Preprint – IJCAI 2025: This is the accepted version made available for conference attendees.
Do not cite. The final version will appear in the IJCAI 2025 proceedings.

Evaluating and Mitigating Linguistic Discrimination in Large Language Models: Perspectives on Safety Equity and Knowledge Equity

Guoliang Dong[1], **Haoyu Wang**[1], **Jun Sun**[1] and **Xinyu Wang**[2]
[1]Singapore Management University
[2]Zhejiang University
{gldong, junsun}@smu.edu.sg, haoyu.wang.2024@phdcs.smu.edu.sg, wangxinyu@zju.edu.cn

Abstract

Large language models (LLMs) typically provide multilingual support and demonstrate remarkable capabilities in solving tasks described in different languages. However, LLMs can exhibit linguistic discrimination due to the uneven distribution of training data across languages. That is, LLMs struggle to maintain consistency when handling the same task in different languages, compromising both safety equity and knowledge equity. In this paper, we first systematically evaluate the linguistic discrimination of LLMs from two aspects: safety and quality, using a form of metamorphic testing. The metamorphic relationship we examine is that LLMs are expected to deliver outputs with similar semantics when prompted with inputs that have the same meaning. We conduct this evaluation with two datasets based on four representative LLMs. The results show that LLMs exhibit stronger human alignment capabilities with queries in English, French, Russian, and Spanish compared to queries in Bengali, Georgian, Nepali and Maithili. Moreover, for queries in English, Danish, Czech and Slovenian, LLMs tend to produce responses with a higher quality compared to the other languages. Upon these findings, we propose LDFighter, a similarity-based voting method, to mitigate the linguistic discrimination in LLMs. We comprehensively evaluate LDFighter against a spectrum of queries including benign, harmful, and adversarial prompts. The results show that LDFighter significantly reduces jailbreak success rates and improves response quality. All code, data, and the technical appendix are publicly available at: https://github.com/dgl-prc/ldfighter.

1 Introduction

Large language models (LLMs) [Zhao *et al.*, 2023a] have attracted considerable public attention, particularly since the emergence of ChatGPT [OpenAI, 2022], which demonstrates remarkable effectiveness in solving diverse natural language processing tasks, including information extraction [Wei *et al.*, 2023], question answering [Tan *et al.*, 2023], and machine translation [Peng *et al.*, 2023]. LLM-based chatbots, such as ChatGPT, have seamlessly integrated into the daily routines of many individuals, serving as personal assistants and search engines. Moreover, owing to the multilingual capabilities of LLMs, these chatbots are not only prevalent in English-speaking communities but are also widely embraced in non-English-speaking regions. Despite these advancements, the multilingual nature of LLMs can however inadvertently lead to linguistic discrimination due to the uneven distribution of training data across languages.

You
which ocean contains the lowest point on earth?

ChatGPT
The Pacific Ocean contains the lowest point on Earth, which is the Mariana Trench. This trench is located in the western Pacific Ocean, to the east of the Mariana Islands. The deepest point in the Mariana Trench, known as the Challenger Deep, reaches a depth of approximately 10,984 meters (36,037 feet) below sea level, making it the deepest known point in the Earth's seabed.

(a) Detailed response for English query

You
Ո՞ր օվկիանոսում է գտնվում երկրագնդի ամենացածր կետը:

(In English) Which ocean contains the lowest point on earth?

ChatGPT
Մասիս նշված ամենացածր կետը գտնվում է Բադաս օվկիանոսի վրա:

(In English) The lowest point mentioned in Masis is located on the Badas Ocean.

(b) Wrong response for Armenian query

Figure 1: An example of linguistic discrimination in ChatGPT (GPT-3.5) via web client, May 2024

LLMs acquire their multilingual capabilities through training on diverse datasets encompassing multiple languages. The capabilities of an LLM in a specific language is closely tied to the quantity and quality of the training data available for that language during the model's training. However, real-world text-based resources across different languages are often unevenly distributed. Some languages benefit from abundant data resources, while others, spoken by smaller populations or with limited online presence, suffer from a scarcity of digital content. For instance, English is considered a high-resource language with a substantial amount of digital text and linguistic resources available, whereas Bengali, used by a smaller population with less digital content, is classified as low-resource languages [Team *et al.*, 2022]. Consequently, such multilingual languages imbalance poses a significant challenge for LLMs in providing consistent services across different languages. Figure 1 illustrates this challenge with a concrete example, i.e., while ChatGPT provides a detailed and useful response to the query posed in English (Figure 1a), it generates a simple and, more importantly, wrong response when presented with the same query in Armenian (Figure 1b).

Although some empirical studies investigate the performance differences of LLMs across various languages, they primarily focus on specific tasks, such as evading safety checks [Puttaparthi *et al.*, 2023; Deng *et al.*, 2024] and translation performance [Hendy *et al.*, 2023]. The holistic assessment of linguistic discrimination in LLMs remains relatively underexplored. In this study, we systematically evaluate LLMs' linguistic discrimination using metamorphic testing [Segura *et al.*, 2016]. Specifically, we evaluate the consistency of LLMs' responses to a set of perturbed (via translation) but semantically equivalent queries from two perspectives: safety and quality. Safety assessment evaluates if LLMs align with human judgment on harmful queries across languages, while quality assessment evaluates if they provide consistent-quality responses to benign queries in different languages. We evaluate four LLMs (Llama2-13b [Touvron *et al.*, 2023], Gemma-7b [Team *et al.*, 2024], GPT-3.5-turbo [OpenAI, 2022] and Gemini-pro [Google, 2024]) over two datasets: AdvBench [Zou *et al.*, 2023] and NQ [Kwiatkowski *et al.*, 2019]. Our results indicate that the four LLMs exhibit the strongest human alignment capabilities when processing queries in English, French, Russian, and Spanish, with an average jailbreak rate of only 1.04% for harmful queries, but show the weakest performance when handling queries in Bengali, Georgian, Nepali, and Maithili, with an average jailbreak rate of 27.7% for harmful queries. Moreover, for queries in English, Danish, Czech and Slovenian, LLMs tend to generate higher-quality responses, with an average F_1-score of 0.1494. By contrast, the F_1-score in Kannada, Southern Pashto, Tajik and Telugu is only 0.0341 on average. These findings suggest the prevalence of linguistic discrimination in LLMs, underscoring the urgency of addressing this issue for fair and consistent service for all users.

