Conditioning LLMs to Generate Code-Switched Text

Maite Heredia Gorka Labaka Jeremy Barnes Aitor Soroa
HiTZ Center - Ixa, University of the Basque Country UPV/EHU
maite.heredia@ehu.eus

Abstract

Code-switching (CS) is still a critical challenge in Natural Language Processing (NLP), due to the limited availability of large-scale, diverse CS datasets for robust training and evaluation. Despite recent advances, the capabilities and limitations of LLMs in handling CS are still not fully understood. In this work, we investigate the extent to which LLMs can be used in a framework for CS text generation, focusing on the English-Spanish language pair. Our proposed methodology consists of back-translating natural CS sentences into monolingual English, and using the resulting parallel corpus to fine-tune LLMs to turn monolingual sentences into CS. We thoroughly analyze the models' performance through a study on human preferences, a qualitative error analysis, an evaluation with popular reference-based metrics and LLM-based judgment. Results show that fine-tuning can be a key step to ensure that current LLMs consistently generate fluent code-switched text and that our methodology generates high-quality outputs, expanding research opportunities in CS communication. We find that traditional metrics do not correlate with human judgment, and although LLM-based evaluation aligns somewhat more closely, the agreement remains limited. We release our code and generated dataset under a CC-BY-NC-SA license.[1]

Keywords: Corpus (Creation, Annotation, etc.); Evaluation Methodologies; Multilinguality; Natural Language Generation; Code-switching

1. Introduction

Code-Switching (CS) consists of mixing two or more languages within a single utterance and is a common phenomenon in multilingual settings (Tucker, 2001). Although it is mainly present in spoken interactions, it can also be found in written interactions on-line (Appel and Muysken, 2005; Sarkisov, 2021), where it appears jointly with other features of informal speech. Example 1 shows an utterance where the speaker switches between English and Spanish.

(1) Why make everybody *sentarse atrás pa' que* everybody has to move *pa' que se salga*.
Why make everybody *sit at the back so that* everybody has to move *so that she may get out.*[2]
(Poplack, 1980)

Despite the prevalence of code-switching, most research in Natural Language Processing (NLP) assumes monolingualism as a standard for human communication. However, this implicit decision means that state-of-the-art models are not able to properly interpret or generate CS data. Even advances in *multilingual* language modelling (Lin et al., 2022; Chowdhery et al., 2023) have not led to significant improvements, and performance on CS data is still poor compared to performance on monolingual data (Aguilar et al., 2020; Winata et al., 2021). This occurs because there is little CS text available in the multilingual pretraining data. Similarly, there are no parallel datasets available to learn to generate CS in a supervised fashion, as one would expect for tasks such as Machine Translation (MT). Finally, existing methodology for evaluating automatically generated CS text, which has specific needs different from other text generation tasks, are still not good enough and fail to capture nuances of CS text (Srivastava and Singh, 2021). It is therefore crucial to develop methodologies to enable models to generate natural CS text and simultaneously implement robust evaluation frameworks that can assess how well NLP systems handle CS across multiple tasks. We argue that both of these goals require models that can conditionally generate CS from monolingual text. Consequently, our research focuses on the development of a methodology to fine-tune and evaluate LLMs on the task of CS generation, following three main research questions:

RQ1: What are the comparative strengths and limitations of fine-tuned versus non-fine-tuned LLMs in generating fluent and natural code-switched text?

RQ2: How can we leverage LLMs to create high-quality pseudo-parallel data for fine-tuning LLMs in CS text generation?

RQ3: Do automatic metrics for Natural Language Generation (NLG) or LLM judges correlate well with human judgment for the task of CS generation?

[1] Code Dataset

[2] In all examples of CS featured in this paper, Spanish parts are shown in italics, in both the original instance and its translation.

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 8937–8953
11–16 May 2026. ©ELRA Language Resources Association (ELRA), 2026

Based on these research questions, we propose a novel approach to generate CS from monolingual text using LLMs and apply it to the English-Spanish pair. We create a new parallel English–CS corpus, *EN-CS*, by leveraging natural CS data and using LLMs to perform back-translation from CS into English, resulting in high-quality pseudo-parallel pairs, suitable for training and evaluating models on CS generation (RQ2). We provide a comprehensive comparison of CS generation using LLMs in both zero-shot and fine-tuned settings, and we compare their performance against that of a dedicated MT model (RQ1). Finally, we evaluate our methodology both qualitatively, with a study on human preferences and a manual error analysis, and quantitatively, using automatic NLG metrics and LLM as a judge, which allows us to study the correlation between human and automatic evaluation for this task (RQ3). The evaluation is conducted in both in-domain and out-of-domain settings.

2. Related Work

Perspectives in linguistics. CS naturally occurs in communities where two or more languages are in contact, making it a subject of interest to fields like sociolinguistics and psycholinguistics. From a social perspective, it can be affected by speakers' attitudes towards the languages and the CS phenomenon itself. In this respect, it is related to notions of prestige and identity (Heredia et al., 2025). For example, in bilingual communities where a language is minoritized, CS can be seen as an intrusion of the majority language (Dewaele and Wei, 2014). However, for migrant communities, it may be a way to preserve their mother tongue and as an "emblem of ethnic identity" (Poplack, 1980). Its importance in different social contexts highlights the need to consider CS in NLP research, as it plays a crucial role in linguistic interactions and, consequently, the development of language technologies.

Datasets & benchmarks for CS. Most code-switched data stems from social media, while other popular data sources include recordings and transcriptions (Winata et al., 2023). Shared tasks using such CS data have been organized for the tasks of Language Identification (Solorio et al., 2014; Molina et al., 2016) and Sentiment Analysis (Patwa et al., 2020). Similarly, two benchmarks exist to evaluate model performance on CS text, covering different language pairs and tasks: LINCE (Aguilar et al., 2020), which covers tasks such as Part Of Speech tagging or Sentiment Analysis; and GLUECoS (Khanuja et al., 2020), which focuses on NLU tasks for Hindi-English. GLUECoS cannot be currently used without access to the X API.

	Train	Dev	Test	Test (ood)
Original	94,728	19,574	33,361	352
Pre-processed	12,933	2,461	5,353	254
EN-CS	10,703	791	1,040	171

Table 1: Size of original LINCE (EN-ES) compared to the automatically filtered instances and the final set of parallel instances, dubbed *EN-CS*.

CS generation. CS generation has seldom been tackled in previous research. Approaches include linguistically informed techniques to find plausible switching points (Pratapa et al., 2018; Gupta et al., 2020; Gregorius and Okadome, 2022; Hsu et al., 2023; Potter and Yuan, 2024), data augmentation techniques (Tarunesh et al., 2021) and, more recently, prompting LLMs for CS generation (Yong et al., 2023; Terblanche et al., 2024). While CS generation is often evaluated by human annotators (Tarunesh et al., 2021; Gregorius and Okadome, 2022), there remains a need for more robust automatic evaluation methodologies to carefully assess the naturalness and fluency of the generated texts, as recently explored by JudgeLLMs (Kuwanto et al., 2024).

3. Parallel Data Creation

In this work we present a novel approach to generate code-switched text from monolingual sentences. As a first step, we create a synthetic parallel corpus from an initial set of English-Spanish CS sentences from the LINCE benchmark (Aguilar et al., 2020) with their English monolingual equivalents, generated by the Command R model (Cohere For AI, 2024). We exploit the fact that LLMs struggle to generate CS text given a monolingual sentence (c.f. Section 5), but are able to more reliably convert a CS sentence to its corresponding monolingual version, especially when the target language is English. After having created this pseudo-parallel corpus, we use it to fine-tune LLMs on the task of conditional code-switching generation, presented in Section 4.

