Between Languages: How Well Do LLMs Navigate Code-Switching?

Anonymous ACL submission

Abstract

Code-switching (CSW) is the act of alternating between two or more languages within a single discourse. This phenomenon is widespread in multilingual communities, and increasingly prevalent in online content, where users naturally mix languages in everyday communication. As a result, Large Language Models (LLMs), now central to content processing and generation, are frequently exposed to code-switched inputs. Given their widespread use, it is crucial to understand how LLMs process and reason about such mixed-language text. This paper presents a systematic evaluation of LLM comprehension under code-switching by generating CSW variants of established reasoning and comprehension benchmarks. While degradation is evident when foreign tokens disrupt English text—even under linguistic constraints—embedding English into other languages often improves comprehension. Though prompting yields mixed results, fine-tuning offers a more stable path to degradation mitigation.

1 Introduction

Code-switching (CSW)—the act of alternating between two or more languages within a single discourse (Das et al., 2023; Zhang et al., 2023; Ochieng et al., 2024)—is a common phenomenon in multilingual communities (Bullock and Toribio, 2009; Parekh et al., 2020; Doğruöz et al., 2021), and increasingly prevalent in online content (Kodali et al., 2024), where users naturally mix languages in everyday informal communications.

Large Language Models (LLMs) have demonstrated remarkable capabilities across a wide range of natural language processing tasks (Zhao et al., 2023). As they are increasingly used to process and generate content, the widespread availability of code-switched inputs makes it crucial to understand how LLMs reason about such mixed-language data, and whether their multilingual fluency reflects genuine understanding or superficial pattern matching (Zhang et al., 2023). To systematically assess LLMs' handling of such data, we turn to insights from linguistic theories that define the structural constraints governing natural code-switching.

Linguistic theories have long studied the structure of code-switching, proposing formal constraints on permissible switch points, such as the Equivalence Constraint Theory (ECT), which posits that switches occur at positions where the surface structures of both languages are grammatically compatible (Poplack, 1978), and the Matrix Language Frame model (MLF), which distinguishes between a Matrix Language (ML) that provides the grammatical frame of the clause and an Embedded Language (EL) that contributes inserted content without disrupting this structure (Myers-Scotton, 1993). These frameworks aim to identify the grammatical boundaries and syntactic compatibility that make code-switching possible and natural. While such theories offer testable hypotheses for analyzing CSW, current efforts in synthetic CSW generation often prioritize producing fluent mixed-language text over probing whether LLMs genuinely internalize and apply these structural constraints in their reasoning (Pratapa et al., 2018; Potter and Yuan, 2024; Kuwanto et al., 2024; Heredia et al., 2025).

Despite the availability of well-established linguistic theories, existing evaluation benchmarks fall short of leveraging these insights to assess deeper comprehension in code-switched contexts. Current benchmarks for evaluating the code-switching capabilities of language models primarily focus on surface-level tasks such as language identification, sentiment analysis, and sequence labeling (Khanuja et al., 2020; Aguilar et al., 2020; Patwa et al., 2020). However, they largely overlook the challenge of evaluating deeper reasoning and semantic understanding in mixed-language settings

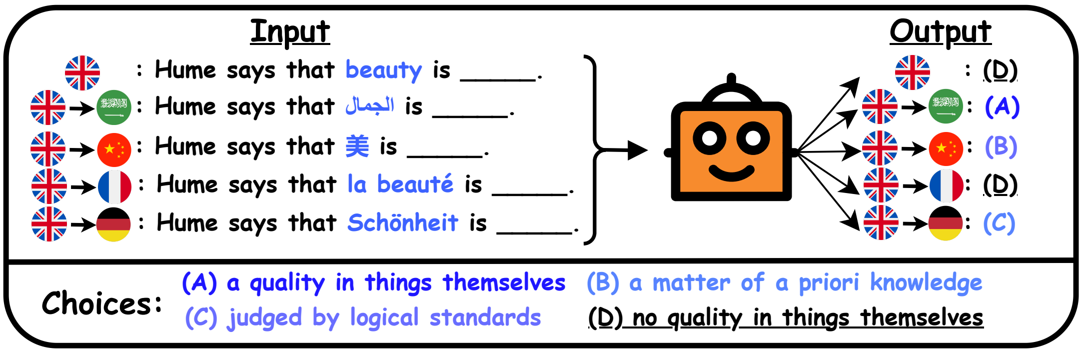

Figure 1: An example illustrating the noun-token code-switching methodology from Experiment 1. The figure demonstrates how different embedded languages (Arabic, French, German, Chinese) for the noun "beauty" in an English matrix sentence can lead to varied model outputs.

(Yadav et al., 2024; Gupta et al., 2024; Ng and Chan, 2024), leaving a critical gap in assessing the true extent of LLMs' code-switched comprehension abilities.

To address these gaps, we introduce a systematic evaluation framework that leverages a constrained, multi-step LLM pipeline to generate linguistically grounded code-switched variants of established benchmarks in reading comprehension, multi-domain knowledge, and natural language inference. Code, data, and benchmarks are publicly available[1]. Our experiments reveal that code-switching has a nuanced impact on LLM comprehension, influenced by the languages involved and the switching style. In particular:

- Embedding non-English tokens into an English matrix language consistently degrades performance, even when the switches follow linguistic constraints, suggesting a structural vulnerability that cannot be explained solely by token-level unfamiliarity.
- Embedding English tokens into non-English matrix languages often improves comprehension, especially for models with limited proficiency in the matrix language, indicating a facilitative role for English in such contexts.
- While strategic prompting can help some models, it negatively affects others, highlighting inconsistency in controllability; by contrast, fine-tuning on code-switched data leads to more stable, albeit partial, performance recovery.

2 Related Work

Code-Switching in Language Models. Early multilingual encoder-based models (e.g., mBERT (Devlin et al., 2019), XLM-R (Conneau et al., 2020)), while effective on monolingual tasks, consistently faltered on code-switched inputs (Winata et al., 2021a). This gap spurred specialized methods for mixed-language text, including new architectures and training regimes (Winata et al., 2019; Liu et al., 2020; Winata et al., 2021b). Although existing benchmarks (Khanuja et al., 2020) supported these efforts, research predominantly focused on encoder-centric models (Winata et al., 2019; Tan and Joty, 2021; Zhu et al., 2023). Consequently, decoder-only architectures, now central to state-of-the-art NLP, have received markedly less scrutiny regarding CSW. While some studies probed adversarial code-mixing in autoregressive models (Das et al., 2022), meaningful evaluation of such models requires access to high-quality, linguistically coherent code-switched text. This has motivated growing interest in controlled CSW text generation.

