Linguistics Theory Meets LLM: Code-Switched Text Generation via Equivalence Constrained Large Language Models

Garry Kuwanto[1], **Chaitanya Agarwal**[2], **Genta Indra Winata**[†3*], **Derry Tanti Wijaya**[†1,4]
[1]Boston University [2]Deccan AI [3]Capital One [4]Monash University Indonesia
gkuwanto@bu.edu, chaitanya@deccan.ai
genta.winata@capitalone.com, derry.wijaya@monash.edu

Abstract

Code-switching is a common practice for millions of multilingual speakers but remains challenging for Large Language Models (LLMs). This paper investigates LLM capabilities in generating code-switched text, conducting extensive experiments across five diverse language pairs: English paired with Hindi, Tamil, Malayalam, and Indonesian, as well as Indonesian-Javanese. Our analysis, grounded in comprehensive human evaluations by native speakers, uncovers a directional asymmetry: LLMs consistently produce higher-quality (more accurate and fluent) code-switched text when prompted with a lower-resource language (e.g., Hindi, Tamil, Javanese) as the source, compared to when a higher-resource language (English, Indonesian) serves as the source. This asymmetry mirrors sociolinguistic patterns, particularly the Matrix Language Frame model, suggesting LLMs implicitly learn common code-switching structures from their training data where regional languages often form the grammatical base. Furthermore, we find that explicit linguistic guidance, applied through Equivalence Constraint Theory (ECT) to identify switching points, primarily benefits generation quality only in the less common, higher-resource-source direction where LLMs intrinsically struggle. These findings highlight a crucial interplay between the implicit linguistic knowledge captured by LLMs and the targeted utility of explicit linguistic constraints. We also introduce CSPREF, a pairwise preference dataset derived from our human evaluations, to facilitate future research in code-switching generation and evaluation.

1 Introduction

Bilingual and multilingual speakers frequently engage in code-switching, the phenomenon where speakers alternate between languages within a single discourse. The widespread of this linguistic phenomenon follows complex syntactic, semantic, and sociolinguistic patterns rather than occurring randomly or as an indication of the lack of language proficiency. Despite being so widespread, code-switching still remains a challenge to model effectively, creating significant barriers to building truly inclusive language technologies.

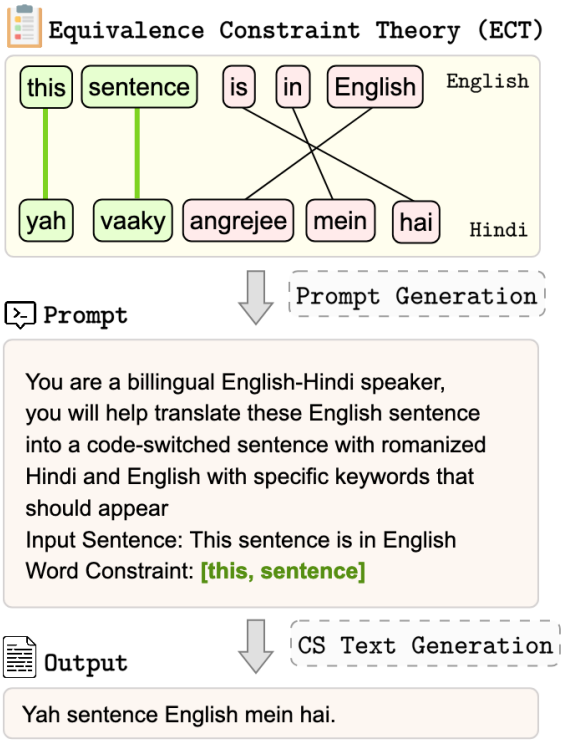

Figure 1: Example of CSPREF. The top panel shows the word-level alignment between English and Hindi. The middle panel displays the input to the LLM, including the original English sentence and word constraints derived from ECT. The bottom panel shows the resulting code-switched output sentence generated by the LLM.

The computational implementation of linguistic theories explaining code-switching constraints has proven difficult. Established frameworks like Equivalence Constraint Theory (Poplack, 1980) and Matrix Language Frame model (Myers-

* †The authors are senior authors.

Proceedings of the 1st Workshop on Computational Developmental Linguistics (CDL), pages 1–14

Scotton, 1997) offer theoretical explanations, but their application in NLP systems remains limited. Current approaches typically adopt either purely data-driven methods or focus on complex syntactic rules (Bhat et al., 2016), with few successfully bridging these approaches. Pratapa and Choudhury (2021) implements linguistic constraints using parse trees from parallel sentences. Similarly, Winata et al. (2019) applies linguistic constraints to generate synthetic code-switched text, finding that combining real and synthetic data improves performance for pretraining Language Models. Large Language Models (LLM) while demonstrating impressive cross-lingual capabilities, continue to struggle with generating natural code-switching (Winata et al., 2021; Zhang et al., 2023) often producing unnatural switching points due to insufficient training data of the same distribution.

To facilitate a comprehensive analysis of code-switching generation and evaluation, we conduct extensive experiments across five diverse language pairs (English-Hindi, English-Tamil, English-Malayalam, English-Indonesian, and Indonesian-Javanese). Our research reveals a striking directional asymmetry in code-switching quality that has significant implications for linguistic inclusion and technological equity. Our major contributions can be summarized as follows:

- We uncover a consistent pattern where LLMs generate higher-quality code-switched text when lower-resource languages serve as the source, without requiring explicit constraints. Conversely, when higher-resource languages serve as the source, additional linguistic guidance significantly improves quality.

- We demonstrate that this directional asymmetry mirrors sociolinguistic patterns observed in natural multilingual communication, where regional languages typically serve as the matrix language while global languages contribute embedded terms.

- We present a comprehensive evaluation dataset that prioritizes native speakers' judgments of code-switched text quality, offering a valuable resource for future research on evaluating code-switching across different language pairs and switching directions.

2 Linguistic Theories of Code-Switching

2.1 Code-Switching Patterns and Constraints

Code-switching, the alternation between two or more languages within a single discourse, represents a complex linguistic phenomenon observed across multilingual communities worldwide. Far from being random or indicative of language deficiency, code-switching follows systematic patterns governed by grammatical and sociocultural constraints (Poplack, 1980; Myers-Scotton, 1994; Muysken and ebrary, 2000.). Research has identified several common patterns, including inter-sentential switching (between sentences), intra-sentential switching (within a sentence), and tag-switching (insertion of a tag phrase) (Jan-Petter and Gumperz, 2007; Poplack, 1988).