3.1. The LINCE benchmark

We use LINCE as a starting point, a popular benchmark that has been widely used to evaluate CS systems (Aguilar et al., 2020), which is available in 6 language pairs. All sentences in LINCE are tokenized, and each token is annotated with a language tag as well as other categories depending on the task. In our work we focus on the English-Spanish pair and filter all sentences in the data that do not contain CS, similarly discarding all the task-specific annotations. Example 2 shows a random instance from LINCE.

	Original	English
Silver	you just have to tell me *que como te va*.	You just have to tell me *how it's going*.
	osea i know we wanna party *pero tampoco no aya asta dallas*	like i know we want to party *but not all the way to dallas*
Gold	*hasta venir a plaza se siente* like home.	*even coming to the square feels* like home.
	me siento tan pendejo right now.	*i feel so stupid* right now.

Table 2: Examples of the *EN-CS* parallel corpus. Left: original code-switched instances, right: generated (silver) or post-edited (gold) English instances.

(2)

estaba	aquí	three	feet	away	.
spa	spa	eng	eng	eng	eng&spa

LINCE comprises around $95,000$ train, $20,000$ development, and $33,000$ test instances for the English-Spanish pair. We deduplicate the instances among splits, and filter and pre-process the instances to ensure that they are suitable for our task by removing links, replacing usernames with the placeholder <*user*>, and detokenizing all instances with the script provided as part of the Moses toolkit (Koehn et al., 2007). After this preprocessing, we obtain a more natural version of the LINCE data. A preliminary analysis reveals that many sentences in LINCE are monolingual or contain a single word in one language, which often corresponds to a borrowing, as shown in Example 3. We adopt a simple heuristic to approximate the distinction between actual CS and cases that involve only a borrowing or are incorrectly labeled. Specifically, we retain only sentences that contain at least two words in each language, which substantially reduces the likelihood of including cases where an isolated borrowing is tagged as CS.

(3) I need a shot of tequila or a glass of scotch to keep me warm right now.

After these pre-processing and filtering steps, we end up with $12,933$ train, $2,461$ development and $5,353$ test instances. The comparison between the original size of LINCE and the final number of sentences selected for our experiments after preprocessing is shown in Table 1.

3.2. *EN-CS*

The next step in our method requires creating a pseudo-parallel English-CS dataset by translating the natural code-switched instances into monolingual text. As there are no available MT systems to convert from English-Spanish CS text to English monolingual text, we instead make use of prompt engineering, using the Command R model (Cohere For AI, 2024), one of the strongest publicly available models at the time.

We perform an initial set of experiments to determine the optimal prompt to generate monolingual English versions of the code-switched data. Ideally, we aim for a prompt that generates translations that maintain the meaning of the original sentences, are fluent and natural, whose grammar is correct, and that do not contain any Spanish words or phrases. After extensive testing (see Appendix A), we use the following prompt in a 5-shot setting: *Now convert this code-switched phrase to English. Leave the parts in English as they are, focus on translating the parts in Spanish.* Finally, we filter output instances that contain profanity that was not present in the source texts or irrelevant information, such as *Of course, here's your translation:*, because preliminary experiments show that these instances were problematic for the generation task and conditioned the outputs too much.

In order to create a valid gold standard test set, we perform a manual post-edition of the the monolingual test translations for $1,040$ instances of the LINCE test set. The post-edition was carried out by three proficient speakers of English and Spanish, who were provided with specific guidelines, as shown in Appendix B.

Table 1 shows the final size of the parallel corpus, which we dub *EN-CS*, after post-processing and post-edition, and Table 2 shows examples of silver and gold instances. The final version of our dataset therefore contains $10,703$ train and 791 development instances with automatically translated English sentences matched to their original CS sentences, and $1,040$ gold instances with post-edited English translations.

3.2.1. Quality assessment

We evaluate the quality of the automatic translations (train/dev) by measuring two dimensions: the overall *fluency* of the sentences and *adequacy* of the translations in respect to the source texts. Two fluent English-speaking annotators evaluate the same 100 random instances using a 5-point Likert scale (Callison-Burch et al., 2007) and obtain 4.6 (fluency) and 4.5 (adequacy) points on average, which show the quality of the generated translations. A quadratic Cohen's κ of 0.57 indicates moderate agreement, likely due to lower Likert scores (1, 2, and 3) being rarely selected by the annotators, which is a known problem for κ (Xu and Lorber, 2014; Barnes et al., 2025). In fact, the raw agreement between annotators is substantially higher:

Model	Generated Output
Original (Gold)	damm *todos se casaron* and we still single lol forever alone
English (Source)	damn everyone got married and we're still single lol forever alone
Llama3	damn *todos se fueron a casarse y nosotras estamos solitarias* lol forever alone
Llama3 Instruct	damm every1 got married and we're still single lol *alonso solit@o* foreverrrr lolololo
Llama3.3-70B$_{\text{fs}}$	damn *todo el mundo se casó y nosotros seguimos solteros* lol forever alone
GPT-4o$_{\text{fs}}$	damn *todos se casaron y nosotros seguimos solteros* lol forever alone
NLLB	damn everyone got marry and its still single lol forever alone

Table 3: Example from the test set and the generated outputs of the different models.

0.71 for fluency and 0.65 for adequacy. See Appendix C for further details.

To estimate the quality of the post-edition process, we compare the post-editions of two annotators on 100 additional random instances. The results show that 75% of the sentences remain unchanged, as they are already adequate. There is a 87.87% similarity between the post-editions of the two annotators, as measured by Levenshtein distance, demonstrating a high degree of consistency and quality in the post-edition process.

4. Experimental settings

Generation settings With *EN-CS* as our starting point, we frame CS generation as a MT task, with English as the source and CS as the target language, where parts of the source sentence have to be translated to Spanish. In our experiments, we fine-tune two small-sized generative models from the Llama family, namely, Llama3 8B and Llama3 Instruct 8B (Dubey et al., 2024).

To fine-tune the models, we use the causal language modelling objective, but with appropriate input formats for the base and instruct models. For the base model, we use templates (Zhu et al., 2024) in the form of "<X>=<Y>", where <X> and <Y> are placeholders for the input English sentence and generated CS, respectively. At inference, the second code-switched part is left empty for the model to fill. For the instruction-tuned model, we provide a system prompt with the instruction, a query by the user in English, and an answer from the assistant with the code-switched target. At inference time, the answer is left blank (See Appendix D for example prompts).

Models are trained using Quantized Low-Rank Adaptation (QLoRA) (Dettmers et al., 2023) with standard parameters: the model is loaded in 4 bit with NF4 quantization data type and bf16 computational data type. The LoRA rank and scaling factor are set to 16 and the dropout to 0.05. We apply the LoRA update matrices to the attention blocks and do not train bias parameters. Regarding the hyperparameters, we only tune the learning rate ($1e^{-4}$, $5e^{-4}$, $1e^{-3}$ and $5e^{-3}$) and training epoch $\in [1 \ldots 10]$, choosing the parameters that give the lowest cross-entropy loss on the development set for each model. We use the transformers package (Wolf et al., 2020) for all training experiments.