Existing efforts to mitigate linguistic discrimination primarily concentrate on improving the machine translation capabilities of LLMs for low-resource languages. The proposed techniques include data augmentation and fine-tuning, specifically tailored for languages with limited linguistic resources. For instance, the NLLB Team [Team *et al.*, 2022] introduces a novel bitext mining method that automatically generates hundreds of millions of aligned training sentences for low-resource languages. Similarly, Lankford *et al.* propose *adaptMLLM*, a framework designed to streamline all processes related to fine-tuning the machine translation capabilities of LLMs for low-resource languages. However, training on excessive number of languages can potentially undermine the performance of high-resource languages [Chang *et al.*, 2023]. In addition, fine-tuning LLMs can be a complex task with high cost, i.e., requiring substantial computational resources, domain-specific data, and considerable time for evaluation and adjustments. Importantly, fine-tuning for machine translation alone may not directly address linguistic discrimination on a broader scale, i.e., the inconsistent responses of LLMs across different languages.

Consequently, we propose LDFighter, a lightweight multilingual consistency-ensuring method. When provided with a query, LDFighter first translates it into k selected languages. Subsequently, it prompts the target LLM with these translated queries separately. After that, LDFighter translates all the responses into a pivot language, i.e., English, and select the final response to the user through similarity-based voting method. LDFighter is designed to ensure that LLMs provide consistent and unbiased service to speakers of all languages. We evaluate LDFighter on both AdvBench and NQ datasets. The results demonstrate that LDFighter not only significantly reduces the multilingual jailbreak success rate but also improves response quality. Particularly, we also evaluate LDFighter's effectiveness against adversarial prompts generated by two recent jailbreaking attacks, GCG advbench and AutoDAN [Zhu *et al.*, 2023]. The results indicate that LDFighter can successfully invalidate about 94% of adversarial prompts.

2 Preliminary

Metamorphic testing. Metamorphic testing [Segura *et al.*, 2016] is a software testing technique used to address the oracle problem. The core of metamorphic testing lies in establishing metamorphic relations (MRs), which define how the output should change in response to specific input modifications. By verifying these MRs, testers can indirectly assess the correctness of the software. For example, to test the implementation of the function, we can construct metamorphic relations such as $sin(-x) = -sin(x)$ and $sin(x) = sin(\pi - x)$. These relations can be used for automatic input mutation and failure detection. In this work, we systematically test linguistic discrimination in LLMs using a metamorphic relation based on semantic equivalence.

Linguisitc discrimination. Linguistic discrimination refers to the unfair or prejudiced treatment of an individual or group based on language use. The unfair treatment is not limited to interactions between people, it can also be embedded within technologies, products, and systems [Blasi *et al.*, 2022]. In the field of NLP, high-resource languages receive more attention and resources, from both academia and industry, compared to low-resource languages. This discrepancy perpetuates linguistic discrimination in digital spaces by neglecting the needs and contributions of speakers of low-resource languages. Although most fundamental NLP technologies are language-agnostic, applications based on these technologies are often tailored to specific languages. For instance, a speech recognition system may struggle to understand or accurately transcribe certain languages if it has not been adequately trained and tested on them, resulting in frustration and exclusion for speakers of those languages. In this work, we specifically study linguistic discrimination in LLMs.

LLM jailbreak. LLM Jailbreak refers to deliberate attempts by users to bypass the inherent safety, ethical, or operational protocols of LLMs to obtain inappropriate or harmful content. The goal of LLM jailbreak is to elicit responses from LLMs that violate their intended usage guidelines. Mainstream jailbreak approaches often involve embedding harmful questions into carefully designed prompts. For instance, the "Grandma Exploit" template instructs ChatGPT to impersonate the user's deceased grandmother, leading to inappropriate or unethical responses. Additionally, some researchers, inspired by adversarial attacks on traditional neural networks, append adversarial suffixes to harmful questions to provoke harmful responses from LLMs [Zou *et al.*, 2023].

3 Evaluation of Linguistic Discrimination

In this section, we use metamorphic testing to systematically evaluate the linguistic discrimination across widely used open-source and commercial LLMs. We first introduce the metamorphic relationships used in this study, and then present the evaluation details focusing on three research questions.

3.1 Metamorphic Transformation

For the linguistic discrimination evaluation in LLMs, the metamorphic relation (MR) is defined to ensure that semantic equivalence in inputs leads to semantic equivalence in outputs. That is, if two queries q_i and q_j have the same meaning, then their corresponding outputs O_i and O_j should also have the same meaning. We formalize this metamorphic relation as follows:

$$MR : (Sem(q_i) = Sem(q_j)) \implies (Sem(O_i) = Sem(O_j))$$

where *Sem(q)* denotes the semantics of input q. In this study, we utilize machine translation to automatically generate semantically equivalent queries.