The structural patterns of code-switching vary significantly across language pairs and communities. For instance, noun phrases are frequently switched in Spanish-English code-switching (Pfaff, 1979), while function words typically remain in the matrix language in Chinese-English mixing (Kamwangamalu and CHER-LENG, 1991). Several factors influence these patterns, including typological similarity between languages, lexical gaps, and discourse functions (Bullock and Toribio, 2009).

Sociolinguistic research has demonstrated that code-switching serves various communicative functions, including expressing solidarity, conveying nuanced meanings, establishing social identity, and filling lexical gaps (Gumperz, 1982; Myers-Scotton, 1993; Auer, 2013). Among bilingual communities with English as one of their languages, a common pattern emerges where the regional language often serves as the matrix language while English provides specific technical terminology, especially in domains like technology, education, and business (Bentahila, 1983; Wei, 2000).

2.2 Equivalence Constraint Theory

The Equivalence Constraint Theory (ECT), first proposed by Poplack (1980), represents one of the most influential frameworks for understanding the grammatical constraints on intra-sentential code-switching. The central premise of ECT is that code-switching is permissible only at points where the surface structures of the two languages align, meaning that switching does not violate the syntactic rules of either language involved.

More formally, ECT posits that code-switching

tends to occur at points where the word order requirements of both languages are satisfied simultaneously. If a potential switch point would create a structure that violates the grammar of either language, speakers typically avoid switching at that point. Sankoff (1998) formalized this constraint, stating that switching is prohibited if it generates a surface structure that would be ungrammatical in either language.

For example, in English-Spanish code-switching, switching between an adjective and noun is permissible when moving from English to Spanish but constrained in the opposite direction due to the differing adjective placement rules (English places adjectives before nouns, while Spanish typically places them after). Consider the sentence: "I bought a libro rojo" (I bought a red book), where switching occurs before "libro" (book). This switch respects both English syntax (determiner before noun) and Spanish syntax (noun before adjective). However, "I bought a rojo libro" would violate Spanish word order rules and is thus less likely to occur naturally (Poplack, 1978).

While ECT has proven effective in predicting many code-switching patterns, it faces limitations with typologically distant languages or in cases where one language strongly dominates as the matrix language (Muysken and ebrary, 2000.; Bhatt, 2013).

2.3 Matrix Language Frame Model

The Matrix Language Frame (MLF) model, developed by Myers-Scotton (1993, 1997), offers an alternative framework that addresses some limitations of ECT, particularly for language pairs with significant structural differences. This model distinguishes between the "matrix language" (ML), which provides the morphosyntactic framework, and the "embedded language" (EL), which contributes specific lexical items. According to the MLF model, the matrix language determines the overall grammatical structure, including word order and functional morphemes, while the embedded language contributes primarily content morphemes. Two key principles govern this interaction:

The Morpheme Order Principle: The order of morphemes must follow that of the matrix language. The System Morpheme Principle: System morphemes (functional elements like determiners, tense markers) must come from the matrix language.

This asymmetric relationship between languages helps explain why certain switching patterns occur more frequently than others. For instance, in many bilingual communities where a regional language interacts with English, the regional language typically serves as the matrix language, with English nouns and technical terminology embedded within the syntactic framework of the regional language (Bhatt, 2013; Sebba, 2009). The MLF model is particularly relevant for understanding directional asymmetry in code-switching patterns. It predicts that switching from the matrix language to the embedded language for content words (especially nouns and adjectives) is more common than the reverse direction (Myers-Scotton, 2002).

2.4 Directional Asymmetry in Natural Code-Switching

A significant yet often overlooked aspect of code-switching is its directional asymmetry across language pairs. Research has consistently demonstrated that the patterns, frequency, and constraints of code-switching differ depending on which language serves as the primary or matrix language (Gardner-Chloros, 2009; Sebba, 2009).

This asymmetry is particularly evident in post-colonial and globalized contexts where English interacts with regional languages. Studies across diverse communities—from Hindi-English in India (Kachru, 1978; Bhatt, 1997) to Swahili-English in East Africa (Myers-Scotton, 2002) and Spanish-English in the United States (Poplack, 1980)—reveal a consistent pattern: when the regional language serves as the matrix language, English lexical items are frequently embedded, especially for technical, academic, or professional domains. However, when English serves as the matrix language, embeddings from the regional language tend to be more limited and often serve cultural or identity-marking functions (Bullock and Toribio, 2009).

Several factors contribute to this asymmetry **Sociolinguistic status**: The relative prestige and domains of use for each language influence switching patterns (Myers-Scotton, 1993). **Lexical accessibility**: Terms may be more readily available or precise in one language than another (Heredia and Altarriba, 2001). **Processing constraints**: The cognitive mechanisms underlying language production may favor certain switching directions (Green and Abutalebi, 2013). **Communicative functions**: Different switching directions serve distinct discourse purposes (Gumperz, 1982).

The asymmetric nature of code-switching has important implications for computational modeling. Models trained primarily on data where one language consistently serves as the matrix may develop biases that affect their ability to generate natural code-switching in the opposite direction. This directional sensitivity suggests that linguistic constraints may be more critical for guiding generation in directions that are less represented in naturally occurring data.

Understanding these directional asymmetries is crucial for developing computational approaches that accurately reflect the complex patterns observed in natural code-switching across diverse multilingual communities.

3 Methodology

3.1 Experimental Design

To investigate directional asymmetry in code-switching, we designed a comprehensive evaluation framework examining code-switched text generation across five diverse language pairs: English-Hindi (en-hi), English-Tamil (en-ta), English-Malayalam (en-ml), English-Indonesian (en-id), and Indonesian-Javanese (id-jv). These pairs were selected to represent varying typological distances, resource availability, and sociolinguistic contexts. For each pair, we examined both possible directional flows: higher-resource language to lower-resource language (e.g., English→ Hindi) and lower-resource language to higher-resource language (e.g., Hindi→ English). Our experimental design focused on evaluating the quality of generated code-switched text under different conditions:

1. Direct Generation: Direct prompting of LLMs to produce code-switched text without explicit linguistic constraints

2. Linguistically-Guided Generation: Generation guided by constraints derived from linguistic theories, specifically Equivalence Constraint Theory (ECT)

3.2 Data Preparation

3.2.1 Obtaining Translations

We utilized existing parallel datasets to obtain high-quality translations across our target language pairs. For Hindi, we used the HinGE dataset (Srivastava and Singh, 2021), while Tamil and Malayalam translations came from the Samanantar dataset (Ramesh et al., 2022). For Indonesian-English and Indonesian-Javanese, we used the NusaX dataset (Winata et al., 2023b). These human-translated parallel sentences provided a reliable basis for our experiments.