Early experiments indicated that fine-tuned models usually produce the desired output up to a punctuation mark and then either begin to translate the sentence again or hallucinate more content. We therefore truncate the output up to a punctuation mark where the length is closest to that of the original sentence (Bawden and Yvon, 2023). Although simple, this has proven to be the method that yields the best results. We additionally experimented with different generation parameters (*length_penalty* and *exponential_length_decay*), as well as trying to control the length of the generation with length codes, but find that the truncation heuristic performs the best. Accordingly, all further experiments will use the truncated output.

In-domain and out-of-domain evaluation To evaluate the performance of models, we test them on the test set of *EN-CS*, and to evaluate their capabilities outside of the domains covered by LINCE, we also propose an out-of-domain evaluation. For this purpose, we gather a series of small creative non-fiction texts originally written using English-Spanish CS (Dwyer, 2017, 2019a,b,c,d, 2020). These are quite different from the other split of the test set in domain and register, as well as more superficial features such as the length of the sentences, that tend to be much longer than those in LINCE. To process this collection of texts in a similar manner to the instances of LINCE, we first divide them into sentences and then obtain the language ID for each token using the models from the CodeSwitch repository.[3] With each sentence tokenized and tagged with its corresponding language, we can then process and filter the instances just like those sourced from LINCE (c.f. Section 3.1).

[3] https://github.com/sagorbrur/codeswitch

Baselines We include few-shot experiments by directly prompting GPT-4o and Llama3.3 70B Instruct to generate CS text using a 5-shot approach (See Appendix D for example prompts). We refer to these systems in the experiments as GPT-4o$_{\mathrm{fs}}$ and Llama3.3-70B$_{\mathrm{fs}}$, respectively. We also include a strong dedicated MT baseline, developed by fine-tuning the NLLB (NLLB Team et al., 2022) model (`nllb-200-distilled-600M`) using *EN-CS*. The model was trained with standard settings.[4]

Table 3 shows an example of the outputs of the different models, compared to the original code-switched sentence, and the English monolingual sentence that they received as input.

5. Qualitative evaluation

As a first step to assess the quality of the outputs produced by the different models, we perform a manual qualitative analysis of the results in two parts: a pairwise tournament-based human evaluation, and an in-depth analysis of the most common errors made by the models and their distribution.

5.1. Preference based evaluation

We perform a tournament-based evaluation that allows us to determine the ranking of models in terms of human preference. A total of 1260 instances are matched against each other, corresponding to the outputs of the five models for 210 English source sentences, as well as the gold standard reference. The evaluation is conducted pairwise, requiring annotators to choose the best out of two sentences or declare a tie. When choosing the best sentence, annotators do not know the original English sentence, nor which model produced what output. This process results in $210 \cdot \binom{6}{2} = 3150$ comparisons, and was carried out by 14 annotators, with each annotator performing at least 100 random comparisons.

Annotators are provided with a series of criteria to choose between the instances, based on the error analysis described in the next section. They must take into account three main criteria, which must be applied in the following order: a) the presence and naturalness of the CS; b) the content and fluency of the sentences; and c) the orthographical errors of the instances (correct punctuation, presence of typos, etc.). Annotators are furthermore asked to avoid declaring ties, unless completely necessary (e.g., in a case where both sentences are completely monolingual and therefore equally incorrect) to compel them to develop a preference. The complete annotation guidelines are available in Appendix E. Inter-annotator agreement on a subset of 200 sentence pairs shows substantial agreement for the in-domain set ($\kappa = 0.61$) and moderate agreement for the out-of-domain set ($\kappa = 0.51$).

Model	In domain Score	In domain Rank	Out of domain Score	Out of domain Rank	Total Score	Total Rank
Gold Standard	369.5	🥇	434.5	🥇	804.0	🥇
Llama3	291.5	🥈	282.0	🥈	573.5	🥈
Llama3 Instruct	270.5	🥉	210.0	4	480.5	4
NLLB	259.5	4	247.5	🥉	507	🥉
GPT-4o$_{\mathrm{fs}}$	251.5	5	162.0	6	413.5	5
Llama3.3-70B$_{\mathrm{fs}}$	207.5	6	164.0	5	371.5	6

Table 4: Ranking of models according to human preference.

We calculate a global score for each model, as follows: every time a model is voted, it gets 1 point, and the loser gets 0 points; in case of ties, both models get 0.5 points each. Table 4 shows the global scores, as well as the ranking of human preferences according to said score (second and third columns). We find that the gold standard reference obtains the highest score, as expected, with an specially stark difference in the out-of-domain set, highlighting the difficulty of this domain for all models. Fine-tuned Llama3 ranks the highest among the automatic methods in both in-domain and out-of-domain settings, which shows its ability to generalize to more challenging domains. The instruction-tuned model Llama3 Instruct obtains worse scores compared to its base model counterpart, which is likely due to instruction tuning reducing certain model capabilities (Li et al., 2024; West and Potts, 2025). It ranks higher than the NLLB model in-domain, but lower in the out-of-domain evaluation and the overall scores. Still, the difference between Llama3 and NLLB shows the potential to rival dedicated models in generation tasks. According to these preferences, fine-tuning LLMs for CS generation can be critical to ensure better results, since the larger models with few-shot prompting rank, both GPT-4o$_{\mathrm{fs}}$ and Llama3.3-70B$_{\mathrm{fs}}$, are outranked by all models in both evaluation settings.

5.2. Error analysis

In order to further explore differences between model performance, we analyze the most common errors made by the CS generation models, both quantitatively and qualitatively. We extend the MT error typology presented in Popović (2018) to CS generation error analysis. To do so, we randomly select a set of 100 outputs from all models and conduct a detailed examination of the types of errors

[4]A batch size of 32, learning rate of 1×10^{-4}, using constant learning rate schedule with $1{,}000$ warmup steps, a gradient clipping threshold of 1.0, and a weight decay of 1×10^{-3}. Training was conducted for $50{,}000 steps$. For evaluation, we selected the checkpoint that achieved the highest BLEU score on the development set.

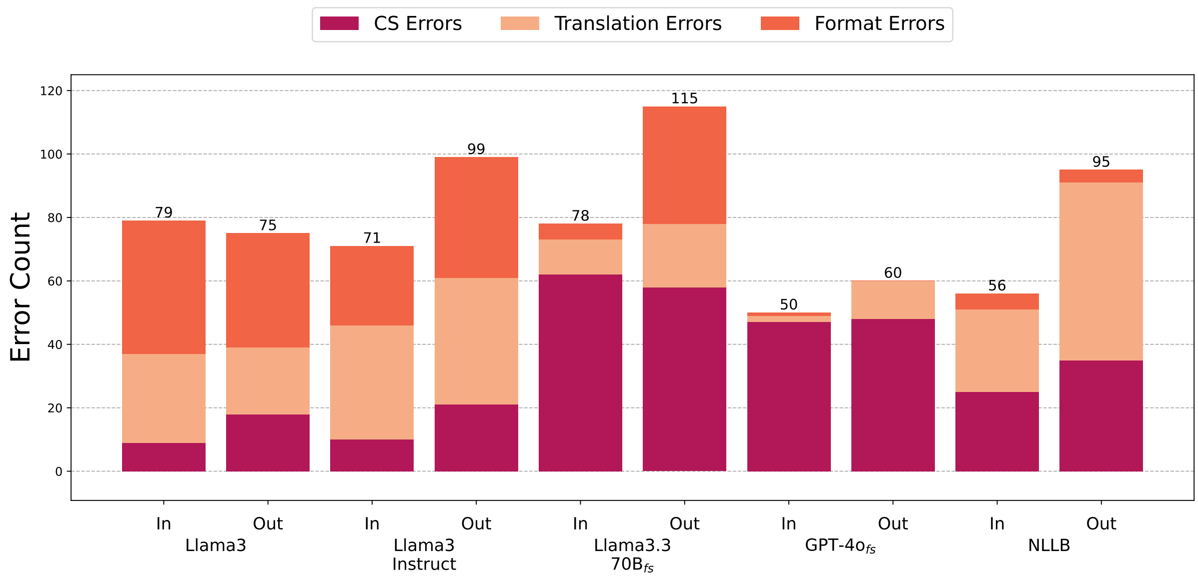

Figure 1: Error distribution by model, obtained by counting the number of instances that present errors of each type. Analyzed both for the in-domain (In) and out-of-domain (Out) sets.

present in them. This thorough analysis allows us to identify recurring patterns and propose a refined error typology specifically for automatic CS generation. This initial error analysis yields 18 total error categories, which we simplify and group into three main error types: a) CS errors, b) Translation Errors, and c) Format errors. The full error typology, along with detailed descriptions for each error type, is provided in Appendix F.