Code-Switched Text Generation. Synthetic code-switched text generation plays a critical role in data augmentation and diversification for multilingual language models (Pratapa et al., 2018; Zhang et al., 2023). Methods range from linguistically motivated approaches—such as the Equivalence Constraint Theory (ECT) (Poplack, 1978) and Matrix Language Frame (MLF) model (Myers-Scotton, 1993)—to heuristic token-level substitutions (Myslín, 2014; and, 2018; Chan et al., 2024). Recent work often relies on word-level aligners to guide borrowing from embedded-language texts while preserving grammatical structure (Kuwanto et al., 2024). Although these techniques aim for token-level accuracy, they overlook the growing capacity of LLMs to perform context-aware, linguistically grounded substitutions. Leveraging this

[1]Links will be provided upon acceptance.

potential, recent studies have explored LLM-based generation using linguistic constraints (Kuwanto et al., 2024), fine-tuning on CSW data (Heredia et al., 2025), or zero-shot prompting (Potter and Yuan, 2024). Still, challenges remain in controlling switch placement, scaling across language pairs, and conducting robust evaluation. Our work addresses these challenges by leveraging modern LLMs to generate linguistically grounded code-switched text, grounded in established theoretical constraints, to support more rigorous evaluation of model comprehension in mixed-language contexts.

Evaluation of LLM CSW Capabilities. LLM code-switching evaluation has largely focused on surface-level tasks through benchmarks like GLUECoS (Khanuja et al., 2020), LINCE (Aguilar et al., 2020), and SemEval (Patwa et al., 2020) (e.g., language ID, sentiment, PoS tagging), thus neglecting deeper semantic or reasoning capabilities. Although more recent studies assess CSW sentiment classification (Winata et al., 2021a), and question answering (Huzaifah et al., 2024), they are limited in scope, emphasizing task-specific metrics over broader comprehension. In contrast, our approach introduces linguistically grounded CSW variants of established comprehension and reasoning tasks, enabling a more rigorous assessment of LLMs' capacity to reason over mixed-language input beyond surface-level performance.

3 Methodology

3.1 Notations

Let

$$\mathcal{B} = \{B_p\}_{p=1}^{P}$$

be a set of P standard benchmarks. Let

$$\mathcal{L} = \{l_j\}_{j=1}^{L}$$

be a set of L languages from which the matrix and embedded languages are selected for code-switched benchmarks generation. Let

$$\mathcal{M} = \{m_k\}_{k=1}^{K}$$

be a set of K LLMs. To evaluate the performance of an LLM $m_k \in \mathcal{M}$ on code-switched text comprehension, we generate a code-switched version of benchmark $B_p \in \mathcal{B}$ using a single matrix language $l_{\text{matrix}} \in \mathcal{L}$ and a set of embedded languages $\mathcal{L}_{\text{embedded}}$, where $\mathcal{L}_{\text{embedded}} \subseteq \mathcal{L} \setminus l_{\text{matrix}}$ and $|\mathcal{L}_{\text{embedded}}| \geq 1$, which we denote by $B_p^{l_{\text{matrix}} \rightarrow \mathcal{L}_{\text{embedded}}}$.

3.2 Code-Switching Methods

To investigate how different code-switching strategies affect LLM comprehension, we generate inputs using two distinct approaches: a linguistically grounded *noun-token* method (Poplack, 1988; Muysken, 2000; Moyer, 2002; Chan et al., 2024) and a heuristic *ratio-token* method (Chan et al., 2024). In the noun-token method, we replace nouns in the matrix language text with their aligned counterparts from a parallel sentence in the embedded language. Substitutions are only applied when they preserve grammatical well-formedness according to two established linguistic constraints: the Equivalence Constraint Theory (ECT), which requires syntactic alignment at switch points, and the Matrix Language Frame (MLF) model, which mandates that the matrix language maintains control over the clause's morpho-syntactic structure. In contrast, the ratio-token method replaces 20% of tokens at random, regardless of linguistic structure. This comparison allows us to isolate the role of syntactic and grammatical constraints in LLM comprehension of code-switched text.

3.3 Code-Switched Text Generation Approaches

Given a corpus of parallel texts, we generate code-switched sentences by substituting embedded-language words into matrix-language sentences using two approaches:

Alignment-Based Approach. We begin by aligning words between matrix and embedded language sentences using the AWESOME aligner (Dou and Neubig, 2021), guided by LaBSE embeddings (Feng et al., 2022). Based on this alignment, we apply two code-switching strategies:
Noun-Token: Matrix-language nouns are identified using the Stanza POS tagger (Qi et al., 2020), then replaced by their aligned counterparts from the embedded-language text guided by Claude 3.5 Sonnet (*Claude*), while ensuring compliance with the Equivalence Constraint Theory (ECT), and the Matrix Language Frame (MLF) model.
Ratio-Token: 20% of aligned tokens are randomly sampled and substituted with embedded-language words, without enforcing any linguistic constraints (Chan et al., 2024).

LLM-Centric Approach Inspired by the recent capabilities of LLMs in code-switched text generation (Potter and Yuan, 2024), we propose a two-step approach using *Claude* to generate CSW text. In step (1), the model identifies and placeholder-masks switching points in the matrix-language sentence—nouns for the noun-token strategy and randomly selected tokens for the ratio-token strategy. In step (2), the placeholders are filled with contextually appropriate words from the embedded-language sentence.

3.4 Code-Switching Approach Evaluation

For each embedded language, we assembled a 300-sample test-set, and generated code-switched variants using both CSW approaches. GPT-4o then conducted blind, pairwise comparisons under the LLM-as-a-Judge framework (Zheng et al., 2023), evaluating fluency, depth of mixing, grammatical validity at switch points, adherence to the Matrix Language Frame model, and overall coherence. In every case, GPT-4o preferred the two-step LLM-Centric approach, demonstrating its superior capacity to produce high-quality, linguistically coherent code-switched text (See Appendix B for details on the embedding model, LLM setup, and CSW approach selection and evaluation).

3.5 Evaluation Metrics

We evaluate models using three key metrics to capture baseline performance and the effects of code-switching: accuracy, weighted average accuracy, and accuracy delta.

Accuracy. For a model $m_k \in \mathcal{M}$ and benchmark B', whether a monolingual test $B_p \in \mathcal{B}$ or its code-switched variant $B_p^{l_{\text{matrix}} \to \mathcal{L}_{\text{embedded}}}$, we define accuracy as:

$$\mathrm{Acc}(m_k, B') = \frac{1}{|B'|} \sum_{i=1}^{|B'|} \mathbb{1}(\mathrm{Correct}(m_k, \mathrm{instance}_i)), \quad (1)$$

where $|B'|$ denotes the number of samples in benchmark B', $\mathrm{instance}_i$ is its i-th example, and $\mathbb{1}(\cdot)$ is the indicator function.