To ensure consistency across experiments, we also generated translations using Llama3 8B, providing a controlled alternative to the human translations. This dual approach allowed us to assess whether any observed directional asymmetry was consistent across both human and machine translations.

3.2.2 Bitext Alignment

For each parallel sentence pair, we applied the GIZA++ tool (Och and Ney, 2003) to obtain word-level alignments between the source and target languages. These alignments were crucial for identifying potential code-switching points according to linguistic constraints.

3.3 Code-Switched Sentence Generation

3.3.1 Direct Generation

For Direct Generation, we prompted large language models to generate code-switched text without providing explicit linguistic constraints. The prompt simply instructed the model to create a code-switched version of the input sentence, incorporating elements from both languages naturally.

3.3.2 Linguistically-Constraint Generation

For the linguistically-guided approach, we incorporated switching constraints derived from Equivalence Constraint Theory. ECT posits that code-switching is natural at points where the word order rules of both languages align, avoiding violations of either language's syntax. We operationalized ECT by using word alignments to identify non-crossing alignment points as valid switching locations. For each input sentence, we constructed prompts that guided the model to generate the sentence with words that are acquired from

3.4 Evaluation Framework

To ensure a robust assessment of code-switching quality, we developed a comprehensive evaluation framework

3.4.1 Automatic Evaluation

Our primary evaluation methodology employs GPT-4o-mini as an automated assessor of code-switching quality. This approach demonstrates substantially higher correlation with human judgments

compared to traditional metrics, with Kendall's tau coefficients of 0.558 for accuracy and 0.514 for fluency (as detailed in 4.

We provide GPT-4o-mini with identical instructions as our human evaluators, asking it to rate generated sentences on discrete scales from 1 (lowest) to 3 (highest) for both accuracy and fluency. The prompt includes:

The original sentence in the first language (L1) The corresponding sentence in the second language (L2) The generated code-switched output

This structured approach allows GPT-4o-mini to perform consistent evaluations across different language pairs and generation methods, focusing specifically on meaning preservation (accuracy) and natural-sounding integration of languages (fluency).

3.4.2 Human Evaluation

We perform human evaluations with native bilingual speakers who actively code-switch in daily life. Indic language evaluators are recruited through DeccanAI (previously SoulAI), which ethically sources and trains annotators in India after assessing their English and native language proficiency. Indonesian evaluators are separately recruited. All annotators score the code-switched sentences for accuracy and fluency using established annotation guidelines.

This on 150 sample of inputs from dataset described in Section 4.3 with 18 different generation settings. Totaling 2700 sentence to rate for each language. In total, we conduct 24,300 human evaluations in total. For each code-switched sentence, we ask 3 unique evaluators to score the accuracy and fluency of the sentence on a discrete scale from 1 (lowest) to 3 (highest). While evaluating, the evaluators can see the parallel sentence pair (both languages) and the LLM generated code-switched sentence.

4 Experimental Setup

4.1 Models

We employ three distinct open-weight LLMs to assess the consistency of our findings across different architectures and training regimes:

- Aya 23 (8B): (Aryabumi et al., 2024) An LLM explicitly designed for multilingual tasks, trained on a diverse language corpus, making it potentially well-suited for code-switching.
- Llama 3 (8B): (Dubey et al., 2024) A widely-used model offering a balance between performance and computational efficiency within the Llama 3 series.
- Llama 3.1 (8B): An improved iteration of Llama 3 8B, incorporating refined training techniques and updated data, potentially offering enhanced capabilities in complex linguistic tasks like code-switching.

All experiments were conducted on a single NVIDIA L40 GPU equipped with 48GB of memory, ensuring consistency in the computational environment.

4.2 Language Pairs and Directions

Our investigation spans five language pairs, chosen to represent different language families, typological characteristics, and sociolinguistic contexts involving English and Indonesian as higher-resource languages: English-Hindi (en-hi), English-Tamil (en-ta), English-Malayalam (en-ml), English-Indonesian (en-id), Indonesian-Javanese (id-jv).

For each pair, we examine code-switched generation in two distinct directions to probe for asymmetry:

Higher-Resource to Code-Switched: Generation initiated from the higher-resource language (English for en-hi, en-ta, en-ml, en-id; Indonesian for id-jv), incorporating the lower-resource language.

Lower-Resource to Code-Switched: Generation initiated from the lower-resource language (Hindi, Tamil, Malayalam, Indonesian, or Javanese), incorporating the higher-resource language.

4.3 Datasets

Our experiments utilize five parallel corpora, each corresponding to a distinct language pair, as detailed in Table 1. For the human translations, we rely on the parallel sentences available in these datasets. In contrast, for the LLM translations we generate translations using the Llama3 8B model.

5 Results

5.1 Automatic Metrics

We first present the results using GPT-4o-mini as an automated evaluator for Accuracy (preserving meaning, GPT4o_a) and Fluency (naturalness of

Language Pair	Source	Size
Hindi-English (hi-en)	HinGE	2,766
Tamil-English (ta-en)	Samanantar (WAT 2020)	2,000
Malayalam-English (ml-en)	Samanantar (WAT 2020)	2,000
Indonesian-English (id-en)	NusaX	1,000
Indonesian-Javanese (id-en)	NusaX	1,000

Table 1: Summary of datasets used in the experiments, including source and size.

