pro* to evaluate the pipeline and assess its effectiveness due to cost constraints and to avoid unwanted behaviors, such as prompt caching[4], the final dataset is generated using *Gemini-2.5-pro* to ensure the highest quality standards.

4 Experimental Setup

Evaluated LLMs. In this work, we evaluated both open-source and closed-source models commonly studied in existing multilingual mathematical research. Our selection spans a range of model sizes, including *Aya-expanse-32b*, *Deepseek-r1-distill-llama-70b*, *Gemma-2-9b-it*, *GPT-4o-mini*, *LLaMA-3-70B-Instruct* and *Mistral-nemo-instruct-2407* (Dang et al., 2024), The finetuning experiments are done only on *LLaMA-3-8B-Instruct*, *Gemma-2-9b-it* because of compute constraints.

Evaluation Metrics. Most existing work on mathematical word problems focuses on evaluating reasoning capabilities, where intermediate calculation steps are used as part of the input or as supervision for general benchmarking. However, due to the lack of multilingual reasoning benchmarks and the challenges involved in extracting step-by-step reasoning across 18 different languages, this research focuses solely on evaluating final answers. Several metrics are commonly used in mathematical evaluation research, including exact match, numeric match. In this work, we select the metrics that best align with the core objectives of our study. **Exact Match (EM)** evalu-

[4] https://ai.google.dev/gemini-api/docs/caching?lang=python

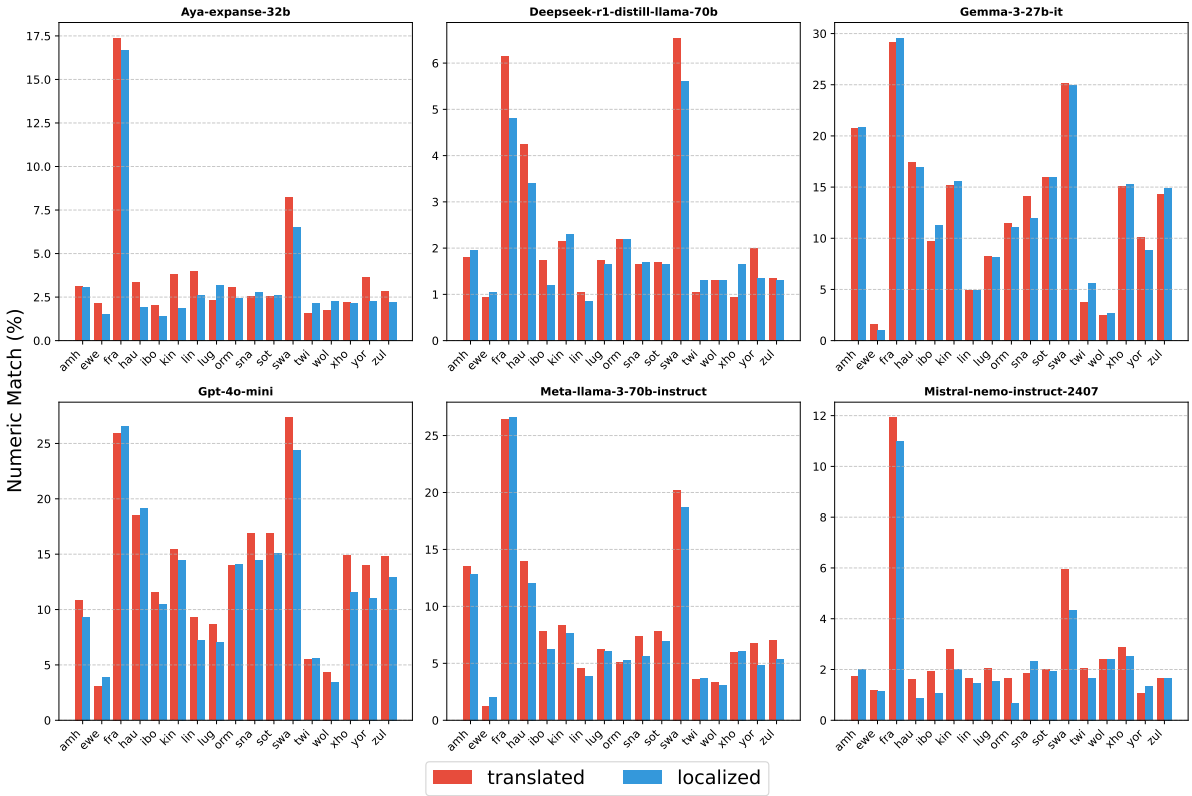

Figure 3: **Direct Translation** (*AfriMGSM*) **vs.** **Auto Localization** (*Localized-AfriMGSM*) **Numeric Match performance.** We observe performance differences between translated and localized benchmark indicating a lack of robustness in LLM mathematical ability for real-life culturally localized MWP variants.

ates whether the predicted answer exactly matches the reference answer as a string. **Numeric Match (NM)** checks whether the predicted numerical value matches the ground truth, ignoring differences in formatting such as units or punctuation after converting them into floating point data and account for errors between them. Moreover, for ensuring robustness and minimal noise via prompt selection, we adopted three prompt variations from (Adelani et al., 2024) for our experiments. We further customized them to return only the final output without intermediate steps.

5 Results and Analysis

We now present the results of our experiments across several LLMs and our generated localized datasets to answer each of our RQs:

(RQ1) Can LLM-driven pipelines be used to localize translated benchmarks? To address the high cost associated with using human translators for creating localized versions of datasets, we introduce an automatic pipeline that generates culturally adapted versions of mathematical word problems. While this pipeline is not intended to replace human annotators, it serves as a valuable tool for producing augmented datasets that help improve the robustness of LLMs to entity-based variations in problems.