CS Errors: Errors of sentences that are either completely monolingual or switch between languages in an unnatural manner, e.g., by repeating the same word in English and Spanish. In Example 4, Llama3.3-70B$_{\mathrm{fs}}$ preserves the original meaning, but the sentence is fully monolingual.

(4)
Source yea... the best i can do is be here for him if he needs me
Output *sí... lo mejor que puedo hacer es estar aquí para él si me necesita*

Translation errors: Critical errors that either change the original meaning of the sentence or introduce mistakes in fluency or grammar, for example, using the wrong tense or word order. Example 5 shows an instance where Llama3 Instruct outputs a seemingly natural code-switched sentence, but the phrase "they got hurt" is not adequately translated and the meaning of the sentence is not preserved.

(5)
Source I wasn't happy because they got hurt.
Output i wasn't happy because *me dolieron*

Format errors: Errors in form that do not make the sentences unintelligible nor change their meaning, such as repetitions of a word or phrase or incorrect punctuation. Example 6, by the model Llama3, accurately preserves the original meaning and introduces CS, but removes the username and adds a smiley face.

(6)
Source <user> old mexican remedies
Output old school *remedios mexicanos* :)

We classify $1,000$ additional instances (200 instances per model, obtained from the same source sentences) into these kind of errors, and show the results in Figure 1.

For the in-domain test set, GPT-4o$_{\mathrm{fs}}$ makes the fewest errors overall (50), closely followed by NLLB (56). However, 90% of GPT's errors (45) and 45% on NLLB's (25) are CS related, indicating that while these systems preserve the meaning of sentences and generate few formatting errors, they often produce entirely monolingual outputs, which is a critical error. In comparison, CS-related mistakes are the least common in fine-tuned LLMs, accounting for less than 15% of the overall error count. This analysis shows that fine-tuned LLMs have effectively learned to switch between languages naturally, though they may still be prone to other less critical types of errors. The Llama3 base model struggles with maintaining the format of the sentences, which makes up 50.68% of its errors, whereas the instruction-tuned model Llama3 Instruct presents more meaning-related issues, $53,45\%$. This suggests that the linguistic knowledge of the model has degraded when tuned on instructions, a phenomenon that has been observed on other related

Model	BLEU		BERTScore		chrF	
	In domain	Out of domain	In domain	Out of domain	In domain	Out of domain
Llama3 8B	<u>34.49</u>	14.93	81.64	75.44	<u>53.17</u>	38.26
Llama3 8B Instruct	33.42	14.59	81.77	74.99	52.01	<u>38.62</u>
Llama3.3-70B$_{\mathrm{fs}}$	22.41	14.10	79.77	78.33	44.57	37.37
GPT-4o$_{\mathrm{fs}}$	32.25	<u>15.65</u>	<u>83.09</u>	<u>78.96</u>	50.48	<u>38.62</u>
NLLB	**35.56**	14.45	**84.11**	76.53	**54.74**	38.38
Identity	33.34	**41.54**	82.31	**83.31**	45.51	**58.03**

Table 5: Results of reference-based metrics the *EN-CS* test set. Best results in bold, second best results underlined.

areas (Fu et al., 2024). When evaluated on a different domain, the Llama3 Instruct and NLLB models produce notably more errors, suggesting that they are overfitting on the train set or have limited ability to generalize to more challenging domains. In particular, NLLB duplicates the number of translation errors. It is also noteworthy that Llama3.3 70B$_{\mathrm{fs}}$ shows a similar increase in errors, despite not being fine-tuned—a behavior that seems to arise due to the domain and longer sentences. This is not the case for Llama3 or GPT-4o$_{\mathrm{fs}}$, which output a similar number of errors for both domains. It is especially interesting that the Llama3 base model generalizes so well, despite extensive fine-tuning. Still, the small size of the out-of-domain test set calls for caution when interpreting these differences.

6. Automatic Evaluation

Previous research highlights the challenges of automatically evaluating code-switching (CS) generation, with many existing metrics showing low correlation with human judgments (Srivastava and Singh, 2021; Kuwanto et al., 2024). On the other hand, recent studies show that using LLMs as evaluators or judges can offer a promising alternative to evaluate generation tasks, as they show a higher alignment with human ratings (Chiang and Lee, 2023; Wang et al., 2023). In this section, we present the results of an automatic evaluation with reference-based NLG metrics (BLEU, BERTScore, chrF) and GPT-4o as a judge, as well as their correlation with human preferences.

6.1. Reference-based metrics

We report the results of BLEU (Papineni et al., 2002), BERTScore (Zhang et al., 2020), and chrF (Popović, 2015). [5] All three are task-agnostic quality metrics that give results between 0-1, based on character-level F-score, n-gram precision and semantic similarity using contextual embeddings[6] respectively. We compute the metrics for all systems, and include an Identity system that simply returns the provided input as the output.

The results of the evaluation can be seen in Table 5. In the in-domain setting, the best model is NLLB, with the highest scores for the three metrics. It is closely followed by GPT-4o$_{\mathrm{fs}}$, with the second highest BERTScore, and fine-tuned Llama3, with the second highest BLEU and chrF. Llama3 Instruct follows closely. The Identity system scores nearly as well as the top-performing models for the in-domain set and obtains the best scores out-of-domain. The strong results from the Identity baseline and few-shot models, which often produce monolingual outputs, suggest that reference-based metrics assign high scores to models that match only the English part of the reference. This reflects the nature of the task and dataset, and highlights the limitations and artifacts of using reference-based metrics to evaluate code-switched generation. Comparing these results with the out-of-domain evaluation, we observe that all models achieve considerably lower scores. Although GPT-4o$_{\mathrm{fs}}$ obtains the highest scores, the margin is relatively small. This drop in performance is likely due to specific characteristics of the out-of-domain dataset that negatively affect all models' predictions, such as longer sentence lengths.

6.2. GPT as a judge

As a complementary automatic assessment of the outputs of our models, we have implemented a zero-shot pair-wise evaluation using GPT-4o as a judge, mimicking the settings of the human evaluation. Details about the implementation are included in Appendix G. Results are shown in Table 6. GPT shows a strong preference for few-shot models, whereas these models are ranked last and second-to-last by humans. Based on the error analysis (c.f Section 5.2), few-shot models tend to make many CS-related mistakes, as they often output purely monolingual sentences, although they are the most fluent overall. This suggests that this disagreement may be caused by the humans adhering to the guidelines and taking the presence of CS as the main criterion, whereas GPT is making

[5]The metrics are implemented using the evaluate library.