Method	Low	High	Δ
GPT4o$_a$			
Direct Generation	1.55	1.39	+0.26
Guided Gen. (Human Trans.)	1.58	1.47	+0.11
Guided Gen. (LLM Trans.)	1.59	1.47	+0.12
GPT4o$_f$			
Direct Generation	1.63	1.75	+0.12
Guided Gen. (Human Trans.)	1.53	1.67	+0.14
Guided Gen. (LLM Trans.)	1.53	1.64	+0.11

Table 2: Average GPT-4 evaluation of Accuracy (GPT4o$_a$) and Fluency (GPT4o$_f$) for the two direction generation across different methods. Scores are in the range of 1–3. Positive Δ shows that Low Resource is better than High Resource.

code-switching, GPT4o$_f$). While automatic metrics for code-switching are challenging, recent work suggests large models like GPT-4o can serve as reasonable proxies for certain quality aspects, especially when compared to traditional metrics (see Section 5.3). Table 2 shows the average scores across all five language pairs for each method and model, separated by the direction of generation.

The automatic evaluation scores in Table 2 provide initial evidence supporting directional asymmetry, with Direct Generation achieving higher accuracy (1.55 vs. 1.39) and fluency (1.75 vs. 1.63) when sourced from the lower-resource language. Interestingly, while the linguistically guided methods show some accuracy gains over direct generation, they appear to result in lower fluency scores according to GPT-4o across both directions. These preliminary automated results underscore the need for human evaluation to fully assess the nuances of code-switching quality, which we address next.

5.2 Human Evaluation Results

Given the limitations of automatic metrics for nuanced linguistic phenomena like code-switching fluency, we now turn to the human evaluation results presented in Table 3 as our primary basis for analysis. These scores, provided by native bilingual speakers actively using code-switching, offer crucial insights into the perceived accuracy and naturalness of the generated text.

lang	GPT4o$_a$	GPT4o$_f$	Human$_a$	Human$_f$
en-hi	**0.33**	**0.41**	**0.59**	0.53
en-ml	0.12	0.10	0.56	**0.63**
en-ta	0.07	0.12	0.45	0.39
id-jv	0.13	0.15	0.14	0.12
en-id	0.11	0.11	0.23	0.29

Table 3: Difference of Average between Low Resource and High Resource for Direct Generation. Difference Value is calculated by $\Delta_m = low_m - high_m$. All values are positive because Low Resource inputs consistently outperforms High Resource inputs. The higher the difference the higher the more prominent the asymmetry

Direct Generation The human evaluation results for direct generation reveal clear directional asymmetry across language pairs, with lower-resource languages consistently outperforming higher-resource languages as source inputs. English-Hindi shows the most pronounced asymmetry with human accuracy and fluency differences of 0.59 and 0.53 respectively, followed by English-Malayalam (0.56, 0.63) and English-Tamil (0.45, 0.39). The Indonesian-Javanese pair exhibits minimal asymmetry (0.14, 0.12), likely due to both languages occupying similar resource levels within their shared ecosystem. English-Indonesian shows moderate differences (0.23, 0.29). These patterns align with sociolinguistic observations that speakers naturally use their regional language as the grammatical foundation while incorporating English terms, rather than embedding regional elements within English grammatical structures.

5.3 Correlation Between Automatic and Human Metrics

To assess the reliability of automatic metrics for this task, we calculated the Kendall's Tau correlation between various automatic scores and the average human ratings (Accuracy and Fluency) on the human-evaluated subset (2,700 samples per language, aggregated). Table 4 shows these correlations

The results confirm findings from previous work (Guzmán et al., 2017; Winata et al., 2023a) that traditional metrics like BLEU and COMET show weak correlation with human judgments of code-switching quality, especially for fluency. No-

	Human$_a$	**Human$_f$**
Human$_a$	1.000	0.768
Human$_f$	0.768	1.000
GPT4o$_a$	0.558	0.504
GPT4o$_f$	0.540	0.514
COMET_avg	0.246	0.290
BLEU*	0.229	0.201

Table 4: Kendall's tau correlation scores between different automatic metrics and human evaluations for Human Accuracy and Human Fluency. *BLEU score can only be calculated for Hindi-English as there is no code-switched references for other language pairs.

Language Pair	**Total**	**Easy**	**Hard**
Hindi-English (hi-en)	17,460	9,621	7,839
Tamil-English (ta-en)	5,034	4,506	528
Malayalam-English (ml-en)	8,664	7,517	1,147
Indonesian-English (id-en)	22,430	2,394	20,036
Indonesian-Javanese (id-jv)	44,606	13,262	31,344

Table 5: Statistics of CSPREF for five language pairs (Hindi-English, Tamil-English, Malayalam-English, Indonesian-English, and Indonesian-Javanese). "Easy" pairs are defined as those with high rating differences, while "Hard" pairs are defined as those with low rating differences.

tably, our GPT-4o-mini based evaluations (GPT4o$_a$, GPT4o$_f$) demonstrate substantially higher correlation with human ratings (Tau around 0.51-0.56) compared to other automatic metrics. This supports its use as a more reliable proxy for large-scale automatic evaluation in this context, although human evaluation remains the gold standard.

5.4 Pairwise Preference Dataset

We construct CSPREF, a pairwise preference dataset using human ratings to evaluate the performance of different models in code-switched text generation. Each pair consists of two generated code-switched sentences compared based on their human-evaluated accuracy and fluency scores. To further analyze the performance, we split the dataset into **easy** and **hard** subsets. The **easy** subset includes pairs where the difference in human ratings is high (indicating a clear preference for one generated sentence), while the **hard** subset consists of pairs with minimal differences (indicating ambiguous preferences). Table 5 provides the statistics for the pairwise dataset across three languages: Hindi, Tamil, and Malayalam. We report the total number of pairs, as well as the breakdown into **easy** and **hard** subsets for each language pair.

6 Discussion

6.1 Explaining Directional Asymmetry

We found a directional asymmetry in code-switching: ECT-guided approaches significantly improve quality when higher-resource languages are the source, but offer minimal benefit with lower-resource source languages (like Indic or Austronesian languages). This aligns with the Matrix Language Frame model (Myers-Scotton, 1993), where one language provides the grammatical framework while another contributes lexical items. In multilingual communities, lower-resource regional languages typically serve as the matrix language with higher-resource languages providing embedded content words (Bhatt, 2013; Myers-Scotton, 2002). Our findings suggest LLMs have implicitly learned these natural patterns during pre-training, developing strong capabilities to generate code-switched text where lower-resource languages serve as the matrix. This matches sociolinguistic patterns in regions like India and Indonesia (Kachru, 1978; Nababan, 1991). Conversely, LLMs struggle with the less common pattern of higher-resource matrix languages with lower-resource embeddings unless explicitly guided by ECT constraints (Poplack, 1988).