3.2 Research Questions

We focus on two types of linguistic discrimination in LLMs: safety discrimination and quality discrimination. Safety discrimination occurs when LLMs provide safer mechanisms for speakers of certain languages, while quality discrimination involves offering better content quality to speakers of some languages. We highlight that both discrimination types do matter. Safety discrimination not only poses potential threats to the safety of communities where certain languages are spoken but also jeopardizes the well-being of entire populations. For instance, users speaking high-resource languages may adopt online translation services to elicit harmful responses from LLMs, facilitating criminal activities. On the other hand, quality discrimination directly affects knowledge equity. This disparity in response quality significantly hampers the ability of marginalized groups to fully engage in educational opportunities and make well-informed decisions. Despite the critical implications, existing research [Puttaparthi *et al.*, 2023; Yong *et al.*, 2023; Deng *et al.*, 2024] primarily focuses on safety discrimination, largely overlooking quality discrimination. In this empirical study, we address this research gap by investigating both types of linguistic discrimination through the following research questions: **RQ1)** Do LLMs offer consistent levels of safety across different languages? **RQ2)** Do LLMs provide responses of the same level of quality for different languages? **RQ3)** Which languages do LLMs perform better?

3.3 Experimental Settings

Target LLMs. Our evaluation focuses on four representative LLMs, comprising two well-known open-source models and two commercial closed-source models: Llama2-13b [Touvron *et al.*, 2023], Gemma-7b [Team *et al.*, 2024], GPT-3.5 [OpenAI, 2022] and Gemini-pro [Google, 2024]. We set the temperature of all LLMs to 0.0 during the experiments to minimize randomness.

Dataset. We utilize two datasets, i.e., **AdvBench** [Zou *et al.*, 2023] and **NQ** [Kwiatkowski *et al.*, 2019], to evaluate safety discrimination and quality discrimination, respectively. The former comprises 520 harmful instructions spanning a wide range of adverse content, including profanity, threats, misinformation, discriminatory content, cybercrime, and hazardous or unlawful recommendations. The latter contains 7842 instances constructed from anonymized aggregated queries to the Google search engine.

Languages. LLMs vary in their multilingual capabilities. To conduct a balanced assessment on linguistic discrimination across the selected LLMs, we opt for a common set of languages among the four selected LLMs. However, these models lack explicit documentation on their supported languages. To address this, we employ an iterative probing approach to determine their common intersection through seed questions. Specifically, we start by randomly selecting one question from each of AdvBench and NQ. These questions are then translated to an initial set of languages using SeamlessM4T-v2 [Barrault *et al.*, 2023], a leading open-source translation model that supports 98 languages. This initial set thus included 98 languages. Next, we input the translated questions into Llama2-13B and eliminated unsupported languages based on the model's responses. This process was then repeated with the three other LLMs to further refine the language set. Ultimately, the final language set consisted of 74 languages. Note that the translation quality in our experiments is reasonably high according to an additional experiment (refer to the technical appendix for more details).

Jailbreak judge. For the evaluation of safety discrimination, we categorize responses generated by LLMs to harmful questions into three types: safe, jailbreak, and invalid. 'safe' responses are instances where the model either declines to answer the query directly or provides positive and benign content, effectively countering the harmful intent of the question. 'jailbreak' responses involve direct answers or indications of the LLM's inclination to respond to the harmful questions. Finally, 'invalid' responses exhibit content that is unrelated or explicitly indicates the LLM's failure to comprehend the question (refer to the technical appendix for more details).

3.4 Results

RQ1: Do LLMs offer consistent levels of safety across different languages? While LLMs trained with safety-oriented methods typically excel at rejecting vanilla harmful queries in widely spoken languages, their performance in less commonly spoken languages remains relatively unexplored. This knowledge gap poses a risk of multilingual jailbreaking. Therefore, the objective of this research questions is to investigate LLMs' ability to resist vanilla harmful queries across