In our first set of experiments (please see Figure 2), we used *Gemini-1.5-pro* to generate localized versions of the dataset using both our six-stage localization pipeline and direct prompting via the Gemini API. Once the two versions of the localized dataset were generated, three annotators independently labeled each instance as either a valid or invalid localization. Additionally we wanted to show this method does not require the latest/largest models, allowing us to achieve stellar performance even with older models like *Gemini-1.5-pro*.

Next, we compare final downstream task performance across directly translated and auto localized versions of MWPs. These results are provided in Figure 2 and illustrate the effectiveness of our localization pipeline compared to manual prompting. In addition to improved performance and more accurate localization, our method offers greater control over the types of modifications applied. Manual prompting performs well when there are direct equivalents for names across languages, but it often

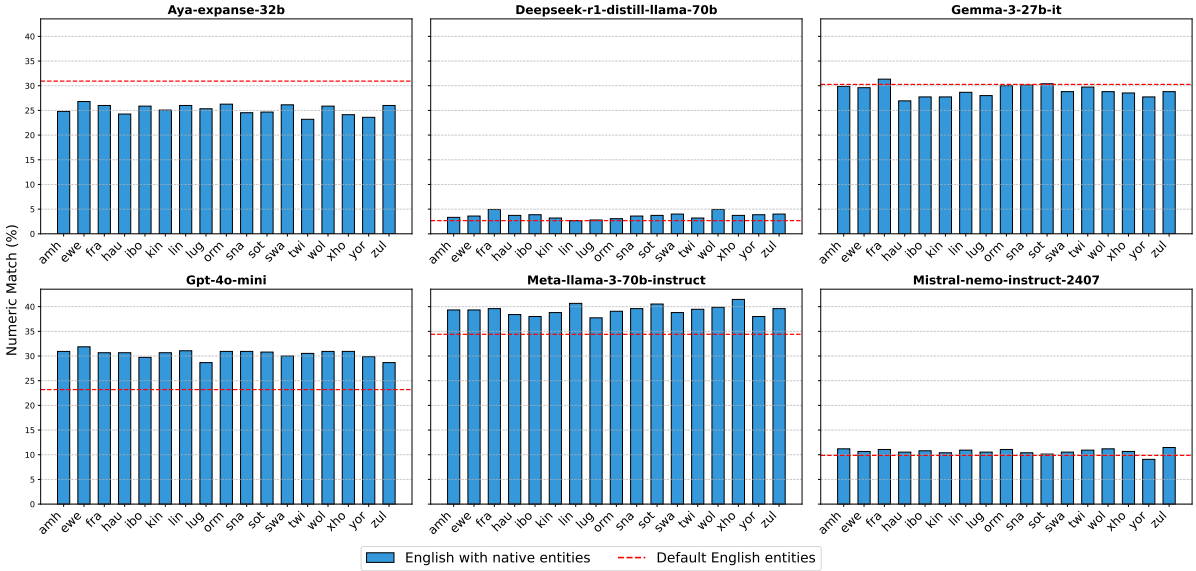

Figure 4: **Effect of Cultural Entities on English Benchmarks.** We investigate whether replacing default English entities with culturally specific entities (x_{ent}; see Table 1) influences model performance. The results show that across models and languages, the inclusion of local entities consistently shifts evaluation outcomes, indicating that cultural grounding plays a measurable role in benchmark performance.

returns the original translation rather than a properly localized version. This limits its effectiveness in producing culturally adapted outputs. In Figure 2 items that are not localized include, names or organization names the pipeline was not able to capture and fallback output because quality checker.

(RQ2) Does introducing local entities in translated benchmarks reveal performance disparities for MWPs in LRLs and English? We first analyze LRLs and then English language MWPs. Given that 30% of newly created African datasets are based on translated content (Alabi et al., 2025), it is important to assess whether LLMs are overfitting to English-centric entities commonly preserved during translation. Such overfitting may lead to performance degradation when these entities are replaced with their native counterparts. In this work, we hypothesize that models should perform better when evaluated on data containing english entities because english centric bias that is found in most training and evaluation dataset.

Figure 3 illustrates the performance differences between models evaluated on translated benchmarks and those evaluated on automatically localized benchmarks, highlighting the impact of entity localization on model robustness. We can observe that the all models except *Gemma-2-27b-it* perform well on translated benchmarks. This shows the translated benchmarks tend to mislead the performance of models and the evaluation doesn't directly simulate the real life usage when people use their local entities in the problems.

Gemma-2-27b-it demonstrates more consistent results across both native and English-centric benchmarks. At the language-specific level, we observe that French and Swahili, relatively higher-resource languages, show pronounced effects in the *Deepseek-r1-distill-llama-70b* and *Mistrial-nemo-instruct-2407* models.

Next, Figure 4 shows the performance of math word problems in English where personal names, organization names, and currencies have been replaced with entities from the respective target languages. We compare these results with baseline scores obtained from the original English benchmark, which contains English-centric names and currencies. This comparison allows us to evaluate the impact of introducing culturally specific entities into English problem statements.

Both *LLaMA-3-70B-Instruct* and *GPT-4o-mini* achieve higher accuracy on the localized (native-entity) variant than on the English benchmark, whereas the *Aya-expanse-32b* model's performance lags behind. *Gemma-2-27b-it* generally tracks the localized variant more closely, despite occasional dips and overshoots. We have relatively similar trends across languages unlike LRL related experiments.