6.2 Linguistic Constraints vs. Implicit Knowledge in LLMs

Our results reveal an interesting observation between explicit linguistic constraints and LLMs' implicit knowledge. In the lower-resource-to-higher-resource direction, LLMs generate fluent code-switched text without guidance, suggesting they've internalized common patterns from training data.

This contributes to the debate on whether LLMs truly learn linguistic rules or simply memorize patterns (McCoy et al., 2020; Linzen and Baroni, 2021). For frequent phenomena like code-switching from lower-resource to higher-resource languages, LLMs develop robust implicit knowledge aligning with linguistic theories like MLF. However, explicit constraints remain valuable for less common patterns.

GPT-4o-mini's effectiveness as an evaluator, with much higher correlation to human judgments than traditional metrics, reinforces this view. The model appears to have internalized not just genera-

tion capabilities but also human quality assessment criteria.

These findings point to a complementary relationship: linguistic theories like ECT provide valuable guidance for uncommon patterns, while LLMs excel at reproducing frequently observed phenomena without explicit constraints.

7 Related Work

Code-switching has been extensively studied from both linguistic and computational perspectives. Early linguistic theories, such as ECT (Poplack, 1980), establishes foundational principles for understanding syntactic boundaries in code-switching. Similarly, research by Joshi (1982) and Pfaff (1979) examine structural constraints and sentence processing in bilingual contexts. Recent computational approaches have adapted these theories into neural models. For instance, Winata et al. (2019) utilized ECT to generate synthetic data for training language models, while Gupta et al. (2020) employed pre-trained models to create code-switched text without explicit constraints. Pratapa and Choudhury (2021) utilized ECT to synthetically generate code-switched text by using the Dependency Tree. And Gupta et al. (2021) adopted a Machine Translation approach to the problem. Comprehensive survey by Sitaram et al. (2019); Winata et al. (2023a) outline the computational challenges and advancements in code-switching research.

Evaluation benchmarks, such as LinCE (Aguilar et al., 2020) and GLUECoS (Khanuja et al., 2020) have standardized model assessments across diverse tasks. Recent studies have also investigated automatic metrics for code-switching (Guzmán et al., 2017) and explored the use of LLMs in understanding code-switched text (De Leon et al., 2024), and also generating (Yong et al., 2023). In this context, our work builds on these foundations by integrating linguistic constraints into LLM-based generation, addressing existing limitations in fluency and accuracy evaluation.

8 Future Work

Future work could address these limitations through several avenues. Using the CSPREF dataset to fine-tune models specifically for code-switching generation could potentially improve performance without explicit constraints. Combining ECT with other theories like MLF might yield a more comprehensive approach to constraint-guided generation. Adapting generation to match individual code-switching patterns based on demographic and language proficiency factors represents another promising direction Figure 2. Creating specialized automatic metrics that better correlate with human judgments of code-switching quality would also significantly advance the field. The asymmetric results also suggest an intriguing direction for future work: exploring whether models can be trained to recognize the "matrix language" in a given context and automatically apply appropriate constraints based on the direction of generation.

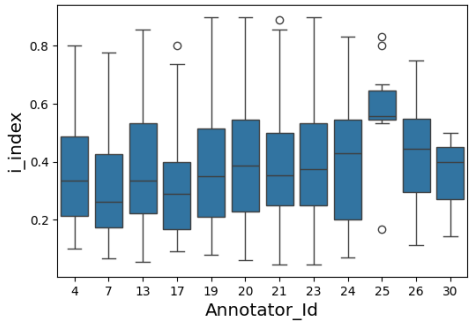

Figure 2: Distribution of I-index across annotators, representing the probability of code-switching at any given token. The I-index indicates the proportion of switch points relative to language-dependent tokens in the corpus. The variability in switching preferences among annotators highlights the individual differences in their judgment of fluency, suggesting that demographic factors may play a role in code-switching evaluation.

9 Conclusion

In this work,we investigated the capabilities of Large Language Models in generating code-switched text. Our evaluation across five language pairs revealed a striking directional asymmetry. Our results consistently show that models generate substantially more accurate and fluent code-switched text when prompted with a lower-resource language as the source, compared to when starting with a higher-resource language like English or Indonesian.

The asymmetry aligns with sociolinguistic patterns observed in natural code-switching and suggests that LLMs have implicitly learned common code-switching patterns during pre-training. Our findings demonstrate that linguistic theory and data-driven approaches can complement each other, with explicit constraints providing valuable guidance for less common code-switching patterns.

Our result underscore the importance of con-

sidering directionality and sociolinguistic context when developing and evaluating multilingual models. The challenges observed in automatic evaluation further emphasize the continued necessity of human judgment, a need we aimed to support by creating the CSPREF pairwise preference dataset. Ultimately, this work contributes to a more nuanced understanding of LLM capabilities and limitations in handling code-switching, paving the way for more linguistically informed and culturally aware language technologies.

Limitations

While our study provides valuable insights into code-switching generation, several limitations warrant discussion. Our language coverage, though spanning Indo-European, Dravidian, and Austronesian language families, could be expanded to include other language families, particularly tonal languages and those with substantially different writing systems, to strengthen our findings. The approach also focuses primarily on syntactic constraints and does not fully account for the sociolinguistic and pragmatic factors that influence code-switching in natural settings. Despite using GPT-4o-mini evaluation and human judgments, we still lack specialized metrics designed specifically for code-switching quality assessment. Additionally, our experiments are limited to relatively small open-source models; larger models might show different patterns or capabilities.

Ethics Statement

All aspects of this research were reviewed and approved by the Institutional Review Board of our organization. Data collection was conducted by DeccanAI for the Hindi, Tamil, and Malayalam evaluations. We compensate human evaluators INR 110 for every 18 sentences they evaluate, which typically takes around 20 minutes. This results in an effective pay rate of INR 330 per hour. The human evaluators work entirely remotely and interact with DeccanAI through their web platform. All evaluators are native speakers of the respective lower-resource languages they assess and are proficient in English. Their language proficiency is evaluated through custom online tests. Most evaluators come from major cities in India where these native languages are spoken and frequently engage in code-switched dialogues. DeccanAI provides training for the evaluators to ensure they are well-calibrated with the annotation guidelines.