[6]BERTscore has been calculated using the embeddings from the model Bert Base Multilingual Cased.

GPT		
Model	Score	Rank
Gold Standard	744.5	1
GPT-4o$_{\mathrm{fs}}$	603.0	2
Llama3.3-70B$_{\mathrm{fs}}$	582.0	3
Llama3 8B	440.5	4
Llama3 8B Instruct	410.0	5
NLLB	370.0	6

Table 6: Ranking of models using GPT as a judge.

decisions based on the style and fluency of the answers. Regarding fine-tuned LLMs, they share the same relative ranking in both evaluations, with Llama3 being the preferred model by humans and GPT. Finally, NLLB is ranked last by GPT, while it is the second best model overall as ranked by humans. Further research is needed to explain this behaviour, but there may be some stylistic features of the NLLB model's outputs that are affecting GPT's preferences.

6.3. Correlation With Human Evaluation

The reference-based metrics used in Section 6.1 are known to have weak correlations with human judgment in NLG tasks (Sai et al., 2022), whereas JudgeLLM-based evaluations seem promising (Chiang and Lee, 2023; Wang et al., 2023). In this section, we compare reference-based metrics and GPT scores with the preference-based scores obtained in Section 5.1, allowing us to examine how closely these automatic metrics align with human preferences.

We calculate Pearson's (ρ) correlation coefficient at instance-level, using the $1,000$ instances employed for the error classification and human evaluation (the output of 5 models for 200 source sentences).[7] Each data point corresponds to the CS output of one particular model for an English source sentence, and we compute the correlation using two values: the score obtained by the model for this instance in the human preference-based evaluation of Section 5.1, and the score it attains if we apply the same strategy using the values of the reference-based metrics to determine the winner, or, in case of JudgeLLM, the scores given by GPT. This allows us to directly compare the two types of evaluations on an instance-by-instance basis.

The correlation coefficients are shown in Figure 2. The top part of the figure shows the correlation using all the instances, whereas the bottom part only considers those instances that showed some type of error, according to the error analysis described

[7]We do not consider the reference CS sentences when calculating the correlations.

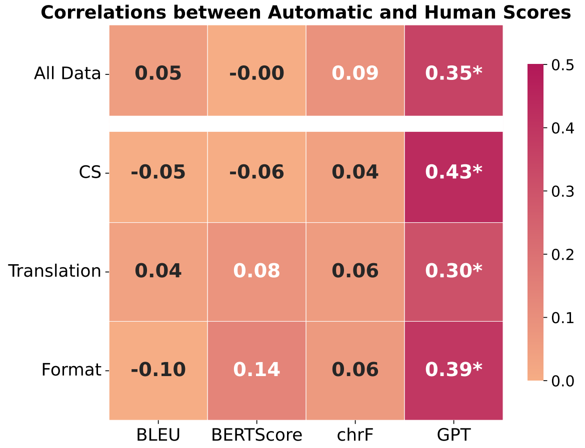

Figure 2: Heatmap of the correlations between human scores and reference-based metrics and scores given by GPT, calculated using the Pearson Correlation Coefficient. The correlations are calculated for all instances, as well as for different subsets of instances, according to the type of errors they exhibit. * indicates statistical significance ($p \leq 0.05$).

in Section 5.2. These results provide a more detailed view of how the metrics behave across different subsets of instances. If we consider all the instances, the maximum ρ correlation value with reference-based metrics is 0.09, which indicates a low alignment with human scores. GPT-4o$_{\mathrm{fs}}$ shows a ρ of 0.35, which is stronger than reference-based metrics, but still too weak to be regarded as a reliable measure for assessing CS generation.

If we instead consider instances with errors from the human evaluation, there is again a higher correlation between human scores and GPT's judgments, with a margin of at least 0.25 points. Instances with CS errors show the lowest overall correlation with reference-based metrics. This likely derives from the fact that human evaluators never prefer an instance without CS as instructed in the guidelines, but reference-based metrics are not sensitive to these nuances, and may assign high scores to instances regardless of whether they contain CS or not. GPT, however, obtains the highest correlation in these kinds of instances.

All in all, these results confirm that several of the most commonly used reference-based metrics for NLG have a weak correlation with human judgments when evaluating CS generation. This underscores the need to research more specialized evaluation methods designed specifically to capture the nuances of this task that correlate with human judgment.

7. Conclusion

In this work, we have presented a methodology to leverage LLMs in the generation of code-switched text from monolingual instances, specifically for the English-Spanish language pair.

Our framework consists of back-translating natural code-switched instances (EN-ES) into monolingual English sentences, and using the resulting parallel corpus, dubbed *EN-CS*, to fine-tune autoregressive models to translate monolingual sentences into CS. This has the advantage of ensuring that the target sentences contain completely natural CS, which has the potential to improve the naturalness of CS generation.

We experiment with fine-tuning base and instruction-tuned LLMs on our dataset using LoRA. For baselines, we include few-shot LLMs (Llama3.3-70B and GPT-4o) and a pretrained NLLB translation system that we also finetune using our dataset. The results indicate that fine-tuned LLMs show higher ranking in a human preference-based evaluation and fewer critical errors than the other baselines, performing better even than proprietary models such as GPT-4o.

We also perform a meta-evaluation of reference-based NLG metrics commonly used for CS evaluation, as well as an LLM judge (GPT-4o). Our analyses show low correlation between human and reference-based evaluations, while the LLM judge achieves moderate correlations. However, particularly in cases with CS errors, no metric is adequate for assessing CS generation. We therefore advocate for more research in specialized evaluation methods.

Limitations

Our research focuses on testing the capabilities of LLMs for CS generation, a field of interest in the research of many applications, yet still in need of more research. While our findings highlight promising potential, we also identify key areas for refinement and improvement, as well as promising lines for future research in this domain.

We want to acknowledge the fact that our approach is dependent on having an initial set of code-switched sentences, which may not be available for all pairs of languages, especially in a low-resource scenario. We believe that it would be interesting to explore the possibility of a cross-lingual approach using our methodology, with English and/or Spanish as pivot languages, that could be useful for transfer knowledge into other less-resourced language pairs.

Finally, as we have pointed out, we are aware of the problems of the automatic metrics that we have used to evaluate the outputs of our models, which do not capture the nuances of our task. In the future, we would like to investigate how to improve this evaluation by designing new methods to automatically evaluate CS generation, focusing on a more linguistic approach able to capture the linguistic and social intricacies of CS.

Acknowledgments

This work is supported by the projects DeepMinor (CNS2023-144375) funded by MTDFP/ and by European Union Next GenerationEU/ PRTR; HumanAIze (AIA2025-163322-C61) funded by MICIU/AEI/10.13039/501100011033; and DeepThought (PID2024-159202OB-C21) funded by MICIU/AEI /10.13039/501100011033 and by ERDF, EU. Maite Heredia holds a PhD grant from the University of the Basque Country UPV/EHU (PIF23/218).

Bibliographical 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.

Elena Álvarez-Mellado and Constantine Lignos. 2022. Detecting unassimilated borrowings in Spanish: An annotated corpus and approaches to modeling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3868–3888, Dublin, Ireland. Association for Computational Linguistics.

Rene Appel and Pieter C. Muysken. 2005. *Language Contact and Bilingualism*. Amsterdam University Press.