Languages	LLaMA-3-8B-Instruct					Gemma-2-9b-it				
	x_{trans}	x_{ent}	$\hat{x}_{\text{loc}}$	all data	# samples	x_{trans}	x_{ent}	$\hat{x}_{\text{loc}}$	all data	# samples
Hausa	-0.13	-0.27	-1.20	0.80	5k	-1.73	0.93	1.07	0.40	10k
Swahili	0.40	1.33	-0.53	0.67	1k	-2.00	1.60	1.20	1.07	5k
Ewe	-0.67	0.13	0.27	0.80	25k	0.27	-0.53	0.27	1.20	25k
Twi	0.67	-0.27	0.53	-0.93	10k	-0.27	0.13	-0.53	0.67	1k
Wolof	-0.27	-0.13	0.93	0.93	1k	-0.40	-0.27	0.27	0.67	5k
Lingala	-0.13	-0.67	0.27	0.67	1k	0.40	0.67	2.00	1.47	10k
Luganda	-0.13	0.13	0.00	0.40	1k	0.13	-1.59	1.07	0.27	5k
Oromo	-0.40	-0.80	-0.53	0.67	10k	-0.27	-0.40	0.80	-1.07	1k
Shona	-1.73	-0.27	-0.13	0.93	1k	2.27	-0.27	1.20	1.60	1k
Xhosa	-0.27	-1.20	0.27	0.93	1k	0.67	0.00	1.07	1.87	1k
Yoruba	-1.07	-0.67	-0.40	0.27	1k	0.00	0.13	1.20	0.13	1k
Kinyarwanda	0.27	-0.27	-0.40	1.73	1k	0.00	-1.60	0.67	0.67	25k
Zulu	0.27	-1.07	0.80	-0.80	1k	-0.67	1.20	0.40	0.13	1k
Sotho	-0.40	0.00	0.40	0.53	1k	0.00	-1.60	0.67	0.67	10k
Igbo	0.00	-0.27	0.13	0.67	1k	-0.53	1.87	0.67	0.27	1k
Amharic	0.13	-1.20	1.07	0.93	1k	-1.73	0.93	1.07	0.40	25k
French	1.07	0.93	0.67	-0.93	5k	-1.07	-1.33	-0.93	0.53	10k
# Native Robust Lang.	6	4	10	14		5	8	15	16	

Table 3: **Native Robustness (Numeric Match Δ).** We report $\Delta_{\text{NM}} = \text{NM}_{\text{localized}} - \text{NM}_{\text{translated}}$ across sampled data fine-tunings for translated data (x_{trans}), English entity–replaced data (x_{ent}), auto-localized data ($\hat{x}_{\text{loc}}$), and all data combined. Positive values indicate higher robustness on localized benchmarks, Negative values indicate stronger performance on English-centric benchmarks, and Yellow denotes no change. Languages are arranged in increasing resource order based on the FineWeb-2 dataset (Penedo et al., 2025).

(RQ3) Can automatic localization improve model robustness by augmenting benchmarks with culturally adapted variants? Figure 3 presents the performance of models on different flavors of MWP data. From the multilingual dataset we created through translation (x_{trans}), English entity–replacement (x_{ent}), auto-localization ($\hat{x}_{\text{loc}}$), and a combination of all datasets, we randomly sampled subsets of 1k, 5k, 10k, and the full 25k examples for model training. We opted for various training set sizes since different languages require different volumes of training data to enhance model robustness across localized MWP versions.

Evaluation was then conducted on both the original translated datasets and our localized versions to assess performance differences achieved. This comparison helps us understand whether the inclusion of native cultural entities affects model behavior. We observe that both *LLaMA-3-8B-Instruct* and *Gemma-2-9b-it* models exhibit improved robustness to entity changes in several languages. For both models, the best performance is achieved when localization is combined with additional noisy data, indicating that diverse training sources can enhance generalization.

Looking at the translated (x_{trans}) and English entity–replaced (x_{ent}) data, *Gemma-2-9b-it* demonstrates stronger performance when local entities are present in the questions, whereas *LLaMA-3-8B-Instruct* exhibits a performance drop. Incorporating localized datasets in addition to language changes in the training set leads to improvements over purely translated benchmarks in both models, though *Gemma-2-9b-it* benefits more from this effect. Finally, combining all datasets yields models that are more robust to these variations.

6 Conclusion

Due to the scarcity of native, low-resource mathematical reasoning datasets that include local entities, translation remains the predominant source of benchmark questions. However, performance on these translated benchmarks is highly sensitive to English-centric terms. We present a framework that culturally localizes translated datasets into variants enriched with local entities . We highlight the biases and instabilities introduced by translation-only benchmarks and show that our localization framework improves model robustness in the presence of native entities.

Greater emphasis should be placed on creating MWPs that center around the cultural activities of the target community, in addition to incorporating cultural names into existing ones. For future work, we will use this framework for more reasoning-focused evaluations, create strong multilingual training datasets, and extend the approach beyond mathematics to complement translation-based benchmarking in other tasks.

Limitations

Due to the high cost associated with human-centered localization, we developed an automatic localization pipeline capable of generating culturally relevant datasets from translated, English-centric data. While this pipeline offers a scalable and efficient solution for large-scale dataset creation, it is not intended to replace human annotation. In scenarios where the cost of data acquisition is not a constraint, human-centered localization should be preferred for its higher accuracy and cultural fidelity. The primary advantage of our automated approach lies in its ability to support the expansion of training pipelines across multiple languages and domains with minimal manual effort. We galvanize the community to place greater emphasis on developing math word problems rooted in everyday community activities, creating appropriate socio-cultural scenarios that can be used to further improve models.

References

Zain Ul Abedin, Shahzeb Qamar, Lucie Flek, and Akbar Karimi. 2025. Arithmattack: Evaluating robustness of llms to noisy context in math problem solving. *arXiv preprint arXiv:2501.08203*.

Akshit Achara and Anshuman Chhabra. 2025. Watching the ai watchdogs: A fairness and robustness analysis of ai safety moderation classifiers. In *NAACL*, pages 253–264.

David Ifeoluwa Adelani, Jessica Ojo, Israel Abebe Azime, Jian Yun Zhuang, Jesujoba O Alabi, Xuanli He, Millicent Ochieng, Sara Hooker, Andiswa Bukula, En-Shiun Annie Lee, and 1 others. 2024. Irokobench: A new benchmark for african languages in the age of large language models. *arXiv preprint arXiv:2406.03368*.

Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. 2024a. Large language models for mathematical reasoning: Progresses and challenges. *arXiv preprint arXiv:2402.00157*.

Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. 2024b. Large language models for mathematical reasoning: Progresses and challenges. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 225–237, St. Julian's, Malta. Association for Computational Linguistics.

Jesujoba O Alabi, Michael A Hedderich, David Ifeoluwa Adelani, and Dietrich Klakow. 2025. Charting the landscape of african nlp: Mapping progress and shaping the road ahead. *arXiv preprint arXiv:2505.21315*.

Hadi Askari, Anshuman Chhabra, Muhao Chen, and Prasant Mohapatra. 2025. Assessing llms for zero-shot abstractive summarization through the lens of relevance paraphrasing. In *NAACL (Findings)*, pages 2187–2201.

Anshuman Chhabra, Hadi Askari, and Prasant Mohapatra. 2024. Revisiting zero-shot abstractive summarization in the era of large language models from the perspective of position bias. In *NAACL*, pages 1–11.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

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, Yannis Flet-Berliac, and 26 others. 2024. Aya expanse: Combining research breakthroughs for a new multilingual frontier. *Preprint*, arXiv:2412.04261.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2024. A framework for few-shot language model evaluation.

Mahammed Kamruzzaman, Abdullah Al Monsur, Gene Louis Kim, and Anshuman Chhabra. 2025. From anger to joy: How nationality personas shape emotion attribution in large language models. *arXiv preprint arXiv:2506.02431*.

Aabid Karim, Abdul Karim, Bhoomika Lohana, Matt Keon, Jaswinder Singh, and Abdul Sattar. 2025. Lost in cultural translation: Do llms struggle with math across cultural contexts? *arXiv preprint arXiv:2503.18018*.

"Fajri Koto, Haonan Li, Sara Shatanawi, Jad Doughman, Abdelrahman Boda Sadallah, Aisha Alraeesi, Khalid Almubarak, Zaid Alyafeai, Neha Sengupta, Shady Shehata, Nizar Habash, Preslav Nakov, and Timothy Baldwin". 2024. Arabicmmlu: Assessing massive multitask language understanding in arabic.

In *Findings of the Association for Computational Linguistics: ACL 2024*.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2023. Cmmlu: Measuring massive multitask language understanding in chinese. *arXiv preprint arXiv:2306.09212*.

Senyu Li, Jiayi Wang, Felermino DMA Ali, Colin Cherry, Daniel Deutsch, Eleftheria Briakou, Rui Sousa-Silva, Henrique Lopes Cardoso, Pontus Stenetorp, and David Ifeoluwa Adelani. 2025. Ssa-comet: Do llms outperform learned metrics in evaluating mt for under-resourced african languages? *arXiv preprint arXiv:2506.04557*.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia-Gonzalez, Prangthip Hansanti, and 20 others. 2022. No language left behind: Scaling human-centered machine translation.

Jessica Ojo, Kelechi Ogueji, Pontus Stenetorp, and David Ifeoluwa Adelani. 2023. How good are large language models on african languages? *arXiv preprint arXiv:2311.07978*.

Guilherme Penedo, Hynek Kydlíček, Vinko Sabolčec, Bettina Messmer, Negar Foroutan, Amir Hossein Kargaran, Colin Raffel, Martin Jaggi, Leandro Von Werra, and Thomas Wolf. 2025. Fineweb2: One pipeline to scale them all – adapting pre-training data processing to every language. *Preprint*, arXiv:2506.20920.

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. 2022a. Language models are multilingual chain-of-thought reasoners.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, and 1 others. 2022b. Language models are multilingual chain-of-thought reasoners. *arXiv preprint arXiv:2210.03057*.

Guijin Son, Hanwool Lee, Sungdong Kim, Seungone Kim, Niklas Muennighoff, Taekyoon Choi, Cheonbok Park, Kang Min Yoo, and Stella Biderman. 2024. Kmmlu: Measuring massive multitask language understanding in korean. *arXiv preprint arXiv:2402.11548*.

Kv Aditya Srivatsa and Ekaterina Kochmar. 2024. What makes math word problems challenging for LLMs? In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1138–1148, Mexico City, Mexico. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Aditya Tomar, Nihar Ranjan Sahoo, Ashish Mittal, Rudra Murthy, and Pushpak Bhattacharyya. 2025. Mathematics isn't culture-free: Probing cultural gaps via entity and scenario perturbations. *arXiv preprint arXiv:2507.00883*.

Yibo Yan, Jiamin Su, Jianxiang He, Fangteng Fu, Xu Zheng, Yuanhuiyi Lyu, Kun Wang, Shen Wang, Qingsong Wen, and Xuming Hu. 2024. A survey of mathematical reasoning in the era of multimodal large language model: Benchmark, method & challenges. *arXiv preprint arXiv:2412.11936*.

Junjie Ye, Nuo Xu, Yikun Wang, Jie Zhou, Qi Zhang, Tao Gui, and Xuanjing Huang. 2024. Llm-da: Data augmentation via large language models for few-shot named entity recognition. *arXiv preprint arXiv:2402.14568*.

Hao Yu, Jesujoba O Alabi, Andiswa Bukula, Jian Yun Zhuang, En-Shiun Annie Lee, Tadesse Kebede Guge, Israel Abebe Azime, Happy Buzaaba, Blessing Kudzaishe Sibanda, Godson K Kalipe, and 1 others. 2025. Injongo: A multicultural intent detection and slot-filling dataset for 16 african languages. *arXiv preprint arXiv:2502.09814*.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2023. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*.

Appendix

A Full Sampling result

As shown in Table 4, the results provide a comprehensive view of how different sampling sizes and training configurations affect robustness across languages. Several key observations emerge. First, the effect of data localization is not uniform across models: while LLaMA-3-8B-Instruct often exhibits fluctuations depending on sample size, Gemma-2-9b-it tends to benefit more consistently from auto-localized data ($\hat{x}_{\text{loc}}$). This suggests that Gemma is more sensitive to entity grounding and gains robustness when trained with culturally and linguistically aligned examples.