Weighted Average Accuracy. To report an aggregate performance measure for a model m_k across multiple benchmarks $\mathcal{B}$, we compute the weighted average accuracy as:

$$\mathrm{Acc}_{\text{weighted}}(m_k, l_{\text{matrix}}, \mathcal{L}_{\text{embedded}}) = \frac{\sum_{B_p \in \mathcal{B}} |B_p| \cdot \mathrm{Acc}(m_k, B_p^{l_{\text{matrix}} \to \mathcal{L}_{\text{embedded}}})}{\sum_{B_p \in \mathcal{B}} |B_p|}, \quad (2)$$

Accuracy Delta (ΔAcc). We quantify the code-switching impact by computing the accuracy delta, i.e., the difference between a model's score on the code-switched benchmark and its score on the original monolingual benchmark, as:

$$\Delta\mathrm{Acc}(m_k, B_p^{l_{\text{matrix}} \to \mathcal{L}_{\text{embedded}}}) = \mathrm{Acc}(m_k, B_p^{l_{\text{matrix}} \to \mathcal{L}_{\text{embedded}}}) - \mathrm{Acc}(m_k, B_p). \quad (3)$$

Positive ΔAcc indicates an improvement under code-switching, negative values a drop.

4 Experimental Setting

Languages selection We consider a set of languages

$$\mathcal{L} = \{\text{English}, \text{Arabic}, \text{German}, \text{French}, \text{Chinese}\}$$

We hypothesize that this set creates varying degrees of semantic, lexical, and syntactic similarities between the matrix language and the embedded languages set, which may differentially affect the degradation caused by code-switching, akin to effects observed in machine translation (Guerin et al., 2024; Mohamed et al., 2025).

Models selection We evaluated LLaMA 3.2 Instruct (3B) and LLaMA 3.1 Instruct (8B, 70B) (Grattafiori et al., 2024), Qwen 2.5 Instruct (3B, 7B, 72B) (Yang et al., 2025), Mistral 7B Instruct (v0.3) (Albert et al., 2023), and ALLaM 7B (Bari et al., 2024), encompassing a wide range of scales and pretraining curricula. *Allam* currently represents the state-of-the-art in Arabic LLMs, while *Qwen* and *Mistral* excel in Chinese and French, respectively, even as they maintain strong multilingual capabilities. The *Llama* family delivers consistently robust multilingual performance, enabling us to isolate the effects of architecture and model scale on code-switching resilience.

Benchmarks selection We assess LLM comprehension on three established tasks: *Belebele* (Bandarkar et al., 2023) for passage-level reading comprehension (with both passages and questions code-switched), *MMLU*[2] (Hendrycks et al., 2020) for broad-domain multiple-choice reasoning (code-switching applied to questions), and *XNLI* (Conneau et al., 2018) natural language inference (both premise and hypothesis code-switched). To ensure consistent, scalable evaluation across models, we used and adapted EleutherAI's Language Model Evaluation Harness (Gao et al., 2024) for our code-switched variants.

[2]https://huggingface.co/datasets/openai/MMMLU

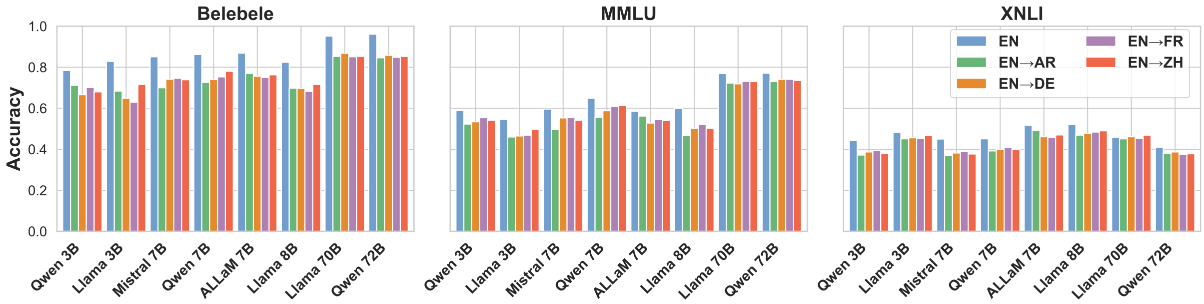

Figure 2: Comparison of LLM accuracy on monolingual English versions of *Belebele*, *MMLU*, and *XNLI* benchmarks (baseline) versus their noun-token code-switched counterparts. English serves as the matrix language, with Arabic (EN→AR), French (EN→FR), German (EN→DE), and Chinese (EN→ZH) as embedded languages.

5 Experiments

5.1 Experiment 1: Linguistically motivated CSW

Setup We use English as the matrix language l_{matrix}, and perform code-switching on the benchmarks with each language in $\mathcal{L} \setminus l_{\text{matrix}}$ as the embedded language separately, using the noun-token code-switching method, and compare the performance of the code-switched benchmarks with the original English benchmarks.

Hypothesis 1 (H1) *We hypothesize that LLM performance on code-switched benchmarks degrades in proportion to the linguistic distance between the matrix and embedded languages.*

Model	EN→AR	EN→DE	EN→FR	EN→ZH	EN
Llama 3B	0.47	0.47	0.47	0.50	**0.54**
Qwen 3B	0.49	0.50	0.52	0.51	**0.56**
Allam 7B	0.55	0.52	0.53	0.53	**0.58**
Mistral 7B	0.47	0.52	0.52	0.51	**0.57**
Qwen 7B	0.52	0.55	0.56	0.57	**0.61**
Llama 8B	0.48	0.51	0.52	0.51	**0.59**
Llama 70B	0.66	0.66	0.67	0.67	**0.70**
Qwen 72B	0.65	0.66	0.65	0.65	**0.69**

Table 1: Weighted average accuracy of selected LLMs on noun-token code-switched benchmarks (EN→ AR, EN→DE, EN→FR, EN→ZH) compared to the monolingual English baseline. Cell colors indicate relative performance from highest (green) to lowest (red). The highest scores are indicated in **bold**.

Results Table 1 and Figure 2 show consistent drops in LLM performance on noun-token code-switched benchmarks compared to their English versions. The extent of degradation varied by embedded language and model. For example, LLaMA-70B's weighted average accuracy declined from 0.70 (English) to 0.66 on EN→AR/EN→DE ($\Delta \approx -0.04$) and 0.67 on EN→ZH ($\Delta \approx -0.03$).