Kalika Bali, Jatin Sharma, Monojit Choudhury, and Yogarshi Vyas. 2014. "I am borrowing ya mixing ?" An analysis of English-Hindi code mixing in Facebook. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 116–126, Doha, Qatar. Association for Computational Linguistics.

Jeremy Barnes, Naiara Perez, Alba Bonet-Jover, and Begoña Altuna. 2025. Summarization metrics for Spanish and Basque: Do automatic scores and LLM-judges correlate with humans?

Anshul Bawa, Pranav Khadpe, Pratik Joshi, Kalika Bali, and Monojit Choudhury. 2020. Do multilingual users prefer chat-bots that code-mix? Let's nudge and find out! *Proc. ACM Hum.-Comput. Interact.*, 4(CSCW1).

Rachel Bawden and François Yvon. 2023. Investigating the translation performance of a large multilingual language model: the case of BLOOM. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 157–170, Tampere, Finland. European Association for Machine Translation.

Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. 2007. (Meta-) evaluation of machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 136–158, Prague, Czech Republic. Association for Computational Linguistics.

Cheng-Han Chiang and Hung-yi Lee. 2023. Can large language models be an alternative to human evaluations? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15607–15631, Toronto, Canada. Association for Computational Linguistics.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Michael Clyne. 1987. Constraints on code switching: how universal are they? *Linguistics*, 25(4):739–764.

Cohere For AI. 2024. c4ai-command-r-v01 (revision 8089a08).

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms.

Jean-Marc Dewaele and Li Wei. 2014. Attitudes towards code-switching among adult mono- and multilingual language users. *Journal of Multilingual and Multicultural Development*, 35(3):235–251.

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.

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

K. Angelique Dwyer. 2017. My brother the mexican. *PORTAL Journal of Multidisciplinary International Studies*, 14.

K. Angelique Dwyer. 2019a. Doce horas: A family border tale. *PORTAL Journal of Multidisciplinary International Studies*, 16:163–165.

K. Angelique Dwyer. 2019b. Gringos mexicanos. *PORTAL Journal of Multidisciplinary International Studies*, 16:160–162.

K. Angelique Dwyer. 2019c. La manda. *PORTAL Journal of Multidisciplinary International Studies*, 16:156–159.

K. Angelique Dwyer. 2019d. Simón. *PORTAL Journal of Multidisciplinary International Studies*, 16:153–155.

K. Angelique Dwyer. 2020. La vaca y coatlicue. *PORTAL Journal of Multidisciplinary International Studies*, 17:179–181.

Joseba Fernandez de Landa, Iker García-Ferrero, Ander Salaberria, and Jon Ander Campos. 2024. Uncovering social changes of the Basque speaking Twitter community during COVID-19 pandemic. In *Proceedings of the 3rd Annual Meeting of the Special Interest Group on Under-resourced Languages @ LREC-COLING 2024*, pages 363–371, Torino, Italia. ELRA and ICCL.

Tingchen Fu, Deng Cai, Lemao Liu, Shuming Shi, and Rui Yan. 2024. Disperse-then-merge: Pushing the limits of instruction tuning via alignment tax reduction. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2967–2985, Bangkok, Thailand. Association for Computational Linguistics.

Saurabh Garg, Tanmay Parekh, and Preethi Jyothi. 2018. Code-switched language models using dual RNNs and same-source pretraining. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3078–3083, Brussels, Belgium. Association for Computational Linguistics.

Hila Gonen and Yoav Goldberg. 2019. Language modeling for code-switching: Evaluation, integration of monolingual data, and discriminative training. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4175–4185, Hong Kong, China. Association for Computational Linguistics.

Bryan Gregorius and Takeshi Okadome. 2022. Generating code-switched text from monolingual text with dependency tree. In *Proceedings of the 20th Annual Workshop of the Australasian Language Technology Association*, pages 90–97, Adelaide, Australia. Australasian Language Technology Association.

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, Online. Association for Computational Linguistics.

Gualberto Guzmán, Joseph Ricard, Jacqueline Serigos, Barbara E. Bullock, and Almeida Jacqueline Toribio. 2017. Metrics for Modeling Code-Switching Across Corpora. In *Proc. Interspeech 2017*, pages 67–71.

Maite Heredia, Marianela Fernández Trinidad, and Miguel Jiménez-Bravo. 2025. Actitudes lingüísticas hacia el cambio de código entre valenciano y castellano. *Revista Española de Lingüística Aplicada/Spanish Journal of Applied Linguistics*.

I-Hung Hsu, Avik Ray, Shubham Garg, Nanyun Peng, and Jing Huang. 2023. Code-switched text synthesis in unseen language pairs. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5137–5151, Toronto, Canada. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

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.

Orreaga Ibarra Murillo. 2014. Tipología y pragmática del code-switching vasco-castellano en el habla informal de jóvenes bilingües. *Lapurdum*, pages 23–40.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, CA, USA.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine

Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

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.

Margaret Li, Weijia Shi, Artidoro Pagnoni, Peter West, and Ari Holtzman. 2024. Predicting vs. acting: A trade-off between world modeling & agent modeling. *ArXiv*, abs/2407.02446.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, Veselin Stoyanov, and Xian Li. 2022. Few-shot learning with multilingual generative language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9019–9052, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Giovanni Molina, Fahad AlGhamdi, Mahmoud Ghoneim, Abdelati Hawwari, Nicolas Rey-Villamizar, Mona Diab, and Thamar Solorio. 2016. Overview for the second shared task on language identification in code-switched data. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 40–49, Austin, Texas. Association for Computational Linguistics.

Iftitahu Nimah, Meng Fang, Vlado Menkovski, and Mykola Pechenizkiy. 2023. NLG evaluation metrics beyond correlation analysis: An empirical metric preference checklist. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1240–1266, Toronto, Canada. Association for Computational Linguistics.

NLLB Team, Marta Ruiz Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Alison Youngblood, Bapi Akula, Loïc Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon L. Spruit, C. Tran, Pierre Yves Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation. *ArXiv*, abs/2207.04672.

Aitor Ormazabal, Mikel Artetxe, Manex Agirrezabal, Aitor Soroa, and Eneko Agirre. 2022. PoeLM: A meter- and rhyme-controllable language model for unsupervised poetry generation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3655–3670, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

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.

Dwija Parikh and Thamar Solorio. 2021. Normalization and back-transliteration for code-switched data. In *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, pages 119–124, 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. 1980. Sometimes I'll start a sentence in Spanish y termino en español: toward a typology of code-switching 1. *Linguistics*, 18:581–618.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Maja Popović. 2018. *Error Classification and Analysis for Machine Translation Quality Assessment*, pages 129–158. Springer International Publishing, Cham.

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.

Sai Krishna Rallabandi and Alan W. Black. 2017. On building mixed lingual speech synthesis systems. In *Interspeech*.

sagorbrur. 2025. *codeswitch: A toolkit for code-switching text generation*. Language resource.

Ananya B. Sai, Akash Kumar Mohankumar, and Mitesh M. Khapra. 2022. A survey of evaluation metrics used for nlg systems. *ACM Comput. Surv.*, 55(2).

E. Sarkisov. 2021. Interlingual interference as a linguistic and cultural characteristic of the current online communication. *Russian Journal of Bilingualism Studies*, 3:16–21.

Noam M. Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. *ArXiv*, abs/1804.04235.

Thamar Solorio, Elizabeth Blair, Suraj Maharjan, Steven Bethard, Mona Diab, Mahmoud Ghoneim, Abdelati Hawwari, Fahad AlGhamdi, Julia Hirschberg, Alison Chang, and Pascale Fung. 2014. Overview for the first shared task on language identification in code-switched data. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 62–72, Doha, Qatar. Association for Computational Linguistics.