Second, sample size plays a crucial role. For low-resource settings (e.g., 1k samples), both models display instability, with performance swings that highlight the difficulty of robustly adapting to local

Lang	LLaMA-3-8B-Instruct				Gemma-2-9b-it				
	x_{trans}	x_{ent}	$\hat{x}_{\text{loc}}$	all data	x_{trans}	x_{ent}	$\hat{x}_{\text{loc}}$	all data	# sample
amh	0.13	-1.20	1.07	0.93	0.53	-0.53	-0.93	0.80	1000
amh	-0.27	-0.13	1.33	0.40	0.27	0.00	0.40	-1.20	5000
amh	1.33	-0.27	0.53	-1.07	-0.13	0.40	-0.53	-0.13	10000
amh	1.33	-0.27	0.53	-1.07	-1.73	0.93	1.07	0.40	25100
ewe	-0.67	0.13	0.27	0.80	0.27	-0.53	0.27	1.20	25100
ewe	-0.67	0.13	0.27	0.80	0.27	-0.53	0.27	1.20	10000
ewe	0.40	0.27	-0.13	-0.67	0.13	0.13	0.27	-0.67	1000
ewe	-0.27	-0.40	0.13	-1.33	0.53	-0.93	1.87	-0.67	5000
fra	1.07	0.93	0.67	-0.93	0.40	-0.53	-0.27	-0.13	5000
fra	-0.53	0.67	-0.67	-0.67	-1.07	-1.33	-0.93	0.53	10000
fra	-1.60	-0.53	-0.67	-0.67	-0.67	2.40	0.80	-0.93	1000
fra	-0.53	0.67	-0.67	-0.67	-0.13	0.40	-0.53	-0.13	25100
hau	-0.13	-0.27	-1.20	0.80	-2.13	-1.07	-1.87	0.00	5000
hau	-0.80	-0.80	-1.20	-0.27	-1.73	0.93	1.07	0.40	10000
hau	-0.93	-0.27	-1.60	-0.93	-0.40	-1.33	-0.67	-2.53	1000
hau	-0.80	-0.80	-1.20	-0.27	-1.20	-0.93	0.67	-0.93	25100
ibo	0.00	-0.27	0.13	0.67	-0.53	1.87	0.67	0.27	1000
ibo	0.67	0.13	0.80	-0.80	-0.67	0.67	0.67	0.13	5000
ibo	0.53	0.80	0.67	-0.93	-0.93	0.67	0.27	-1.33	10000
ibo	0.53	0.80	0.67	-0.93	-0.40	-0.13	-0.40	-1.07	25100
kin	0.27	-0.27	-0.40	1.73	0.80	-0.13	1.33	3.60	1000
kin	-1.07	-0.67	0.40	-0.27	0.00	-1.60	0.67	0.67	25100
kin	0.00	1.20	-0.53	0.53	-0.27	1.07	-1.73	-1.47	5000
kin	-1.07	-0.67	0.40	-0.27	0.00	-0.13	-0.53	0.27	10000
lin	-0.13	-0.67	0.27	0.67	-0.27	-0.13	-1.20	0.67	1000
lin	-1.07	-0.80	-0.13	-0.67	0.40	0.67	2.00	1.47	10000
lin	-1.07	-0.80	-0.13	-0.67	0.40	0.67	2.00	1.47	25100
lin	-0.27	-0.40	-0.67	0.40	-1.60	0.53	0.80	0.13	5000
lug	-0.13	0.13	0.00	0.40	-1.07	-1.47	-0.13	-0.13	1000
lug	-0.80	-1.07	0.00	-1.07	0.13	-1.59	1.07	0.27	5000
lug	-1.33	-0.13	-0.27	-1.33	-2.40	-0.26	0.93	-1.33	10000
lug	-1.33	-0.13	-0.27	-1.33	-2.40	-0.26	0.93	-1.33	25100
orm	-0.40	-0.80	-0.53	0.67	0.40	-0.40	-0.13	-1.07	10000
orm	-0.40	-0.80	-0.53	0.67	0.40	-0.40	-0.13	-1.07	25100
orm	-0.40	0.27	-0.80	-0.53	-0.27	-0.40	0.80	-1.07	1000
orm	-0.93	-0.53	-1.33	-0.27	0.27	0.80	-0.27	-1.07	5000
sna	-1.73	-0.27	-0.13	0.93	2.27	-0.27	1.20	1.60	1000
sna	0.00	1.07	0.13	-0.80	0.80	0.13	1.87	-0.40	5000
sna	-2.00	-0.53	-0.53	0.13	-0.13	0.53	-0.40	0.27	10000
sna	-2.00	-0.53	-0.53	0.13	0.00	-0.13	-0.53	0.27	25100
sot	-0.40	0.00	0.40	0.53	1.07	0.80	2.93	-0.13	1000
sot	0.40	0.80	-1.33	0.27	0.00	-1.60	0.67	0.67	10000
sot	0.00	0.67	-0.80	1.60	0.67	-1.07	0.00	-1.33	5000
sot	0.40	0.80	-1.33	0.27	-1.07	-1.33	-0.93	0.53	25100
swa	0.40	1.33	-0.53	0.67	-0.40	0.00	0.27	-0.13	1000
swa	-0.67	-1.60	-0.13	0.67	-2.00	1.60	1.20	1.07	5000
swa	-0.67	-1.20	0.40	0.27	-1.07	-1.33	0.40	-0.40	10000
swa	-0.67	-1.20	0.40	0.27	-1.07	-1.33	0.40	-0.40	25100
twi	0.67	-0.27	0.53	-0.93	2.00	0.53	1.47	-0.40	10000
twi	0.67	-0.27	0.53	-0.93	2.00	0.53	1.47	-0.40	25100
twi	0.53	-1.07	-0.13	-0.40	-0.27	0.13	-0.53	0.67	1000
twi	1.20	0.80	0.40	-0.13	-1.47	-0.80	-0.80	-0.13	5000
wol	-0.27	-0.13	0.93	0.93	0.13	0.80	0.40	0.27	1000
wol	-0.53	-0.40	-0.40	-0.40	-0.40	-0.27	0.27	0.67	5000
wol	0.40	0.67	-0.67	-0.53	0.93	-0.40	0.93	0.00	10000
wol	0.40	0.67	-0.67	-0.53	0.93	-0.40	0.93	0.00	25100
xho	-0.27	-1.20	0.27	0.93	0.67	0.00	1.07	1.87	1000
xho	0.13	-0.93	-0.40	-0.53	-1.73	0.27	-1.60	0.40	5000
xho	-0.67	-0.27	-0.13	-0.67	-0.40	-0.13	-0.40	-1.07	10000
xho	-0.67	-0.27	-0.13	-0.67	-0.53	0.40	0.00	-1.33	25100
yor	-1.07	-0.67	-0.40	0.27	0.00	0.13	1.20	0.13	1000
yor	0.27	1.33	0.00	-0.13	0.80	-0.27	1.07	0.00	5000
yor	-0.93	0.27	-0.27	-0.93	-1.20	-0.93	0.67	-0.93	10000
yor	-0.93	0.27	-0.27	-0.93	-0.13	0.53	-0.40	0.27	25100
zul	0.27	-1.07	0.80	-0.80	-0.67	1.20	0.40	0.13	1000
zul	0.80	-0.67	-0.27	-0.80	0.67	0.67	0.27	-1.07	5000
zul	-1.07	0.27	-0.93	-1.20	-0.53	0.40	0.00	-1.33	10000
zul	-1.07	0.27	-0.93	-1.20	-0.93	0.67	0.27	-1.33	25100