Mistral-7B showed minimal loss on EN→FR ($\Delta \approx -0.01$), and ALLaM-7B retained relatively strong performance on EN→AR ($\Delta \approx -0.06$). Qwen models exhibited consistent degradation across languages (e.g., Qwen-7B: $\Delta \approx -0.03$ to -0.06), with larger models achieving better absolute scores but similar relative drops. These trends held across all three tasks, underscoring both the general difficulty of CSW and the role of language-specific model strengths.

5.2 Experiment 2: Non-linguistically motivated code-switching

Setup In this experiment, we retain the experimental framework of Experiment 1, replacing the linguistically motivated noun-token CSW method with the ratio-token method.

Hypothesis 2 (H2) *We hypothesize that non-linguistically motivated code-switching leads to sharper performance degradation in LLMs than that observed on linguistically motivated code-switching, as such input is less likely to align with patterns encountered during pre-training.*

Results Results are show in Table 2. All models exhibited a decline in weighted average accuracy, consistent with the patterns observed in Experiment 1. The extent of degradation varied with model size and language pairing. Smaller models experienced the most pronounced drops; for example, *Llama 3B* decreased from 0.54 (EN) to 0.43 on EN→DE ($\Delta = -0.11$) and to 0.47 on EN→AR ($\Delta = -0.07$). In contrast, *Llama 70B* showed minimal degradation, with weighted average accuracy decreasing from 0.70 to 0.68 across all embedded languages ($\Delta \approx -0.02$). Language-specific resilience was also observed. *Allam 7B* and *Mistral 7B* relatively strong performance on EN→AR on EN→FR, respectively. *Qwen 7B* exhibited consis-

Model	EN→AR	EN→DE	EN→FR	EN→ZH	EN
Llama 3B	0.47	0.43	0.46	0.51	**0.54**
Qwen 3B	0.50	0.51	0.52	0.51	**0.56**
Allam 7B	0.56	0.51	0.53	0.54	**0.58**
Mistral 7B	0.49	0.52	0.53	0.52	**0.57**
Qwen 7B	0.53	0.55	0.56	0.57	**0.61**
Llama 8B	0.50	0.52	0.53	0.54	**0.59**
Llama 70B	0.68	0.67	0.68	0.68	**0.70**
Qwen 72B	0.66	0.66	0.66	0.66	**0.69**

Table 2: Weighted average accuracy of selected LLMs on ratio-token code-switched benchmarks (EN→ AR, EN→DE, EN→FR, EN→ZH) compared to the monolingual English baseline. Cell colors indicate relative performance from highest (green) to lowest (red). The highest scores are indicated in **bold**.

Model	AR→EN		DE→EN		FR→EN		ZH→EN	
	Orig	CSW	Orig	CSW	Orig	CSW	Orig	CSW
Llama 3B	0.37	**0.45**	0.35	**0.38**	0.43	**0.45**	0.42	**0.47**
Qwen 3B	0.40	**0.48**	0.49	**0.52**	0.50	**0.53**	*0.48*	*0.48*
Allam 7B	0.51	**0.52**	0.39	**0.43**	0.49	**0.52**	0.44	**0.51**
Mistral 7B	0.35	**0.48**	0.50	**0.54**	0.52	**0.55**	0.46	**0.53**
Qwen 7B	0.47	**0.52**	0.51	**0.53**	0.56	**0.57**	**0.56**	0.55
Llama 8B	0.38	**0.44**	*0.50*	*0.50*	0.50	**0.52**	0.49	**0.53**
Llama 70B	0.61	**0.66**	0.67	**0.67**	*0.68*	*0.68*	0.64	**0.66**
Qwen 72B	0.63	**0.66**	*0.68*	*0.68*	*0.68*	*0.68*	*0.66*	*0.66*

Table 3: Weighted average accuracy of LLMs on monolingual (Orig) versus English-embedded code-switched (CSW) benchmarks across Arabic, German, French, and Chinese, rounded to two decimals. **Bold** indicates the higher score in each Orig/CSW pair. *Italic* indicates instances where performance did not change between the original and code-switched versions.

tent, moderate degradation, decreasing from 0.61 to a range of 0.53–0.57 depending on the embedded language ($\Delta = -0.08$ to -0.04).

6 Ablations

Building on Section 5, which found comparable degradation from noun-token and ratio-token code-switching, we proceed with ablation studies using exclusively the noun-token method.

6.1 English as an embedded language

To assess whether embedding English improves comprehension in other matrix languages, we reversed the language roles from the main experiments, using each language in $\mathcal{L} \setminus l_{\text{matrix}}$ as the matrix language, and English as the sole embedded language. We generated code-switched versions ($B_p^{l_{\text{matrix}} \to \{\text{English}\}}$) of the *Belebele*, *MMLU*, and *XNLI* benchmarks. By comparing model performance on these variants against their original monolingual counterparts, we aimed to assess any comprehension enhancement attributable to the embedded English words.

Results are presented in Table 3. Embedding English into lower-resource matrix languages often improved model performance or, at minimum, avoided large degradations. Gains were especially prominent when models lacked proficiency in the matrix language. For instance, *Mistral 7B*'s weighted average accuracy in Arabic rose from 0.35 to 0.48 ($\Delta = +0.13$), while its score in Chinese increased by +0.07 points. In contrast, when models already demonstrated strong matrix language proficiency, improvements were minimal or absent. *Allam 7B* (Arabic) and *Mistral 7B* (French) saw gains of only +0.01 and +0.03, respectively. High-performing models such as *Llama 70B* and *Qwen 72B* showed no change in several settings. Only one case showed a minor drop: *Qwen 7B* on Chinese ($\Delta \approx -0.01$). This suggests that embedded English may introduce interference when matrix language representations are already strong.

6.2 When Code-Switching Goes Extreme

To assess performance under more complex multilingual mixing, an "extreme" code-switching experiment was conducted on the *MMLU* benchmark. English served as the matrix language, with nouns code-switched using three distinct embedded languages sets: **Setting 1** featured a non-Latin script pair ($\mathcal{L}_{\text{embedded}} = \{\text{Arabic,Chinese}\}$), **Setting 2** used a Latin script pair ($\mathcal{L}_{\text{embedded}} = \{\text{French,German}\}$), and **Setting 3** combined all four languages ($\mathcal{L}_{\text{embedded}} = \{\text{Arabic,Chinese,French,German}\}$). For generating the code-switched text across these settings, Claude was, additionally, prompted to borrow words evenly from the specified embedded languages for each instance. Table 4 demonstrates that all models experience a decline in *MMLU* accuracy under extreme code-switching relative to the monolingual English baseline. For example, *Llama 70B*'s score decreases from 0.77 to between 0.70 and 0.72, and *Qwen 72B*'s from 0.77 to 0.73–0.74. Analyzing language-script effects by comparing the non-Latin mix (Setting 1) against the Latin mix (Setting 2) reveals no uniform penalty for non-Latin scripts. *Allam 7B* achieves a higher accuracy with the non-Latin pair (0.56 vs. 0.54), whereas *Mistral 7B* performs better with the Latin pair (0.56 vs. 0.53). Moreover, extending the embedded set to all four languages (Setting 3) does not invariably