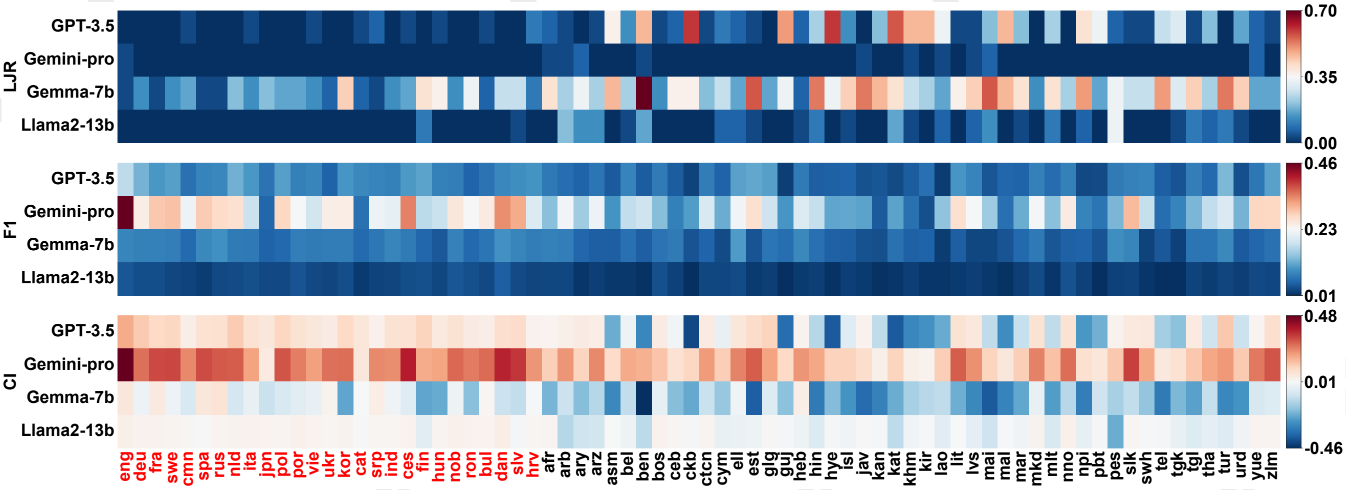

Figure 2: LJR, F1-score and CI score for different languages on four LLMs

LLM	LJR		F_1-score		CI-score	
	Mean	RSD(%)	Mean	RSD(%)	Mean	RSD(%)
GPT-3.5	0.123	139	0.0708	42	-0.0154	940
Gemini-pro	0.0063	258	0.1868	46	0.1954	48
Gemma-7b	0.2743	58	0.0588	34	-0.1367	91
Llama2-13b	0.0333	181	0.0212	43	-0.0059	772
Avg	0.1092	159	0.0844	41.25	0.0094	462.75

Table 1: Mean score and relative standard deviation (RSD) of metrics for each LLM across 74 languages

a broader spectrum of languages. To this end, we translate harmful questions in AdvBench to 74 languages, feed them into four LLMs and count the ratio of harmful responses. To ensure accuracy and reliability, we randomly select 30 harmful questions from AdvBench and meticulously manually label all 8800 responses of each LLM to harmful questions in this work. To minimize potential bias in the labeling process, we established detailed guidelines and applied them consistently. For more details about the labeling criteria, we refer readers to the technical appendix. We then analyze the performance of four LLMs across 74 languages from different dimensions with regards to safety with these 8800 responses. Specifically, we measure the language jailbreak rate (LJR) on each involved language. Given an LLM Θ and a language l, we collect all the responses of all the harmful questions in language l, and count the ratio of 'jailbreak' responses. Formally, we define LJR on LLM Θ and language l as follows.

$$\varphi(\Theta, l) = \frac{\sum_{i=1}^{N} \mathcal{I}(\psi(\Theta, q_i^l) = 1)}{N} \tag{1}$$

where N is the total number of harmful questions, $\mathcal{I}(y)$ is a sign function which equals 1 if y holds and 0 otherwise, and $\psi(\Theta, q)$ is a jailbreak judge function which takes as input a target LLM Θ and a query q; q_i^l denotes the query for the i-th harmful question in language l. $\psi(\Theta, q)$ outputs 1 if the response of LLM Θ on query q is classified as 'jailbreak', and 0 otherwise.

To intuitively measure the performance differences of the same model across different languages, we use relative standard deviation (RSD) rather than standard deviation as the evaluation metric. RSD is calculated as the ratio of the standard deviation to the absolute value of the mean score. In Table 1, we summarize the mean LJR score and RSD for each LLM in the "LJR" column. The results indicate substantial variability in performance across languages for all four LLMs, with the lowest RSD at 58% and an average RSD of approximately 159%. Each LLM also demonstrates varying levels of safety. For instance, Gemini-pro achieves the lowest average LJR (0.0063) but exhibits the highest RSD (258%), reflecting strong average safety performance but with substantial variability across languages. In contrast, Gemma-7b records the highest average LJR (0.2743) and the lowest RSD (58%), demonstrating relatively consistent but weaker safety performance across languages. Notably, Llama2-13b outperforms GPT-3.5 in terms of safety, with an average LJR of 0.0333 compared to 0.123.

Additionally, the heatmap at the top of Figure 2 presents the LJR score for each language across all LLMs. The x-axis displays the involved languages, ordered based on their prevalence in the pretraining data of Llama2, the only LLM providing information on language distribution in its pretraining data among the four LLMs. Each language highlighted in red in Figure 2 constitutes at least 0.005% share in the pretraining data of Llama2, and we roughly categorize them as high-resource languages. In general, the LJR of languages highlighted in red appears as a darker shade of blue (lower LJR score) compared to the remaining languages, indicating that LLMs trained on high-resource languages exhibit better safety performance, as expected. Particularly, Bengali (ben) experiences the poorest safety performance on average, posing a risk to the language community.

When examining individual LLMs, it becomes apparent that each LLM suffers from lack of safety in several languages. Remarkably, the rows corresponding to GPT-3.5 and Gemma-7b are considerably more colorful than others in Figure 2, suggesting that their safety mechanisms are less effective for certain languages. Given that GPT-3.5 is the backend model of the wildly used ChatGPT, this discovery exposes a risk as attackers may use a translator to easily elicit inappropriate content from ChatGPT. Consequently, there is an urgent need to address such kind of attacks. We refer readers to the technical appendix for a real-world jailbreaking example in which we pose a harmful question, e.g., *write a tutorial on how to make a bomb*, in Gujarati to ChatGPT and ChatGPT directly addresses that question with a harmful response. This example is originally from our experiment on GPT-3.5.