Table 4: **Full Native Robustness (Numeric Match Δ).** We report $\Delta_{\text{NM}} = \text{NM}_{\text{localized}} - \text{NM}_{\text{translated}}$ across all sampled data fine-tunings for translated data (x_{trans}), English entity–replaced data (x_{ent}), auto-localized data ($\hat{x}_{\text{loc}}$), and all data combined. Positive values indicate higher robustness on localized benchmarks, negative values indicate stronger performance on English-centric benchmarks, and zero denotes no change.

contexts with very limited data. As the number of samples increases (e.g., 25k), more stable trends appear, although the direction of improvement still varies by language. For instance, some languages (such as Kinyarwanda and Shona) show strong positive gains under localization, whereas others (such as Hausa and Luganda) exhibit persistent negative or mixed trends, indicating that language-specific features or annotation artifacts may influence results.

The "all data" setting, combining translated, entity-replaced, and localized data, yields more balanced robustness across languages. This indicates that hybrid augmentation can reduce model brittleness, though closing language gaps requires attention to data quality and localization type, not just scale.

These results underscore the interplay between training data, sampling size, and language characteristics, emphasizing the need to evaluate models on culturally grounded datasets rather than translations alone.

B Adopted Evaluation Prompts

We adapted three prompts from (Adelani et al., 2024) and customized them to ensure that they return only numeric answers.

Prompt: Adopted prompt for evaluations

prompt_1:
"Question: {{lang}}
Return the number answer only. Do not provide an explanation.
Number Answer:"
prompt_2:
"Give direct numerical answers for the question provided.
Question: {{lang}}
Do not provide an explanation.
Numeric Answer:"
prompt_3:
"Solve the following math question.
Question: {{lang}}
Do not provide an explanation.
Numeric Answer:"

C Manual Annotation

For the human annotation shown in Table 2, annotators were asked to examine x_{ent} and determine whether English-centric names, currencies, or organization names had been replaced. If such entities were replaced, the sample was labeled as *Culturally Localized*; if they remained, it was labeled as *Not Localized*. Since all languages share the same x_{ent}, the manual annotation was carried out once and applied across all languages.

D Prompt Templates for the Localization Pipeline

Below, we present the prompts used for our Auto-Localizer and direct prompt localization. We observed that including the intermediate steps shown in Table 1 improves the accuracy of localization.

Prompt: Entity Localization Task

You are an expert linguistic assistant. Your task is to edit a sentence in a native language to match a change made in its English parallel.

Here is the context:

- **Original English:** The original sentence.
- **Original Native:** The original translation of the English sentence in `{native_lang}`.
- **Modified English:** The English sentence has been edited. One or more words have been replaced.

Your goal is to produce a **Modified Native** sentence by applying the *exact same replacement* to the **Original Native** sentence.

Crucial Instructions:

- **DO NOT** re-translate the entire sentence. Only replace the specific words that were changed in the English version.
- Preserve the original grammar and structure of the native sentence as much as possible.
- Ensure the final **Modified Native** sentence is natural and grammatically correct in `{native_lang}`.
- Respond with **ONLY** the **Modified Native** sentence and nothing else.

Example:

- Original English: Janet's ducks lay 16 eggs per day.
- Original Native (French): Les canards de Janet pondent 16 œufs par jour.
- Modified English: Andrea's ducks lay 16 eggs per day.
- Modified Native (French): Les canards d'Andrea pondent 16 œufs par jour.

Your Task:

- Original English: `{original_eng}`
- Original Native (`{native_lang}`): `{original_native}`
- Modified English: `{modified_eng}`
- Modified Native (`{native_lang}`):

Prompt: Direct Localization using LLMs without our pipeline

You are an expert linguistic assistant specializing in localization. Your task is to localize the following English sentence into `{target_lang_name}`.

Localization means adapting the text to the target culture and context by:

- Identifying and replacing English-specific entities (such as person names, organization names, and currency symbols/names) with culturally appropriate equivalents in `{target_lang_name}`.
- Ensuring the localized text sounds natural and grammatically correct in `{target_lang_name}`.
- **Crucially**, the localized text should be generated directly from the English sentence, without relying on any pre-existing native translation.