For the Indonesian-Javanese language pair, annotators were recruited separately through our Indonesian university partners. These evaluators were compensated at a rate of IDR 2,000,000, in line with local research assistant compensation rates. All Indonesian annotators were native speakers of both Indonesian and Javanese, with most coming from various cities across Central and East Java, representing different dialectal backgrounds.

References

Gustavo Aguilar, Sudipta Kar, and Thamar Solorio. 2020. Lince: A centralized benchmark for linguistic code-switching evaluation. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1803–1813.

Viraat Aryabumi, John Dang, Dwarak Talupuru, Saurabh Dash, David Cairuz, Hangyu Lin, Bharat Venkitesh, Madeline Smith, Kelly Marchisio, Sebastian Ruder, and 1 others. 2024. Aya 23: Open weight releases to further multilingual progress. *arXiv preprint arXiv:2405.15032*.

Peter Auer. 2013. *Code-switching in conversation: Language, interaction and identity*. Routledge.

Abdelali Bentahila. 1983. Motivations for code-switching among arabic-french bilinguals in morocco. *Language & communication*, 3(3):233–243.

Gayatri Bhat, Monojit Choudhury, and Kalika Bali. 2016. Grammatical constraints on intra-sentential code-switching: From theories to working models. *Preprint*, arXiv:1612.04538.

Rakesh M Bhatt. 2013. Optimization in bilingual language use. *Bilingualism: Language and cognition*, 16(4):740–742.

Rakesh Mohan Bhatt. 1997. Code-switching, constraints, and optimal grammars. *Lingua*, 102(4):223–251.

Barbara E Bullock and Almeida Jacqueline Ed Toribio. 2009. *The Cambridge handbook of linguistic code-switching*. Cambridge university press.

Frances Adriana Laureano De Leon, Harish Tayyar Madabushi, and Mark Lee. 2024. Code-mixed probes show how pre-trained models generalise on code-switched text. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3457–3468.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Penelope Gardner-Chloros. 2009. *Code-switching*. Cambridge university press.

David W Green and Jubin Abutalebi. 2013. Language control in bilinguals: The adaptive control hypothesis. *Journal of cognitive psychology*, 25(5):515–530.

John J Gumperz. 1982. *Discourse strategies*. 1. Cambridge University Press.

Abhirut Gupta, Aditya Vavre, and Sunita Sarawagi. 2021. Training data augmentation for code-mixed translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5760–5766.

Deepak Gupta, Asif Ekbal, and Pushpak Bhattacharyya. 2020. A semi-supervised approach to generate the code-mixed text using pre-trained encoder and transfer learning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2267–2280.

Gualberto A Guzmán, Joseph Ricard, Jacqueline Serigos, Barbara E Bullock, and Almeida Jacqueline Toribio. 2017. Metrics for modeling code-switching across corpora. In *Interspeech*, pages 67–71.

Roberto R Heredia and Jeanette Altarriba. 2001. Bilingual language mixing: Why do bilinguals code-switch? *Current directions in psychological science*, 10(5):164–168.

Blom Jan-Petter and John J. Gumperz. 2007. Social meaning in linguistic structure: code-switching in norway. In *The Bilingualism Reader*.

Aravind Joshi. 1982. Processing of sentences with intra-sentential code-switching. In *Coling 1982: Proceedings of the Ninth International Conference on Computational Linguistics*.

Braj B Kachru. 1978. Toward structuring code-mixing: An indian perspective. *International Journal of the Sociology of Language*.

Nkonko M Kamwangamalu and LEE CHER-LENG. 1991. Chinese-english code-mixing: a case of matrix language assignment. *World Englishes*, 10(3):247–261.

Simran Khanuja, Sandipan Dandapat, Anirudh Srinivasan, Sunayana Sitaram, and Monojit Choudhury. 2020. Gluecos: An evaluation benchmark for code-switched nlp. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3575–3585.

Tal Linzen and Marco Baroni. 2021. Syntactic structure from deep learning. *Annual Review of Linguistics*, 7(1):195–212.

R. Thomas McCoy, Junghyun Min, and Tal Linzen. 2020. BERTs of a feather do not generalize together: Large variability in generalization across models with similar test set performance. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 217–227, Online. Association for Computational Linguistics.

Pieter. Muysken and Inc. ebrary. 2000. *Bilingual speech*. Cambridge University Press,, Cambridge, UK ;.

Carol Myers-Scotton. 1993. *Social motivations for codeswitching: Evidence from Africa*. Oxford University Press.

Carol Myers-Scotton. 1994. Social motivations for codeswitching. evidence from africa. *Multilingua-Journal of Interlanguage Communication*, 13(4):387–424.

Carol Myers-Scotton. 1997. *Duelling languages: Grammatical structure in codeswitching*. Oxford University Press.

Carol Myers-Scotton. 2002. *Contact linguistics: Bilingual encounters and grammatical outcomes*. Oxford University Press, USA.

PWJ Nababan. 1991. Language in education: The case of indonesia. *International review of education*, 37:115–131.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.

Carol W Pfaff. 1979. Constraints on language mixing: Intrasentential code-switching and borrowing in spanish/english. *Language*, pages 291–318.

Shana Poplack. 1978. *Syntactic structure and social function of code-switching*, volume 2. Centro de Estudios Puertorriqueños,[City University of New York].

Shana Poplack. 1980. Sometimes i'll start a sentence in spanish y termino en espanol: toward a typology of code-switching1. *Linguistics*, 18(7-8):581–618.

Shana Poplack. 1988. Contrasting patterns of code-switching in two communities. *Codeswitching: Anthropological and Sociolinguistic Perspectives. New York: Mouton de Gruyter*, pages 215–244.

Adithya Pratapa and Monojit Choudhury. 2021. Comparing grammatical theories of code-mixing. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 158–167.

Gowtham Ramesh, Sumanth Doddapaneni, Aravinth Bheemaraj, Mayank Jobanputra, Raghavan Ak, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Divyanshu Kakwani, Navneet Kumar, and 1 others. 2022. Samanantar: The largest publicly available parallel corpora collection for 11 indic languages. *Transactions of the Association for Computational Linguistics*, 10:145–162.