Model	Setting 1	Setting 2	Setting 3	EN
Llama 3B	0.48	0.46	0.47	**0.55**
Qwen 3B	0.54	0.55	0.53	**0.59**
Allam 7B	0.56	0.54	0.54	**0.58**
Mistral 7B	0.53	0.56	0.55	**0.59**
Qwen 7B	0.58	0.60	0.59	**0.65**
Llama 8B	0.49	0.51	0.49	**0.60**
Llama 70B	0.72	0.70	0.70	**0.77**
Qwen 72B	0.74	0.74	0.73	**0.77**

Table 4: *MMLU* accuracy for extreme code-switching with l_{matrix} = English and $\mathcal{L}_{\text{embedded}} = \{\text{Arabic}, \text{Chinese}\}$ (Setting 1), $\mathcal{L}_{\text{embedded}} = \{\text{French}, \text{German}\}$ (Setting 2), and $\mathcal{L}_{\text{embedded}} = \{\text{Arabic}, \text{Chinese}, \text{French}, \text{German}\}$ (Setting 3), alongside the monolingual English baseline. The highest scores are indicated in **bold**.

yield the lowest scores, while *Llama 70B* (0.70) and *Qwen 72B* (0.73) record their minima in Setting 3, other models exhibit accuracies intermediate between those in Settings 1 and 2.

7 Mitigation strategies

To mitigate the performance declines induced by code-switching, we investigate two strategies: a prompt-based approach, which prepends explicit instructions to code-switched inputs, and a model-based approach, which fine-tunes LLMs on synthetic CSW data.

7.1 Prompt-based Mitigation

Each noun-token code-switched benchmark instance was prepended with an explicit instruction indicating that the input involves English mixed with an embedded language. Further details on the prompts used per benchmark are provided in Appendix C.

Model	EN→AR	EN→DE	EN→FR	EN→ZH	EN
Llama 3B	0.31	0.34	0.32	0.32	**0.54**
Qwen 3B	0.51	0.53	0.54	0.53	**0.56**
Mistral 7B	0.46	0.50	0.50	0.50	**0.57**
Allam 7B	0.56	0.53	0.54	0.53	**0.58**
Qwen 7B	0.54	0.56	0.58	0.59	**0.61**
Llama 8B	0.41	0.47	0.48	0.47	**0.59**
Llama 70B	0.53	0.53	0.64	0.50	**0.70**
Qwen 72B	0.70	0.71	0.71	**0.72**	0.69

Table 5: Impact of an instructional prompt on LLM weighted average accuracy for noun-token code-switched benchmarks. English serves as the matrix language, with results shown for various embedded languages. The highest scores are indicated in **bold**

The results of the prompt-based mitigation approach, presented in Table 5, show considerable variation across models when compared to unprompted noun-token code-switching (Table 1). For some models, most notably the *Qwen* family, the addition of an explicit instruction led to consistent performance gains. *Qwen 72B* improved across all language pairs, most remarkably surpassing its monolingual English weighted average accuracy (EN→ZH: 0.72 vs. EN: 0.69). Similarly, *Qwen 7B* also benefited, with EN→ZH improving from 0.57 to 0.59 ($\Delta = +0.02$). *Allam 7B* exhibited minor improvements as well, such as EN→AR increasing from 0.55 to 0.56 ($\Delta = +0.01$).

Conversely, for other models, particularly the *Llama* family and *Mistral 7B*, the prompt-based strategy was frequently detrimental. *Llama 8B* saw weighted average accuracy declines across all embedded languages (e.g., EN→FR dropped from 0.52 to 0.48, $\Delta = -0.04$). More substantial drops were observed for *Llama 70B*, especially on EN→AR and EN→ZH, where performance fell by 13 and 17 points respectively. *Llama 3B* and *Mistral 7B* similarly exhibited declines (e.g., *Llama 3B* EN→AR: 0.47 to 0.31, $\Delta = -0.16$).

7.2 Model-based Mitigation

Directly fine-tuning LLMs on code-switched text presents another avenue for mitigation. For this, *Llama 8B* was selected, primarily due to its limited responsiveness to prompting within its size category. A parallel corpus of TED Talk transcripts (Qi et al., 2018) spanning English, Arabic, Chinese, French, and German was utilized. The instruction-tuning dataset was constructed by first selecting samples from the parallel corpus where the English sentence length was greater than 70 words. This filtering yielded approximately 3,650 pairs per language combination. Noun-token code-switching, with English as a matrix language, was then applied to these, resulting in an instruction-tuning dataset of approximately 14,600 training samples. The instruction required the model to generate the code-switched text from the original English and embedded-language sentences, using five distinct prompt templates to ensure instructions diversity (further details in Appendix D).

The impact of this instruction fine-tuning is illustrated in Figure 3. The baseline *Llama 8B* model achieved an English-only weighted average accuracy of 0.59 on the combined benchmarks. Introducing noun-token code-switching without fine-tuning resulted in a weighted average accuracy re-

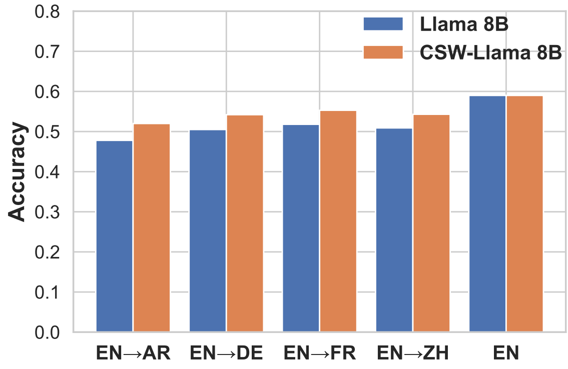

Figure 3: Comparison of *Llama 8B* and its instruction-tuned variant (*CSW-Llama 8B*) on monolingual English benchmarks (*Belebele*, *MMLU*, and *XNLI*) versus their noun-token code-switched counterparts. English serves as the matrix language, with Arabic, French, German, and Chinese, as embedded languages.

duction of up to 0.11 points, depending on the embedded language. After fine-tuning on the code-switched corpus (yielding *CSW-Llama 8B*), a partial recovery of performance was observed. The most significant improvement was for the EN→AR setting, where the weighted average accuracy increased by +0.04 points over the baseline. The smallest gain was for EN→FR, with an increase of +0.03 points.