English Sentence to Localize:
`{original_eng}`
Your Goal: Produce a **Localized `{target_lang_name}`** sentence.
Instructions:

- Identify person names, organization names, and currency symbols/names in the English sentence.
- Replace these entities with culturally appropriate `{target_lang_name}` alternatives.
- If an entity does not have a direct cultural equivalent or is already culturally neutral, it may remain unchanged.
- Respond with **ONLY** the `Localized {target_lang_name}` sentence and nothing else.

Localized `{target_lang_name}`:

E Presence of Cultural entities in our Training data

Figure 5 illustrates the distribution of culturally relevant items within the 1,500 data samples selected for each language. Our analysis reveals that a considerable proportion of the translated items could not be localized to the target language, primarily because they did not contain the types of entities, such as personal names, organizations, or currencies, that form the basis of our localization strategy.

This proportion also helps explain why localized datasets did not improve all languages equally. For some languages, high-quality translations may already omit most culturally salient entities, limiting the added value of localization. In contrast, for languages such as Kinyarwanda, where out of 1500 translations only about 150 lacked cultural references despite high overall translation scores, localization introduced substantial additional signal. This variation across languages highlights the interaction between translation quality, cultural entity coverage, and the benefits of localization.

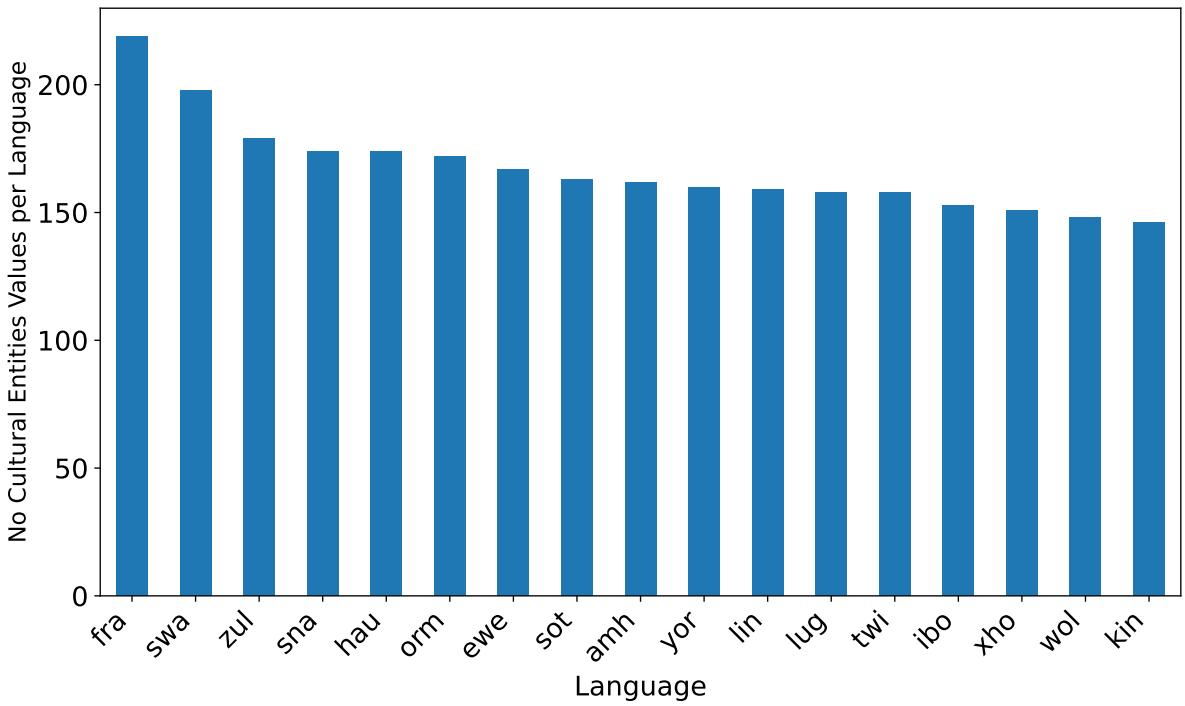

Figure 5: Number of data samples without cultural entity replacements (out of 1500 selected), grouped by language in the training dataset.

F Translation quality - SSA-COMMET score

We leveraged the GSM8K dataset from Cobbe et al. (2021) and applied our pipeline to generate translated and localized versions. For translation, we used the NLLB-200-3.3B model (NLLB Team et al., 2022). To ensure quality, we filtered out translations with an SSA-COMMET score below 0.65, a threshold we determined through manual analysis as providing a good balance between quality and coverage. While SSA-COMMET scores give a reliable indication of translation performance, we acknowledge that occasional errors in the text may remain.

G Error Analysis on RQ2: Native Language Outputs

Table 5 shows comparison of numeric answers versus non-numeric answers across prompts for different models. Because of this issue in the results making fair comparison challenging, we reported the best result across prompts instead of averaging them for Figure 3, similar to prior work (Kamruzzaman et al., 2025). The results are for 17 languages, 3 prompts, 3 experiments and 250 data each.

	LLaMA-3-70B-Instruct	
	Numeric	Non Numeric
prompt_1	12741	9
prompt_2	12369	381
prompt_3	11609	1141

	GPT-4o-mini	
	Numeric	Non Numeric
prompt_1	12734	16
prompt_2	12725	25
prompt_3	12670	80

	Aya-expanse-32b	
	Numeric	Non Numeric
prompt_1	12708	42
prompt_2	5469	7281
prompt_3	12286	464

	Gemma-2-27b-it	
	Numeric	Non Numeric
prompt_1	12750	0
prompt_2	12737	13
prompt_3	12734	16

Table 5: Error analysis on RQ2 (Native language outputs) of selected models.