David Sankoff. 1998. A formal production-based explanation of the facts of code-switching. *Bilingualism: language and cognition*, 1(1):39–50.

Mark Sebba. 2009. Sociolinguistic approaches to writing systems research. *Writing systems research*, 1(1):35–49.

Sunayana Sitaram, Khyathi Raghavi Chandu, Sai Krishna Rallabandi, and Alan W Black. 2019. A survey of code-switched speech and language processing. *arXiv preprint arXiv:1904.00784*.

Vivek Srivastava and Mayank Singh. 2021. Hinge: A dataset for generation and evaluation of code-mixed hinglish text. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 200–208.

Lee Wei. 2000. *The bilingualism reader*, volume 11. Routledge London.

Genta Winata, Alham Fikri Aji, Zheng Xin Yong, and Thamar Solorio. 2023a. The decades progress on code-switching research in nlp: A systematic survey on trends and challenges. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2936–2978.

Genta Indra Winata, Alham Fikri Aji, Samuel Cahyawijaya, Rahmad Mahendra, Fajri Koto, Ade Romadhony, Kemal Kurniawan, David Moeljadi, Radityo Eko Prasojo, Pascale Fung, Timothy Baldwin, Jey Han Lau, Rico Sennrich, and Sebastian Ruder. 2023b. NusaX: Multilingual parallel sentiment dataset for 10 Indonesian local languages. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 815–834, Dubrovnik, Croatia. Association for Computational Linguistics.

Genta Indra Winata, Samuel Cahyawijaya, Zihan Liu, Zhaojiang Lin, Andrea Madotto, and Pascale Fung. 2021. Are multilingual models effective in code-switching? In *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, pages 142–153.

Genta Indra Winata, Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2019. Code-switched language models using neural based synthetic data from parallel sentences. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 271–280.

Zheng Xin Yong, Ruochen Zhang, Jessica Forde, Skyler Wang, Arjun Subramonian, Holy Lovenia, Samuel Cahyawijaya, Genta Winata, Lintang Sutawika, Jan Christian Blaise Cruz, and 1 others. 2023. Prompting multilingual large language models to generate code-mixed texts: The case of south east asian languages. In *Proceedings of the 6th Workshop on Computational Approaches to Linguistic Code-Switching*, pages 43–63.

Ruochen Zhang, Samuel Cahyawijaya, Jan Christian Blaise Cruz, Genta Winata, and Alham Aji. 2023. Multilingual large language models are not (yet) code-switchers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12567–12582.

A Relaxed Equivalence Constraint Theory (ECT)

Winata et al. (2019) apply ECT by simplifying sentences in terms of a linear grammatical structure and allowing lexical substitution on *non-crossing alignments* between parallel sentences (e.g., lexical substitution between "sentence" and "vaaky" in Figure 1). Denoting L_1 as the source language and L_2 as the target language, given a sentence in L_1 comprising an array of words $u_t = a_1, a_2, ..., a_m$ and a corresponding sentence in L_2 comprising an array of words $v_t = b_1, b_2, ..., b_m$, the alignment between a_i and b_i does not satisfy the constraint if there exists a pair a_j and b_j such that ($a_i < a_j$ and $b_i > b_j$) or ($a_i > a_j$ and $b_i < b_j$). If a switch occurs at this point, it alters the grammatical order in both languages, rendering the switch unacceptable. During the generation step, we permit *any* switches that do not violate this constraint.

This relaxation allows for greater flexibility in identifying potential switching points, accommodating the complexities of real-world code-switching patterns while maintaining grammatical coherence. Our implementation expands on the linear grammatical structure and non-crossing alignment criteria, introducing additional flexibility to capture a broader range of code-switching phenomena. In the following section, we outline our approach to implementing the relaxed ECT for identifying switching points, developing code-switched sentence generation techniques, and establishing a comprehensive evaluation framework.

B Switching Point Algorithm

The algorithm for the process of getting valid switching points is as described in Algorithm 1

C Prompt Details

In Table 6 we present the specific prompts used across different methods in our code-switching experiments. The Translate prompt is used when we generate the translations to get alignment. The Direct Generation prompt is used when we evaluate LLM to generate codeswitching. The ECT prompt is used for both Human translated and Machine

Algorithm 1: Identification of Valid Switching Points

```
Result: List of valid switching points
GetValidSwitchingPoints(pairs)
valid_pairs ← [ ];
for i ← 1 to length(pairs) do
    valid ← true;
    for j ← 1 to length(pairs) do
        (a_i, b_i) ← pairs[i];
        (a_j, b_j) ← pairs[j];
        if (a_i < a_j and b_i > b_j) or
         (a_i > a_j and b_i < b_j) then
            valid ← false;
            break;
        end
    end
    if valid then
        Append pairs[i] to valid_pairs;
    end
end
return valid_pairs;
```

Translated Linguistically-Guided Genration. The GPT Eval prompt defines the structure for evaluating code-switching output based on accuracy and fluency.

D Human Evaluation

D.1 Annotation Guidelines

The following guidelines are provided to human evaluators to assess the model's responses. Evaluators rate the generated sentences based on two criteria: **Accuracy** and **Fluency**. The original sentence is in English, Indian local languages (Hindi, Malayalam, and Tamil), and evaluators must adhere to the rubrics outlined below.

D.1.1 General Guidelines

- **MUST**: Be objective while rating the responses.
- **MUST**: Strictly follow the rubrics for Accuracy and Fluency evaluation.
- Score each criterion on a scale from 1 to 3, where 1 is the lowest and 3 is the highest.
- Ignore formatting, and any additional explanatory text generated by the language model. Only focus on meaning and context.
- If the model fails to generate a response, assign a score of 1 for both Accuracy and Fluency.

D.1.2 Accuracy

Accuracy measures how well the generated sentence preserves the meaning and information of the original sentence and whether the code-switched terms are used correctly. The scores are as follows:

- **1. Low Accuracy**:
  - Significant deviations from the original meaning.
  - Key information is missing, altered, or repeated redundantly.
  - Code-switched terms are incorrect or inappropriate.
  - Introduces new information not present in the original sentence.
  - Key details are altered or repeated redundantly.
- **2. Moderate Accuracy**:
  - Minor deviations from the original meaning.
  - Most key information is present but may have slight errors.
  - Most code-switched terms are appropriate but with minor mistakes.
- **3. High Accuracy**:
  - Preserves the original meaning fully.
  - All key information is present and correct.
  - Code-switched terms are accurate and appropriately used.