8 Discussion and Conclusion

As LLMs increasingly process multilingual and mixed-language inputs, understanding their comprehension limits is paramount. This study systematically evaluated LLM performance on code-switched text, yielding multifaceted insights into information processing under these conditions. Our findings reveal several nuanced insights.

LLM comprehension of English as a matrix language is significantly disrupted by the introduction of elements from other languages. Our experiments consistently show that inserting tokens from other languages—Arabic, Chinese, French, or German—into English text leads to a drop in LLM comprehension. This drop does not appear to stem solely from unfamiliarity with code-switching, as similar performance declines were observed when randomly inserting foreign tokens (as in the ratio-token method from Experiment 2). Instead, these findings point to a more fundamental difficulty: LLMs struggle to process disrupted monolingual structures and integrate mixed linguistic signals effectively.

Embedding English tokens into other languages often improves LLM comprehension of the original text. LLMs frequently exhibited improved comprehension on non-English texts when English tokens were embedded, surpassing their baseline performance on the original monolingual versions of the same benchmarks.

Code-switching complexity does not linearly correlate with performance degradation. In our "extreme" code-switching experiments, increasing the number of embedded languages or mixing script types did not consistently lead to greater declines in model performance compared to simpler two-language settings. These findings suggest that degradation is not a direct function of multilingual complexity, but rather emerges from a nuanced interaction between specific language combinations and model-specific linguistic representations.

While prompting helps some models mitigate degradation, fine-tuning offers a more reliable solution. We evaluated two strategies for mitigating the effects of code-switching: prompt-based and model-based. Explicitly prepending instructions about upcoming code-switched input (Table 5) proved effective for some architectures—most notably the *Qwen* family. However, this strategy was less effective, or even detrimental, for others like *Llama* and *Mistral*, likely due to interference with their internal processing. For models that did not benefit from prompting, such as *Llama 8B*, we explored direct instruction fine-tuning on code-switched data. This approach led to a more consistent improvement. As shown in Figure 3, *Llama 8B*, which suffered performance drops under prompting, partially recovered its accuracy after instruction tuning—demonstrating that fine-tuning is a more promising path for improving LLM robustness to code-switching.

These findings underscore that while LLMs exhibit impressive multilingual capabilities, code-switching introduces specific comprehension challenges distinct from monolingual processing. The asymmetric impact of English as a matrix versus embedded language highlights areas requiring further research. While mitigation is possible, the model-specific nature of these solutions points towards the need for more adaptive approaches to ensure reliable LLM performance in real-world multilingual environments.

Limitations

While our study adopts a controlled evaluation setup for both linguistically and non-linguistically motivated code-switching, the noun-token approach we employ reflects one of the fundamental forms of linguistically grounded, naturalistic switching. However, more complex forms of code-switching may induce more severe performance degradation. Future work should investigate how higher-complexity switching patterns affect LLMs' understanding.

Additionally, in our non-linguistically motivated ratio-token experiments, the substitution rate was fixed at 20%. Exploring how variation in this ratio affects model behavior could yield a more nuanced understanding of the impact of non-linguistically grounded switching on LLM comprehension.

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, Marseille, France. European Language Resources Association.

Q Jiang Albert, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, and Devendra Singh Chaplot. 2023. Mistral 7b. *arXiv*.

Li Nguyen and. 2018. Borrowing or code-switching? traces of community norms in vietnamese-english speech. *Australian Journal of Linguistics*, 38(4):443–466.

Lucas Bandarkar, Davis Liang, Benjamin Muller, Mikel Artetxe, Satya Narayan Shukla, Donald Husa, Naman Goyal, Abhinandan Krishnan, Luke Zettlemoyer, and Madian Khabsa. 2023. The belebele benchmark: a parallel reading comprehension dataset in 122 language variants. *arXiv preprint arXiv:2308.16884*.

M Saiful Bari, Yazeed Alnumay, Norah A Alzahrani, Nouf M Alotaibi, Hisham A Alyahya, Sultan AlRashed, Faisal A Mirza, Shaykhah Z Alsubaie, Hassan A Alahmed, Ghadah Alabduljabbar, et al. 2024. Allam: Large language models for arabic and english. *arXiv preprint arXiv:2407.15390*.

Barbara E. Bullock and Almeida Jacqueline Toribio. 2009. *The Cambridge Handbook of Linguistic Code-switching*. Cambridge Handbooks in Language and Linguistics. Cambridge University Press.

Kelvin Wey Han Chan, Christopher Bryant, Li Nguyen, Andrew Caines, and Zheng Yuan. 2024. Grammatical error correction for code-switched sentences by learners of English. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 7926–7938, Torino, Italia. ELRA and ICCL.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Richeek Das, Sahasra Ranjan, Shreya Pathak, and Preethi Jyothi. 2023. Improving pretraining techniques for code-switched NLP. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1176–1191, Toronto, Canada. Association for Computational Linguistics.

Sourya Dipta Das, Ayan Basak, Soumil Mandal, and Dipankar Das. 2022. Advcodemix: Adversarial attack on code-mixed data. In *Proceedings of the 5th Joint International Conference on Data Science & Management of Data (9th ACM IKDD CODS and 27th COMAD)*, pages 125–129.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

A. Seza Doğruöz, Sunayana Sitaram, Barbara E. Bullock, and Almeida Jacqueline Toribio. 2021. A survey of code-switching: Linguistic and social perspectives for language technologies. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1654–1666, Online. Association for Computational Linguistics.

Zi-Yi Dou and Graham Neubig. 2021. Word alignment by fine-tuning embeddings on parallel corpora. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2112–2128, Online. Association for Computational Linguistics.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2024. The language model evaluation harness.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Nicolas Guerin, Shane Steinert-Threlkeld, and Emmanuel Chemla. 2024. The impact of syntactic and semantic proximity on machine translation with back-translation. *arXiv preprint arXiv:2403.18031*.

Ayushman Gupta, Akhil Bhogal, and Kripabandhu Ghosh. 2024. Code-mixer ya nahi: Novel approaches to measuring multilingual llms' code-mixing capabilities. *arXiv preprint arXiv:2410.11079*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Maite Heredia, Gorka Labaka, Jeremy Barnes, and Aitor Soroa. 2025. Conditioning llms to generate code-switched text: A methodology grounded in naturally occurring data. *arXiv preprint arXiv:2502.12924*.

Muhammad Huzaifah, Weihua Zheng, Nattapol Chanpaisit, and Kui Wu. 2024. Evaluating code-switching translation with large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 6381–6394, Torino, Italia. ELRA and ICCL.