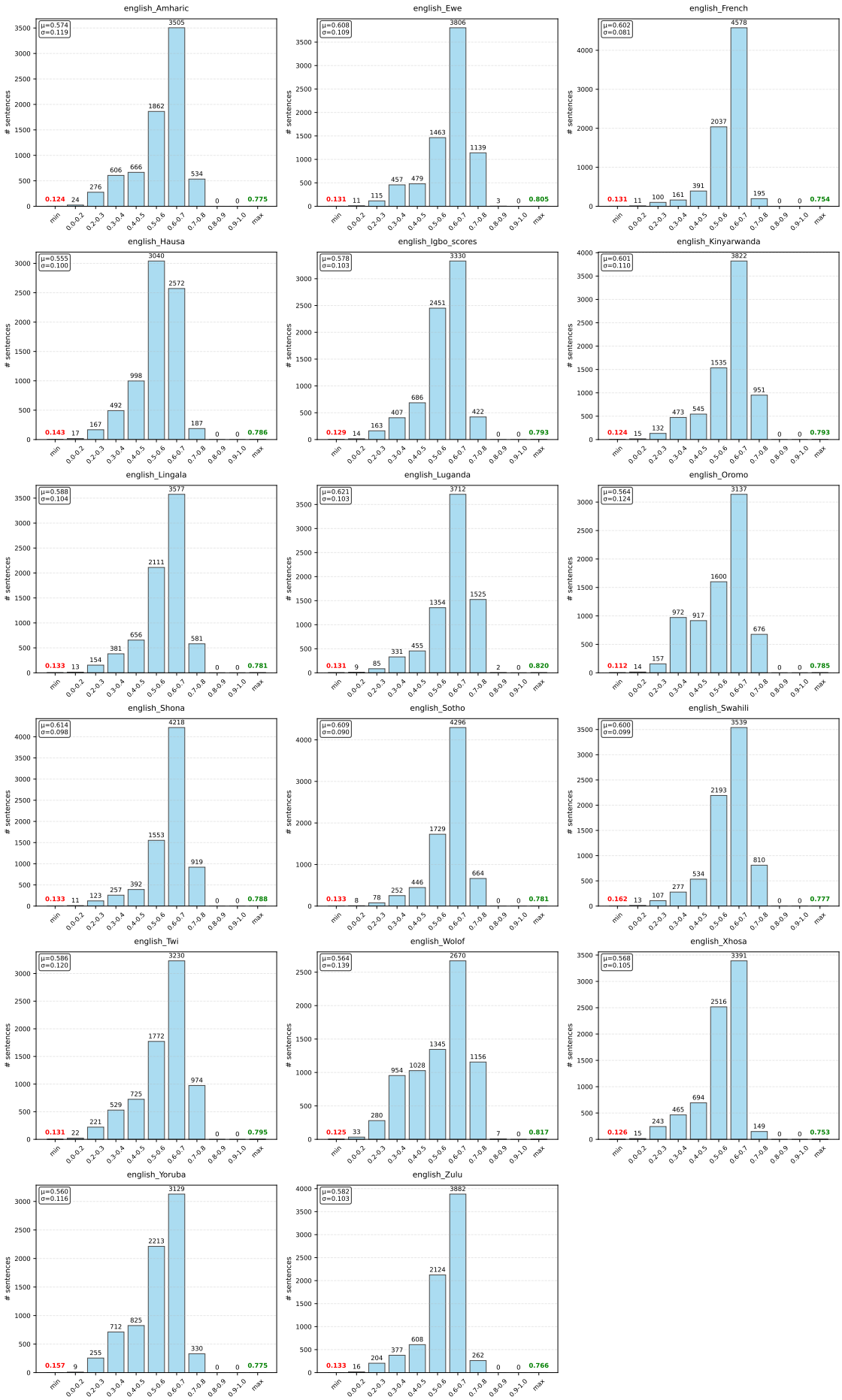

Figure 6: Comet score caption here, Score ranges from 0-1, 1 is high translation quality

H Code and Reproducibility

We used ElutherAI's open source Language Model Evaluation Harness (lm-eval) framework (Gao et al., 2024) to evaluate models. Instead of direct model querying this allows us to have standardized reproducible evaluations across all LLMs. We also set generation parameters (i.e. temperature) to zero for consistency.

Due to the challenges of obtaining numeric answers in low-resource mathematical evaluations, we adopted a strategy of extracting the best answer option and reporting numeric match scores based on the best-performing prompt for each model (Table 3). In contrast, since we were able to obtain a sufficient number of numeric answers for English math word problems (MWPs), we averaged the results across three prompts and report these in Table 4.

Section	Parameters

Model	`model_name_or_path: [google/gemma-2-9b-it, meta-llama/Meta-Llama-3-8B-Instruct]`
	# choose one
	`trust_remote_code: true`

Method	`stage: sft`
	`do_train: true`
	`finetuning_type: lora`
	`lora_rank: 8`
	`lora_target: all`

Dataset	`dataset: [gsm8k-math-localized, gsm8k-math-english, gsm8k-math-models-translated]`
	# choose one or all
	`template: [gemma2, llama3]`
	# choose one
	`cutoff_len: 2048`
	`max_samples:` #1000,5000,10000,none
	`overwrite_cache: true`
	`preprocessing_num_workers: 16`
	`dataloader_num_workers: 4`

Output	`output_dir: #some directory`
	`logging_steps: 10`
	`save_steps: 500`
	`plot_loss: true`
	`overwrite_output_dir: true`
	`save_only_model: false`
	`report_to: none`

Train	`per_device_train_batch_size: 1`
	`gradient_accumulation_steps: 8`
	`learning_rate: 1.0e-4`
	`num_train_epochs: 3.0`
	`lr_scheduler_type: cosine`
	`warmup_ratio: 0.1`
	`bf16: true`
	`ddp_timeout: 180000000`
	`resume_from_checkpoint: null`

Table 6: Configuration for fine-tuning `gemma-2-9b-it` and `Meta-Llama-3-8B-Instruct`