Vivek Srivastava and Mayank Singh. 2021. Challenges and limitations with the metrics measuring the complexity of code-mixed text. In *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, pages 6–14, Online. Association for Computational Linguistics.

Ishan Tarunesh, Syamantak Kumar, and Preethi Jyothi. 2021. From machine translation to code-switching: Generating high-quality code-switched text. 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 3154–3169, Online. Association for Computational Linguistics.

Michelle Terblanche, Kayode Olaleye, and Vukosi Marivate. 2024. Prompting towards alleviating code-switched data scarcity in under-resourced languages with GPT as a pivot. In *Proceedings of the 3rd Annual Meeting of the Special Interest Group on Under-resourced Languages @ LREC-COLING 2024*, pages 272–282, Torino, Italia. ELRA and ICCL.

Maxim Tkachenko, Mikhail Malyuk, Andrey Holmanyuk, and Nikolai Liubimov. 2020-2022. Label Studio: Data labeling software. Open source software available from https://github.com/heartexlabs/label-studio.

G Richard Tucker. 2001. A global perspective on bilingualism and bilingual education. *Georgetown University Round table on Languages and Linguistics 1999*.

Jiaan Wang, Yunlong Liang, Fandong Meng, Zengkui Sun, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. 2023. Is ChatGPT a good NLG evaluator? a preliminary study. In *Proceedings of the 4th New Frontiers in Summarization Workshop*, pages 1–11, Singapore. Association for Computational Linguistics.

Peter West and Christopher Potts. 2025. Base models beat aligned models at randomness and creativity. In *Second Conference on Language Modeling*.

Genta Winata, Alham Fikri Aji, Zheng Xin Yong, and Thamar Solorio. 2023. 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, Toronto, Canada. 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, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Jitao Xu and François Yvon. 2021. Can you traducir this? Machine translation for code-switched input. In *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, pages 84–94, Online. Association for Computational Linguistics.

Shu Xu and Michael F. Lorber. 2014. Interrater agreement statistics with skewed data: evaluation of alternatives to cohen's kappa. *Journal of consulting and clinical psychology*, 82(6):1219–1227.

Zheng Xin Yong, Ruochen Zhang, Jessica Forde, Skyler Wang, Arjun Subramonian, Holy Lovenia, Samuel Cahyawijaya, Genta Winata, Lintang Sutawika, Jan Christian Blaise Cruz, Yin Lin Tan, Long Phan, Long Phan, Rowena Garcia, Thamar Solorio, and Alham Fikri Aji. 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, Singapore. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. *ArXiv*, abs/1904.09675.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2024. Multilingual machine translation with large language models: Empirical results and analysis. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2765–2781, Mexico City, Mexico. Association for Computational Linguistics.

A. Prompt-tuning for CS-EN translation

For CS→EN translation of the LINCE benchmark, we test the prompts in Table 7, combined with 0-, 1- and 5-shot strategies. The prompts include the instructions explained in different ways, including more or less information.

For the few-shot strategies, the prompt includes the following template at the beginning, alongside a set of manually selected examples that are representative of some phenomena we want to cover in our prompt:

Here are {n} examples of a code-switched text that has been converted to {lang}:

Testing the different prompts, we are able to choose the one whose outputs are closest to our needs, taking into consideration the trade-off between including too little and too much level of specificity in the instructions to the models.

Regarding the few-shot strategies, we find out that giving some examples to the models results in outputs that are more aligned with the expected output, which is logical, since this allows the models to more faithfully replicate the examples provided. The more examples given, the more the model is able to comply to leaving the punctuation marks as they are and not standardizing the spelling, but also it tends to add more colloquial terms and alternate spellings. Here are the final 5 examples that were selected as they cover the majority of the phenomena that models were observed to struggle with:

Input cuando me gusta algo nunca lo hay mi fucking size o no tengo el dinero.

Output when I like something there's never my size or I don't have the money.

Input excelente compartir contigo gracias por tu amistad <user> u rock

Output excellent sharing with you thank you for your friendship <user> u rock

Input fuhk it tacos de frijol

Output fuhk it bean tacos

Input <user> como se llama esa app i wanna play it lmfao

Output <user> what's that app called i wanna play it lmfao

Input i tried putting fake eyelashes on rn lmao me ebarre de glue todo el pinche ojo jajajaja #osoalmil jajaja

Output i tried putting fake eyelashes on rn lmao i put glue all over my damn eye hahahaha #superclumsy hahaha

Convert this code-switched phrase to English.
Convert this code-switched phrase to English without correcting the original spelling, focus on translating the parts in Spanish.
Convert this code-switched phrase to English. Leave the parts in Spanish as they are, focus on translating the parts in Spanish.
Convert this code-switched phrase to English. Directly output the translation and don't correct the original spelling, focus on translating the parts in Spanish.

Table 7: Different prompts that have been used to convert the code-switched instances into English, with different levels of specificity. Final prompt in bold.

B. Post-edition Guidelines

The original sentence should contain **CS** and be **translatable**. The main reasons to **remove** an instance altogether are:

- If the sentence is very clearly monolingual and the CS has been detected incorrectly (eg, the case of interlingual homographs such as *has*).
- When the sentence is bilingual for metalinguistic reasons, because it makes the translation tricky and hard to understand, and in most cases it's not even CS.
- The part that is in the other language is a named entity, such as a title, a name, ...
- If the code-switched part is not translatable or very hard to translate, probably because it's a borrowing. Ambiguous and a little bit up to the annotator.
- If the tweet is saying the same thing in both languages (making it monolingual doesn't make sense).
- Some instances are tweets that are part of a conversation or thread and taken out of context are very hard to understand/intelligible.
- Some tweets are not translatable because of wordplay that doesn't transfer to monolingual speech.

The result should be a **monolingual** sentence that has roughly the **same meaning** as the original sentence. The main reasons to edit a translation are:

- If the meaning changes or the model has hallucinated extra information that wasn't present in the original sentence.
- If there are still some words in the Spanish.
- Attempts to translate named entities.
- Remove "meta comments" from the model about the task.

It is not necessary to correct things like:

- Punctuation marks.
- Different spellings of the same word.
- Words of phrases that the model has changed for synonyms.

Base model
I want to not work and make money. = quiero no trabajar and make money
Instruction-tuned model
system prompt: "*You are a bilingual speaker of English and Spanish. Translate the following English sentence into code-switched text between both languages:*" **user**: "*I want to not work and make money.*" **assistant**: "*quiero no trabajar and make money*"
Few-shot prompting
system prompt: "*You are a bilingual speaker of English and Spanish. Translate the following English sentence into code-switched text between both languages. Do not add any comments or explanations:*" **user**: Source example n **assistant**: Target example n **user**: "*I want to not work and make money.*" **assistant**: ...

Table 8: Examples and format of prompts used for finetuning base and instruction-tuned models and for few-shot prompting

C. Inter-annotator agreement

Figure 3 shows the distribution of the scores given by to annotators on the fluency and adequacy of the instances translated by Command R. Although the inter-annotator agreement shows moderate agreement ($\kappa = 0.57$), the distributions between the annotators are very similar to each other.

D. Fine-tuning / Few-shot prompting

In Table 8, we can see the prompt used for a)fine-tuning of base models; b) fine-tuning of instruction-tuned models; and c) 5-shot prompting. For both fine-tuning settings, at inference time the second part of the prompt that contains the target CS sentence is left blank for the model to complete. For the few-shot prompting approach, the examples are selected randomly among the rest of the instances of the test set.