Pranjal Khanuja et al. 2020. Improving code-switched nlp using data augmentation. In *Proceedings of ACL 2020*, pages 1860–1871.

Prashant Kodali, Anmol Goel, Likhith Asapu, Vamshi Krishna Bonagiri, Anirudh Govil, Monojit Choudhury, Manish Shrivastava, and Ponnurangam Kumaraguru. 2024. From human judgements to predictive models: Unravelling acceptability in code-mixed sentences. *arXiv preprint arXiv:2405.05572*.

Garry Kuwanto, Chaitanya Agarwal, Genta Indra Winata, and Derry Tanti Wijaya. 2024. Linguistics theory meets llm: Code-switched text generation via equivalence constrained large language models. *arXiv preprint arXiv:2410.22660*.

Zihan Liu, Genta Indra Winata, Zhaojiang Lin, Peng Xu, and Pascale Fung. 2020. Attention-informed mixed-language training for zero-shot cross-lingual task-oriented dialogue systems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8433–8440.

Amr Mohamed, Mingmeng Geng, Michalis Vazirgiannis, and Guokan Shang. 2025. Llm as a broken telephone: Iterative generation distorts information. *arXiv preprint arXiv:2502.20258*.

Melissa G. Moyer. 2002. Pieter muysken, bilingual speech: A typology of code-mixing. cambridge: Cambridge university press, 2000. pp. xvi, 306. hb 59.95. *Language in Society*, 31(4):621–624.

P. Muysken. 2000. *Bilingual Speech: A Typology of Code-Mixing*. Cambridge University Press.

R. Myers-Scotton. 1993. *Social Motivations for Code-Switching: Evidence from Africa*. Oxford University Press.

Mark Myslín. 2014. Codeswitching and predictability of meaning in discourse. In *Codeswitching and predictability of meaning in discourse*.

Lynnette Hui Xian Ng and Luo Qi Chan. 2024. What talking you?: Translating code-mixed messaging texts to english. *arXiv preprint arXiv:2411.05253*.

Millicent Ochieng, Varun Gumma, Sunayana Sitaram, Jindong Wang, Vishrav Chaudhary, Keshet Ronen, Kalika Bali, and Jacki O'Neill. 2024. Beyond metrics: evaluating llms' effectiveness in culturally nuanced, low-resource real-world scenarios. *arXiv preprint arXiv:2406.00343*.

Tanmay Parekh, Emily Ahn, Yulia Tsvetkov, and Alan W Black. 2020. Understanding linguistic accommodation in code-switched human-machine dialogues. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 565–577, Online. Association for Computational Linguistics.

Parth Patwa, Gustavo Aguilar, Sudipta Kar, Suraj Pandey, Srinivas PYKL, Björn Gambäck, Tanmoy Chakraborty, Thamar Solorio, and Amitava Das. 2020. SemEval-2020 task 9: Overview of sentiment analysis of code-mixed tweets. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 774–790, Barcelona (online). International Committee for Computational Linguistics.

Shana Poplack. 1988. *8. Contrasting patterns of codeswitching in two communities*, pages 215–244. De Gruyter Mouton, Berlin, New York.

Susan Poplack. 1978. Sometimes i'll start a sentence in spanish y termino en español: Toward a typology of code-switching. *Linguistics*, 16(7-8):581–618.

Tom Potter and Zheng Yuan. 2024. LLM-based code-switched text generation for grammatical error correction. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16957–16965, Miami, Florida, USA. Association for Computational Linguistics.

Adithya Pratapa, Gayatri Bhat, Monojit Choudhury, Sunayana Sitaram, Sandipan Dandapat, and Kalika Bali. 2018. Language modeling for code-mixing: The role of linguistic theory based synthetic data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1543–1553, Melbourne, Australia. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

Ye Qi, Devendra Sachan, Matthieu Felix, Sarguna Padmanabhan, and Graham Neubig. 2018. When and why are pre-trained word embeddings useful for neural machine translation? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 529–535, New Orleans, Louisiana. Association for Computational Linguistics.

Samson Tan and Shafiq Joty. 2021. Code-mixing on sesame street: Dawn of the adversarial polyglots. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3596–3616, Online. Association for Computational Linguistics.

Genta Winata et al. 2021a. Multilingual pretrained models are effective for code-switching nlp. In *Proceedings of EMNLP 2021*, pages 2345–2356.

Genta Indra Winata, Andrea Madotto, Zhaojiang Lin, Rosanne Liu, Jason Yosinski, and Pascale Fung. 2021b. Language models are few-shot multilingual learners. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 1–15, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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, Hong Kong, China. Association for Computational Linguistics.

Anjali Yadav, Tanya Garg, Matej Klemen, Matej Ulcar, Basant Agarwal, and Marko Robnik Sikonja. 2024. Code-mixed sentiment and hate-speech prediction. *arXiv preprint arXiv:2405.12929*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Ruochen Zhang, Samuel Cahyawijaya, Jan Christian Blaise Cruz, Genta Winata, and Alham Fikri 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, Singapore. Association for Computational Linguistics.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2).

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

Zhihong Zhu, Xuxin Cheng, Zhiqi Huang, Dongsheng Chen, and Yuexian Zou. 2023. Enhancing code-switching for cross-lingual SLU: A unified view of semantic and grammatical coherence. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7849–7856, Singapore. Association for Computational Linguistics.

A Additional Details

All experiments were conducted using NVIDIA A100 (40GB VRAM) and A10 (24GB VRAM) GPU clusters. The compute allocation totaled 22 GPU-days, comprising 8 GPU-days on 8×A100 nodes and 14 GPU-days on 4×A10 nodes.

B Code-Switched Text Generation Approaches and Component Selection

This section details our selection process for model components used in generating code-switched (CSW) text, as introduced in Section 3. Our objective was to identify the most effective LLM and alignment backbone for producing fluent, grammatically valid CSW outputs suitable for benchmark construction.

B.1 LLM Selection for Generation

We compared Claude 3.5 Sonnet and GPT-4o as generation modules for both the Alignment-Based and LLM-Centric pipelines. For each matrix–embedded language pair (EN→AR, ZH, FR, DE), we sampled 100 samples from the *Belebele*, *MMLU*, and *XNLI* benchmarks. Both models generated noun-token CSW sentences under linguistically grounded prompting that adhered to the Equivalence Constraint Theory (ECT) and Matrix Language Frame (MLF) model.

Bilingual annotators conducted pairwise preference evaluations of the outputs, focusing on a single criterion: which code-switched sentence sounded more natural to them. Claude was consistently favored, with preference rates ranging from 52% to 62% across languages, as shown in Table 6. Accordingly, Claude was selected as the generation model for all subsequent CSW construction.