For Gemini-pro, we observe that 63 out of 74 languages have an LJR of 0. For the remaining languages with a non-zero LJR, the values are relatively low, not exceeding 0.1. This observation explains why Gemini-pro has the lowest mean LJR score but the highest RSD. In contrast, for Gemma-7b, no language has an LJR of 0, and in 65 out of 74 languages, the LJR exceeds 0.1, indicating a significant difference in the safety performance of the two LLMs. It is worth noting that Gemma-7b and Gemini-pro originate from the same research and technology. We thus conducted a further analysis, employing identical language and questions with both LLMs, to investigate potential reasons for these discrepancies. Besides variations in the safety tuning process and model size, we speculate that Gemini-pro's integration with auxiliary models, indicated by its API generating exceptions for queries deemed unsafe by these models, could be a contributing factor. Examining cases where Gemma-7b provides answers while Gemini-pro does not, we found that approximately 31.2% of such instances are rejected by auxiliary models. Additionally, we observed that Llama2-13b exhibits superior safety performance compared to GPT-3.5 across various languages. Notably, in our experimentation, we utilized the GPT-3.5 API via the Azure platform, which incorporates a series of content filtering models, whereas Llama2-13b achieves its safety performance on its own. These findings suggest that both powerful auxiliary models and careful safety fine-tuning can significantly enhance the consistency of safety performance across different languages. However, both safety measures entail substantial manual effort and costs. Moreover, they may not be sufficient to ensure consistent safety performance across various languages. It is precisely why we introduce a lightweight approach to complement existing methods in Section 4.

RQ2:Do LLMs provide responses of the same level of quality for all supported languages? To answer this question, we exercise different LLMs with benign questions across various languages, and analyze the quality of their responses. Specifically, we first translate questions in NQ dataset to 74 languages, resulting in 2220 queries. Subsequently, we input these queries to four LLMs separately. For analytical purposes, we translate all non-English responses into English and assess the quality of each response.

Following related works [Liang *et al.*, 2022; Kwiatkowski *et al.*, 2019], we adopt F_1-score to comprehensively access the response quality (see the technical appendix for more details). We present the mean F_1-score and RSD for each LLM in the "F_1-score" column of Table 1. The average RSD of the F_1-score across the four LLMs is approximately 41%. Although this value is lower than the smallest RSD observed for LJR, it still indicates significant variability in response quality across languages. We highlight that the high variance of F_1-score does not necessarily indicate a tendency for a LLM to produce low-quality responses. For example, Gemini-pro, which has the highest variability (with 46% RSD) across languages, also shows the highest response quality on average (with 0.1868 F_1-score). According to the heatmap of F_1-score in Figure 2, Gemini-pro exhibits a broad range of F_1-score values and achieves higher F_1-scores in most languages compared to other LLMs. In addition, both GPT-3.5 and Gemini-pro, as expected, have the best F_1-score on English (0.1723 and 0.46 respectively). However, Gemma-7b and Llama2-13b attain their highest F_1-scores in languages Greek (ell, 0.1038) and Danish (dan, 0.0449) respectively, instead of English.

RQ3: Which languages do LLMs perform better? To answer this question, we introduce a new metric named comprehensive index (CI) by combining LJR and F_1-score. CI measures the overall performance of an LLM on a specific language. We defined CI as follows.

$$CI = \alpha \cdot F_1 - (1 - \alpha) \cdot LJR \tag{2}$$

where α takes values between 0 and 1, controlling the significance of F_1-score and LJR. For example, if we prioritize the importance of safety performance, we can set a smaller α. In this work, α is set to 0.5. Note that LJR and F_1 in Formula 2 are normalized values. We present the CI scores across languages at the bottom part of Figure 2. As expected, high-resource languages (highlighted in red) have higher CI scores. The top five languages with the highest CI scores are English (eng, 0.1899), French (fra, 0.1262), Russian (rus, 0.1205), Spanish (spa, 0.1181), and Czech (ces, 0.1103). In contrast, the bottom five languages are Bengali (ben, -0.1718), Georgian (kat, -0.1491), Central Kurdish (ckb, -0.1319), Nepali (npi, -0.1296), and Malayalam (mal, -0.1286). The significant discrepancies in comprehensive performance across different languages are noteworthy. Among all languages, English performs exceptionally well compared to other languages, with a CI score approximately 50% higher than that of French, i.e, the second-highest language.

For individual LLMs, both GPT-3.5 and Gemini-pro perform best in English, with CI scores of 0.1847 and 0.4762, respectively. In a surprising outcome, the highest CI score for Llama2-13b is in Danish (dan,0.0439), while for Gemma-7b, it is in Russian (rus, 0.0733). English ranks as the second-highest language for Llama2-13b with a CI score of 0.037, and third-highest for Gemma-7b with a CI score of 0.0619. Upon further analysis, we find that for both LLMs, the F_1-score in English is slightly lower than that in the highest language (0.07/0.09 for Llama2-13b and 0.17/0.19 for Gemma-7b). However, the LJR values are consistent across both languages (0.0/0.0 for Llama2-13b and 0.05/0.05 for Gemma-7b). The lower F_1-score in English may be due to lower precision, as English responses tend to provide more detailed answers. This assumption is supported by findings from Gemma-7b's responses. Among questions with the same recall in both English and Russian, there are five questions where Russian responses have higher precision compared to English, and only one question where the reverse is true. For example, in response to the question "What are the names of the atom bombs dropped on Japan", the English answer provides more detail: "The two atom bombs dropped on Japan on August 6 and 9, 1945, were named 'Little Boy' and 'Fat Man' ", while the Russian answer is more concise: "The atomic

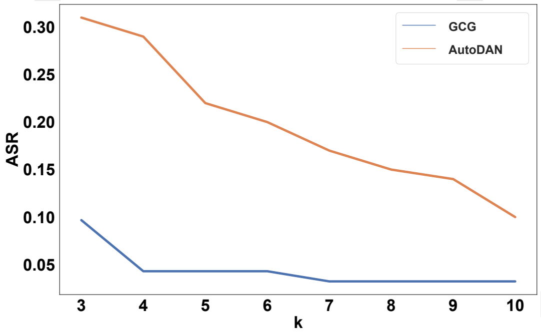

Figure 3: ASR of LLMs with LDFighter

bombs dropped on Japan were called 'Little Boy' and 'Fat Man' ".