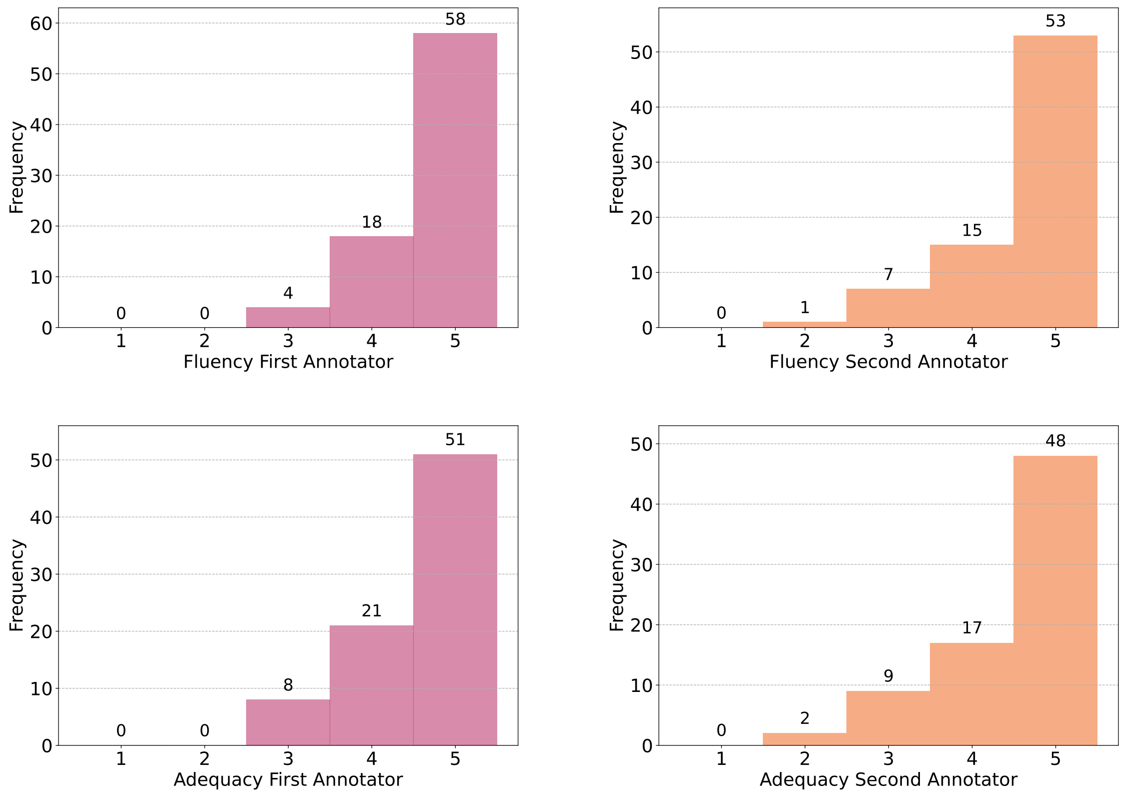

Figure 3: Distribution of Adequacy and Fluency scores per annotator.

E. Pairwise Annotation Guidelines

The main objective of this task is two evaluate a pair of sentences that **should contain code-switching between English and Spanish**. It should be noted that models have been trained with texts extracted from social media and informal conversations, so **the outputs of the models are expected to present traits of informality**, such as common typos, that at first should not be considered errors, because they are within the expected behaviour of the models. The criteria to choose between both sentences is to be applied **in the following order**:

1. **Code-switching**
    1.1. **Presence of code-switching**: For a sentence to be a suitable candidate it must have tokens in both languages. A completely monolingual sentence will always be wrong.
    1.2. **Naturalness of the code-switching**: A switch between both languages can be unnatural. There are different linguistic constraints. For example, a switch is only possible at a point in a sentence where it does not violate the syntactic rules of either language.
2. **Content and fluency**
    2.1. **Content**: Sentences must have meaning as a whole, they have to be understandable, without extra content disconnected from the rest of the message or abrupt interruptions.
    2.2. **Agreement**: Sentences must have the right gender and number agreement.
    2.3. **Conjugation**: Verbs have to be correctly conjugated.
3. **Form**: Additional errors that can be used in case none of the above are applicable.
    3.1. **Repetitions** of the same word or phrase.
    3.2. **Misspelled words / uncommon typos**
    3.3. **Wrong punctuation marks**
    3.4. **Extra characters**

Ties are only contemplated in two situations:

- Two sentences that are **equally wrong**, that is to say, they are both either completely monolingual or unintelligible.
- Two sentences that are **exactly the same** and thus no criteria can be used to break the tie.

In case no criteria is applicable to a pair, we ask the annotators to choose their preferred sentence, using their own judgment o additional criteria they might observe in the specific pair of sentences.

F. Error Typology

1. **CS errors**
   1.1. **No CS** - the sentence is entirely monolingual.
   1.2. **Unnatural CS** - the sentence contains unnatural CS, either due to unnatural switching points, or unnatural register.
   1.3. **Repetition in both languages** - the sentence contains the same information repeated in both languages, rather than CS.
2. **Translation errors**
   2.1. **Made-up words** - the words in the output look like English or Spanish but do no actually exist.
   2.2. **Wrong translation** - the translation of a word or phrase is incorrect.
   2.3. **Wrong conjugation** - a verb is translated with the right lexeme but a seemingly made-up conjugation.
   2.4. **Wrong agreement** - there is a mistake in agreement in gender or number.
   2.5. **Wrong meaning** - a word or phrase has been translated into a sense that does not fit into the context.
   2.6. **Wrong order** - the words are right but they are written in the wrong order.
   2.7. **Wrong tense** - the verbal tense is not consistent through the sentence.
   2.8. **Unintelligible** - it is not possible to understand the sentence in English nor in Spanish.
   2.9. **Instruction misunderstanding** - the task has been misunderstood, e.g., the model makes a "comment" about the content of the output or explains a word.
3. **Format errors**
   3.1. **Extra words** - the sentence contains seemingly random extra words that do not affect its meaning.
   3.2. **Extra characters** - the sentence contains more non-word characters than the original, e.g., '???' instead of '??'.
   3.3. **Hallucinations** - the sentence contains new words or phrases not derived from the original text.
   3.4. **Start over** - the sentence is finalized, but the model begins a second translation of the same sentence.
   3.5. **Duplications** - some words or phrases of the sentence are duplicated.

G. Implementation of GPT as judge

For the implementation of GPT as judge, the developer and user prompts in Table 9 have been used to prompt GPT-4o. To calculate the scores, we first check the answers the directly contain the desired format, "A", "B or "T", which are the most common. For the rest of the outputs that did not follow this format, it is possible to extract the labels using simple regular expressions. Then, the scores are calculated just like the human score: every time a model is voted, it gets 1 point, and the loser gets 0 points; in case of ties, both models get 0.5 points each.

developer: "You are a helpful bilingual system that knows how to code-switch between English and Spanish and how to distinguish natural sentences. Your only job is to judge sentences and output a verdict A, B or T." **user**:"Which one of the next two automatically generated sentences with code-switching is more natural? The most important criterion is that the sentences must have code-switching to even be considered eligible. Two sentences can be tied if they are equally wrong. A: {s1} B: {s2} Answer(A/B/T):"

Table 9: Prompt used for GPT to act as a judge.