D.1.3 Fluency

Fluency measures how natural and easy to understand the generated sentence is, considering grammar, syntax, and the smooth integration of code-switching. The scores are as follows:

- **1. Low Fluency**:
  - The sentence is difficult to understand or awkward.
  - Poor grammar or syntax in either language.
  - Code-switching disrupts the flow of the sentence.

Method	Prompt
Translate	Translate the following `lang1` sentence to `lang2`: `<Input Sentence>`
Baseline	You are a Bilingual `lang1-lang2` speaker, you will help translate these `lang1` sentences into a code-mixed sentence with Romanized `lang2` and `lang1` `<Input Sentence>`
ECT Prompt	You are a Bilingual `lang1-lang2` speaker, you will help translate these `lang1` sentences into a code-mixed sentence with Romanized `lang2` and `lang1` with specific keywords that should to appear. `<Input Sentence>` Words wanted: `<List of Words>`
GPT Eval	You are provided with triplets of sentences. The first two sentence in each triplet is the original monolingual sentences. The second sentence is a generated code-switched sentence. Your task is to evaluate the generated sentence based on two criteria: Accuracy and Fluency. You will score each criterion on a scale from 1 to 3, where 1 is the lowest and 3 is the highest. When evaluating the generated sentences, focus on the content and meaning. Ignore any extra formatting, alignment artifacts, or additional explanatory text. Judge the sentence to determine its accuracy and fluency. original_l1: `<Original Lang1 Sentence>` original_l2: `<Original Lang2 Sentence>` generated: `<Code Switched Sentence>`

Table 6: Prompts used in our experiment.

- **2. Moderate Fluency**:
  - The sentence is understandable but may have awkward or unnatural phrasing.
  - Acceptable grammar and syntax in both languages.
  - Code-switching is somewhat smooth but not perfectly integrated.
- **3. High Fluency**:
  - The sentence is natural and easy to understand.
  - Good grammar and syntax in both languages.
  - Code-switching is smooth and seamless, enhancing the sentence flow.

D.2 Detailed Values

Table 7 shows the mean scores of GPT and Human judgements for all the combinations of experiments that we did.

D.3 Inter Annotator Agreement

As seen in Table 8, the inter-annotator agreement, measured by Krippendorff's alpha, reveals varying levels of consensus across languages, with the highest agreement for Hindi. While Fluency is generally lower, this is expected as Fluency is more of a subjective measure.

For Indonesian-Javanese, we observed lower agreement scores compared to other language pairs. This can be attributed to our annotators coming from different cities across Java, where regional variations in Javanese dialects led to different interpretations of certain common words. These dialectal differences affected how annotators judged the appropriateness of specific code-switched terms, particularly when evaluating fluency in contexts where regional expressions were used.

Throughout the evaluation process, we continuously monitored the quality of annotations by measuring inter-annotator agreement at regular intervals. If the agreement metric indicated significant divergence in scores, particularly when individual annotators' ratings deviated notably from the group consensus, we conducted alignment meetings. These meetings were used to clarify the guidelines and ensure a consistent understanding of the evaluation criteria among the annotators. During

		GPT4o$_a$		GPT4o$_f$		Human$_a$		Human$_f$	
	direction	high	low	high	low	high	low	high	low
lang	method								
en-hi	Direct Generation	1.69	2.02	1.75	2.14	1.75	2.33	1.79	2.32
	Guided Gen. (Human Trans.)	1.63	1.90	1.70	1.98	1.72	2.21	1.73	2.12
	Guided Gen. (LLM Trans.)	1.65	1.86	1.71	1.96	1.75	2.19	1.75	2.14
en-ml	Direct Generation	1.21	1.33	1.39	1.43	1.15	1.81	1.15	1.78
	Guided Gen. (Human Trans.)	1.36	1.27	1.46	1.38	1.17	1.72	1.16	1.66
	Guided Gen. (LLM Trans.)	1.38	1.33	1.48	1.40	1.16	1.66	1.15	1.61
en-ta	Direct Generation	1.26	1.33	1.31	1.43	1.11	1.56	1.15	1.54
	Guided Gen. (Human Trans.)	1.39	1.36	1.46	1.52	1.17	1.35	1.19	1.38
	Guided Gen. (LLM Trans.)	1.29	1.35	1.34	1.49	1.15	1.35	1.17	1.38
id-jv	Direct Generation	1.43	1.86	1.82	1.97	2.11	2.52	2.01	2.13
	Guided Gen. (Human Trans.)	1.48	1.56	1.62	1.69	2.27	2.40	1.97	1.99
	Guided Gen. (LLM Trans.)	1.48	1.52	1.66	1.62	2.29	2.36	1.99	1.97
en-id	Direct Generation	1.48	1.59	1.66	1.77	2.44	2.67	2.18	2.49
	Guided Gen. (Human Trans.)	1.51	1.56	1.67	1.77	2.53	2.56	2.20	2.26
	Guided Gen. (LLM Trans.)	1.58	1.61	1.77	1.74	2.47	2.52	2.17	2.21

Table 7: Mean scores of Human Accuracy (Human$_a$), Human Fluency (Human$_f$), and GPT4-based evaluations (GPT4o$_a$ for Accuracy and GPT4o$_f$ for Fluency). Scores are grouped by translation direction (Higher Resource to Lower Reousrce and vice versa.)

these sessions, any inconsistencies were discussed and resolved to improve consistency, especially in subjective aspects like Fluency. This iterative process helped ensure the reliability of the final evaluations and minimized discrepancies in the ratings.

Language	**Fluency**	**Accuracy**
Tamil-English	0.321	0.445
Malayalam-English	0.405	0.423
Hindi-English	0.646	0.720
Indonesian-English	0.535	0.606
Indonesian-Javanese	0.274	0.317

Table 8: Krippendorff's alpha scores for inter-annotator agreement on Fluency and Accuracy across Tamil, Malayalam, and Hindi.