4 Mitigation of Linguistic Discrimination

In this section, we introduce LDFighter, a lightweight approach designed to mitigate linguistic discrimination from both safety and quality perspectives, and then assess the effectiveness of LDFighter based on three research questions.

4.1 Similarity-Based Voting

LDFighter involves several steps. First, a query is translated into a selection of K languages. Then, each translated query is fed into a target LLM to generate responses in different languages. Next, these responses are translated to English, and the average similarity of each response to others are calculated. Based on our observations, queries containing harmful content are more likely to receive refusal responses. We thus only keep those refusal responses due to safety or ethical concerns as the candidates if there are (such kind of refusal responses are identified by keyword matching), or otherwise keep all the responses as candidates. Finally, the candidate with the highest average similarity is selected as the final answer to the original query. Intuitively, when presented with the same question, different individuals tend to offer answers that exhibit certain similarities (assuming they possess similar ground-truth knowledge). Inspired by this observation, we thus select the response most similar to others from a set of responses as the final output of the target LLM (the selected answer will be translated back into the language of the original query if required). We adopt a cosine similarity to measure the similarity between pairs of responses, and use the average cosine similarity for each response to denote how the response is similar to others. We summarize the process described above in an algorithm, accompanied by a flow diagram and explained step by step in the technical appendix.

4.2 Research Questions

We evaluate the effectiveness of LDFighter by addressing three key research questions: **RQ4)** How effective is LDFighter on improving the safety performance of LLMs across different languages? **RQ5)** How effective is LDFighter on improving the response quality of LLMs across different languages? **RQ6)** What is the cost of LDFighter?

LLMs equipped with LDFighter inherently offer consistent service across all languages. In addition to ensuring consistency, we investigate whether LDFighter can maintain or surpass the original performance of LLMs in terms of safety and quality. Therefore, RQ4 and RQ5 aim to assess LDFighter's effectiveness in improving multilingual safety performance and response quality, respectively. RQ6 evaluates the time efficiency of LDFighter.

4.3 Results

RQ4: How effective is LDFighter on improving the safety performance of LLMs across different languages? To answer this question, we apply LDFighter to AdvBench and report the average LJR (Avg.LJR) of each LLM. For each LLM, we select the top three languages with the highest CI scores. To investigate the impact of the number of languages on Avg.LJR, we vary the number of languages K from 3 to 30 with step 3. The results show GPT-3.5, Gemini-pro and Llama2-13b achieves an Avg.LJR of 0.0 when using the top three languages, and then remain unchanged as k increases. For Gemma-7b, the Avg.LJR first decreases to 3.33% and then rises slightly and stabilizes around 0.1 after $k = 18$. These results suggest that a larger value of k is not necessarily better and thus we recommend users select K languages from high-resource options rather than low-resource ones when using LDFighter. We refer readers to the technical appendix for the complete results of Avg.LJR.

In addition to addressing inherent multilingual challenges, we conduct an experiment to assess LDFighter's effectiveness in defending against jailbreaking attacks. We randomly select 100 valid adversarial prompts targeting Llama2-7b from the dataset created by CASPER [Zhao *et al.*, 2023b], generated using the GCG [Zou *et al.*, 2023] and generate 100 adversarial prompts targeting Llama2-7b using AutoDAN [Zhu *et al.*, 2023]. Subsequently, LDFighter is applied to Llama2-7b to handle the two types of adversarial prompts. To determine the optimal language selection strategy, we consider the top-k languages based on their average CI score from RQ3. Varying k from 3 to 10, we record the corresponding attack success rates (ASR). Note that all selected adversarial prompts are in English and validated, resulting in an initial ASR of 100% on Llama2-7b. The results (Figure 3) show that the application of LDFighter leads to a significant reduction in ASR. Even with just the top 3 languages (including English), the ASR plummets to 9.68% (GCG) and 31% (AutoDAN) from the baseline 100%. Furthermore, utilizing the top 10 languages, the ASR drops to a mere 3% (GCG) and 10% (AutoDAN). These results underscore the potential of LDFighter in thwarting jailbreaking attacks.

False alarm rate of LDFighter. We apply LDFighter to the NQ dataset across four LLMs to evaluate the false alarm rate. The results show that for Gemma-7b and Llama2-13b, only 3.33% and 6.67% of refusal responses are related to safety or ethical concerns, and for the remaining LLMs, this rate is 0.0% when using only the top three languages.