Embedded Language	Claude (%)	GPT-4o (%)
Arabic	55	45
Chinese	57	43
French	52	48
German	62	38

Table 6: Human preferences for CSW text generated by Claude vs. GPT-4o (100 examples per language pair).

B.2 Embedding Backbone Selection

To identify the best embedding model for alignment in the Alignment-Based Pipeline, we evaluated AWESOME with mBERT (AWESOME's default embedding model) and LaBSE. For each language pair, 300 noun-token CSW sentences were generated using alignments from each configuration, with substitution handled by Claude.

Using GPT-4o as an LLM-based judge, we found that LaBSE-based alignments consistently yielded more natural and fluent code-switched outputs than those derived from mBERT, with clear preferences observed for Arabic (89.0%), Chinese (91.3%), and French (74.7%). For German, the preference was more modest (55.3%), though still in favor of LaBSE. GPT-4o was selected as the evaluator due to its strong multilingual capabilities and demonstrated aptitude in code-switching understanding across typologically diverse languages. Importantly, using GPT-4o rather than Claude to evaluate outputs avoids the potential biases introduced by self-evaluation, such as output familiarity or training data memorization, thus providing a more neutral and reliable assessment of generation quality. Results presented in Table 7, informed our decision to adopt LaBSE as the default embedding backbone for alignment in all subsequent experiments.

Embedded Language	LaBSE (%)	mBERT (%)
Arabic	89.0	11.0
Chinese	91.3	8.7
French	74.7	25.3
German	55.3	44.7

Table 7: GPT-4o preference rates for CSW text generated using LaBSE vs. mBERT alignments. Percentages reflect outcome ratios from 300 evaluation instances per language.

B.3 Final Generation Approach Selection

We compared the Alignment-Based Pipeline and the LLM-Centric Method for generating noun-token CSW text across 100 samples per language and benchmark. Results are presented in Table 8. Pairwise evaluation via GPT-4o favored the LLM-Centric approach in all settings, with the strongest preferences for Chinese (66%) and French (63.8%). Based on these results, we adopt the LLM-Centric Method for all noun-token CSW benchmark construction, while retaining the Alignment-Based Pipeline for tasks requiring explicit control over substitution rates (e.g., ratio-token generation).

Embedded Language	LLM-Centric (%)	Alignment-Based (%)
Arabic	56.1	43.9
Chinese	66.0	34.0
French	63.8	36.2
German	53.4	46.6

Table 8: GPT-4o preferences between generation methods for noun-token CSW outputs.

```
You have two code-switched sentences, A and B, each blending English (matrix
    language) with {second_language}. Follow these steps and then choose the better
    sentence (A or B):

1. Assess Fluency: check which sentence flows most naturally, like plausible
    bilingual speech.
2. Assess Depth of Mixing: check which sentence meaningfully integrates both
    languages rather than inserting isolated tokens.
3. Assess Switch Grammar: check which sentence has grammatically valid switch points
     under Equivalence Constraint Theory.
4. Assess Consistency: check which sentence uses English as its grammatical frame
    and embeds {second_language} elements appropriately under the Matrix Language
    Frame model.
5. Assess Overall Coherence: check which sentence remains clear and plausible as a
    whole despite the language mixing.

After evaluating all five criteria, return A or B with no further explanation.

Sentences:
A: {sentence_one}
B: {sentence_two}

Output:
```

Figure 4: The prompt given to Claude 3.5 Sonnet for choosing the best summary between the baseline and LLM-generated summaries.

C Instructional Prompt for Prompt-Based Mitigation

Belebele Prompt

```
You are an expert in understanding code-switched text. You will be given a passage
    and a question in code-switched English and Arabic. You have to understand them
    and respond to the given question with best answer: A, B, C, or D.
```

Figure 5: Instructional prompt prepended for *Belebele* multiple-choice QA tasks.

MMLU Prompt

```
You are an expert in understanding code-switched text. You will be given a question
    in code-switched English and Arabic. You have to understand it and respond to
    the given question with best answer: A, B, C, or D.
```

Figure 6: Instructional prompt prepended for *MMLU* multiple-choice QA tasks.

XNLI Prompt

```
You are an expert in understanding code-switched text. You will be given two code-
    switched passages that correspond to a premise and a hypothesis in code-switched
     English and Arabic text. You have to understand them and respond with the best
    answer: 0, 1, or 2.
```

Figure 7: Instructional prompt prepended for *XNLI* natural language inference tasks.

D Instruction Tuning for Model-Based Mitigation

We fine-tuned *LLaMA-3.1-8B-Instruct* to improve its comprehension of code-switched text using a targeted instruction-tuning dataset. Full-model training was conducted over a single epoch using a learning rate of 2×10^{-6} with linear decay and 5% warmup. Training leveraged mixed-precision BF16 and dynamic sequence packing within a 4096-token window, and a batch-size of four.

D.1 Dataset Preparation

The training data was derived from parallel TED Talk translations (Qi et al., 2018), selecting English sentences longer than 70 words and their Arabic, Chinese, French, and German equivalents. Each English sentence was converted into four code-switched variants using the LLM-Centric Method (Appendix B.3). The final dataset included over 14,000 examples, shuffled and formatted as instruction–response pairs.

D.2 Prompt Templates for Instruction Tuning

To prevent overfitting to fixed phrasing, each training instance was paired with a randomly selected prompt from a pool of five semantically equivalent instruction templates. These templates varied in their surface structure but uniformly instructed the model to blend the matrix English sentence with embedded nouns from the translation. Figures 8–12 illustrate the five styles used.

```
Take this English sentence and infuse it with <LANGUAGE> code-switching:
English: "<ENGLISH_SENTENCE>"
<LANGUAGE>: "<TRANSLATION_SENTENCE>"
```

Figure 8: Infusion-style template.

```
Convert the following English line into a code-switched mix with <LANGUAGE>:
English: "<ENGLISH_SENTENCE>"
<LANGUAGE>: "<TRANSLATION_SENTENCE>"
```

Figure 9: Conversion-style template.

```
Blend English and <LANGUAGE> in the sentence below:
English text: "<ENGLISH_SENTENCE>"
<LANGUAGE> equivalent: "<TRANSLATION_SENTENCE>"
```

Figure 10: Blending-style template.

```
Generate a code-switched rendition by swapping in <LANGUAGE>:
English original: "<ENGLISH_SENTENCE>"
<LANGUAGE> snippet: "<TRANSLATION_SENTENCE>"
```

Figure 11: Rendition-style template.

```
Switch parts of this English sentence into <LANGUAGE>:
English: "<ENGLISH_SENTENCE>"
<LANGUAGE>: "<TRANSLATION_SENTENCE>"
```

Figure 12: Switching-style template.