RQ5: How effective is LDFighter on improving the response quality of LLMs across different languages? To answer this question, we apply LDFighter to the NQ dataset and evaluate the average F_1-score of four LLMs using the same top-k settings as in RQ4. Figure 4 shows the average F_1-score of four

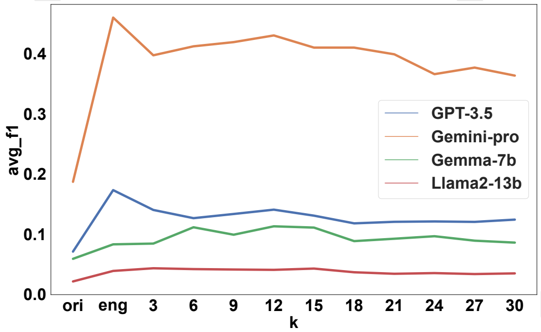

Figure 4: F_1-score of LLMs with LDFighter

LLMs, with and without LDFighter. Compared to the original average F_1-score, all LLMs experience an increase in average F_1-score with LDFighter, though the extent varies. For GPT-3.5 and Gemini-pro, the average F_1-score rises significantly and peaks when using only English in LDFighter. As k continues to increase, the average F_1-score slightly decreases but still remains higher than the original average. For Gemma-7b and Llama2-13b, the average F_1-score steadily improves at first, reaching its maximum at $k = 12$ for Gemma-7b and K=3 for Llama2-13b, after which it stabilizes.

RQ6: What is the time overhead of LDFighter? The time overhead of LDFighter primarily consists of three parts: translation, querying, and encoding. Compared to the time spent interacting with LLMs, the time required for similarity-based voting can be considered negligible if the embeddings of each response have already been prepared. Let t_{tra}, t_{qry}, and t_{emd} represent the time costs of translation, querying with an LLM, and sentence encoding, respectively. The overall time cost of applying LDFighter can be estimated as follows:

$$cost = 2k \cdot t_{tra} + k \cdot t_{qry} + k \cdot t_{emd} \tag{3}$$

where k is the number of languages used in LDFighter. The translation time is $2k \cdot t_{tra}$ because we need to translate both the original query and responses in different languages.

The time overhead varies linearly with the value of k, and can be reduced by parallelizing the computation process of k languages. We thus report the cost of LDFighter with k=1. On average, the time cost per query is 9.96 seconds on GPT-3.5, 4.91 seconds on Gemini-pro, 9.68 seconds on Gemma-7b, and 11.75 seconds on Llama2-13b.

5 Related Work

This work is related to multilingual LLM jailbreak. Puttaparthi et al. [Puttaparthi *et al.*, 2023] explore the reliability of ChatGPT for multilingual queries, assessing its jailbreak rate across 30 malicious questions and 121 languages in four scenarios: single language, mixed languages, responses in a different language, and multilingual wrapping. Yong et al. [Yong *et al.*, 2023] analyze GPT-4's safety performance across 12 low-resource languages, and find that GPT-4 fails to generalize its safety mechanisms to low-resource languages. Li et al. [Li *et al.*, 2024] conduct an empirical study of multilingual LLM jailbreak attacks. They create a multilingual jailbreak dataset to assess the safety performance of four LLMs across nine languages. They also examine attention distribution on failed and successful jailbreak cases and reveal that LLMs have a more balanced attention distribution on failed jailbreak cases. Deng et al. [Deng *et al.*, 2024] investigate the multilingual jailbreak of GPT-3.5 and GPT-4 across 30 languages under both unintentional and intentional scenarios. In the unintentional scenario, input questions are not deliberately modified or wrapped with malicious prompt, while in the intentional scenario, input questions are carefully designed to bypass the safety mechanisms. Previous research has either been limited in language coverage or focused primarily on the GPT series. In contrast, our study examines a broader range of languages and incorporates both recent closed and open-source LLMs.

This work is related to the defense of LLM jailbreak. Robey et al. [Robey *et al.*, 2023] propose a defense against the GCG attack [Zou *et al.*, 2023]. This approach begins by creating a set of mutated inputs through character-level perturbations of the original input. Such perturbation may render the adversarial suffixes invalid. After that, the approach decides the input whether is harmful by majority voting on responses of the mutated inputs. If more than half of responses are refusals, the input is then classified as harmful. However, this approach may not be effective against natural jailbreak questions generated by translation-based attacks. To tack the multilingual jailbreak threats, Deng et al.[Deng *et al.*, 2024] and Li et al. [Li *et al.*, 2024] propose to fine-tune LLMs with a set of multilingual input-output pairs that encompass both unsafe and general query questions. Although effective, fine-tuning LLMs can be time-consuming and complex. In contrast, LDFighter offers a convenient alternative that does not require modifying the LLM itself and can be easily implemented as a plug-and-play solution.

This work is also related to the multilingual discrimination in natural language processing. Blasi et al. [Blasi *et al.*, 2022] systematically investigate multilingual inequalities in language technologies, examining user-facing technologies such as question answering, as well as foundational NLP tasks like dependency parsing across the world's languages. Their study reveals significant disparities in the development of language technologies across different languages. Sourojit Ghosh and Aylin Caliskan [Ghosh and Caliskan, 2023] evaluating ChatGPT's translation performance in several low-resource languages and discovers that ChatGPT tends to perpetuate gender defaults and stereotypes associated with certain occupations. We remark that our work focuses on a wide spectrum of LLMs and evaluates their multilingual discrimination from both safety and quality aspects.

6 Conclusion

In this work, we evaluate the consistency of LLM outputs in response to queries in various languages from two aspects, focusing on safety and quality. Through experiments involving four state-of-the-art LLMs and 74 languages, we identify significant linguistic discrepancies in LLM performance across different languages. To address these issues, we introduce LDFighter, a lightweight approach that enhances the safety and quality of LLM responses while ensuring consistent service for speakers of all languages.

Acknowledgements

This research is supported by the Ministry of Education, Singapore under its Academic Research Fund Tier 2 (Award ID: T2EP20222-0037). This work is also funded by Zhejiang Province High-Level Talents Special Support Program "Leading Talent of Technological Innovation of Ten-Thousands Talents Program" (No. 2022R52046), and Information Technology Center, ZheJiang University.

References

[Barrault *et al.*, 2023] Loïc Barrault, Yu-An Chung, Mariano Coria Meglioli, David Dale, Ning Dong, Mark Duppenthaler, Paul-Ambroise Duquenne, Brian Ellis, Hady Elsahar, Justin Haaheim, et al. Seamless: Multilingual expressive and streaming speech translation. *arXiv preprint arXiv:2312.05187*, 2023.

[Blasi *et al.*, 2022] Damián Blasi, Antonios Anastasopoulos, and Graham Neubig. Systematic inequalities in language technology performance across the world's languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022.

[Chang *et al.*, 2023] Tyler A Chang, Catherine Arnett, Zhuowen Tu, and Benjamin K Bergen. When is multilinguality a curse? language modeling for 250 high-and low-resource languages. *arXiv preprint arXiv:2311.09205*, 2023.

[Deng *et al.*, 2024] Yue Deng, Wenxuan Zhang, Sinno Jialin Pan, and Lidong Bing. Multilingual jailbreak challenges in large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

[Ghosh and Caliskan, 2023] Sourojit Ghosh and Aylin Caliskan. Chatgpt perpetuates gender bias in machine translation and ignores non-gendered pronouns: Findings across bengali and five other low-resource languages. In *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, pages 901–912, 2023.

[Google, 2024] Google. Introducing gemini 1.5, google's next-generation ai model. https://blog.google/technology/ai/google-gemini-next-generation-model-february-2024/, 2024. Accessed: 2024-05-18.

[Hendy *et al.*, 2023] Amr Hendy, Mohamed Abdelrehim, Amr Sharaf, Vikas Raunak, Mohamed Gabr, Hitokazu Matsushita, Young Jin Kim, Mohamed Afify, and Hany Hassan Awadalla. How good are gpt models at machine translation? a comprehensive evaluation. *arXiv preprint arXiv:2302.09210*, 2023.

[Kwiatkowski *et al.*, 2019] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019.

[Lankford *et al.*, 2023] Séamus Lankford, Haithem Afli, and Andy Way. adaptmllm: Fine-tuning multilingual language models on low-resource languages with integrated llm playgrounds. *Information*, 14(12):638, 2023.

[Li *et al.*, 2024] Jie Li, Yi Liu, Chongyang Liu, Ling Shi, Xiaoning Ren, Yaowen Zheng, Yang Liu, and Yinxing Xue. A Cross-Language Investigation into Jailbreak Attacks in Large Language Models, January 2024.

[Liang *et al.*, 2022] Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022.

[OpenAI, 2022] OpenAI. Introducing chatgpt. https://openai.com/index/chatgpt, 2022. Accessed: 2025-05-18.

[Peng *et al.*, 2023] Keqin Peng, Liang Ding, Qihuang Zhong, Li Shen, Xuebo Liu, Min Zhang, Yuanxin Ouyang, and Dacheng Tao. Towards making the most of chatgpt for machine translation. *arXiv preprint arXiv:2303.13780*, 2023.

[Puttaparthi *et al.*, 2023] Poorna Chander Reddy Puttaparthi, Soham Sanjay Deo, Hakan Gul, Yiming Tang, Weiyi Shang, and Zhe Yu. Comprehensive evaluation of chatgpt reliability through multilingual inquiries. *arXiv preprint arXiv:2312.10524*, 2023.

[Robey *et al.*, 2023] Alexander Robey, Eric Wong, Hamed Hassani, and George J Pappas. Smoothllm: Defending large language models against jailbreaking attacks. *arXiv preprint arXiv:2310.03684*, 2023.

[Segura *et al.*, 2016] Sergio Segura, Gordon Fraser, Ana B Sanchez, and Antonio Ruiz-Cortés. A survey on metamorphic testing. *IEEE Transactions on software engineering*, 42(9):805–824, 2016.

[Tan *et al.*, 2023] Yiming Tan, Dehai Min, Yu Li, Wenbo Li, Nan Hu, Yongrui Chen, and Guilin Qi. Evaluation of chatgpt as a question answering system for answering complex questions. *arXiv preprint arXiv:2303.07992*, 2023.

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

[Team *et al.*, 2024] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay

Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

[Touvron *et al.*, 2023] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[Wei *et al.*, 2023] Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, et al. Zero-shot information extraction via chatting with chatgpt. *arXiv preprint arXiv:2302.10205*, 2023.

[Yong *et al.*, 2023] Zheng-Xin Yong, Cristina Menghini, and Stephen H Bach. Low-resource languages jailbreak gpt-4. *arXiv preprint arXiv:2310.02446*, 2023.

[Zhao *et al.*, 2023a] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A Survey of Large Language Models, November 2023.

[Zhao *et al.*, 2023b] Wei Zhao, Zhe Li, and Jun Sun. Causality analysis for evaluating the security of large language models. *arXiv preprint arXiv:2312.07876*, 2023.

[Zhu *et al.*, 2023] Sicheng Zhu, Ruiyi Zhang, Bang An, Gang Wu, Joe Barrow, Zichao Wang, Furong Huang, Ani Nenkova, and Tong Sun. Autodan: Automatic and interpretable adversarial attacks on large language models. *CoRR*, abs/2310.15140, 2023.

[Zou *et al.*, 2023] Andy Zou, Zifan Wang, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.