Leveraging Large Language Models for Code-Mixed Data Augmentation in Sentiment Analysis

Linda Zeng
The Harker School
500 Saratoga Ave, San Jose, CA 95129
26lindaz@students.harker.org

Abstract

Code-mixing (CM), where speakers blend languages within a single expression, is prevalent in multilingual societies but poses challenges for natural language processing due to its complexity and limited data. We propose using a large language model to generate synthetic CM data, which is then used to enhance the performance of task-specific models for CM sentiment analysis. Our results show that in Spanish-English, synthetic data improved the F1 score by 9.32%, outperforming previous augmentation techniques. However, in Malayalam-English, synthetic data only helped when the baseline was low; with strong natural data, additional synthetic data offered little benefit. Human evaluation confirmed that this approach is a simple, cost-effective way to generate natural-sounding CM sentences, particularly beneficial for low baselines. Our findings suggest that few-shot prompting of large language models is a promising method for CM data augmentation and has significant impact on improving sentiment analysis, an important element in the development of social influence systems.

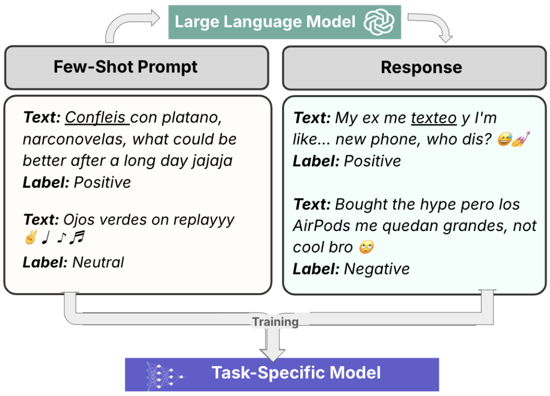

Figure 1: Overall system workflow with examples of Spanish-English CM tweets as natural data (left) and synthetic data (right). Underlined words represent Spanish-English hybrid words, examples of the complexities introduced by CM. Translations of CM sentences into English are provided in Appendix A.

1 Introduction

Code-mixing (CM), or code-switching, is the practice of switching between languages within a conversation or utterance. This practice is integral to multilingual societies, particularly in Mexico and urban India (Parshad et al., 2016), and is also significant in computer-mediated communication and social media, where multilingual users are predominant (Rijhwani et al., 2017). Despite its ubiquity, CM is mostly spoken and found in personal messages, making training data scarce and leading to poorer Natural Language Processing (NLP) model performance compared to monolingual text (Pratapa et al., 2018; Yong et al., 2023).

Social influence (SI) refers to the changes in thoughts, feelings, attitudes, or behaviors resulting from interactions with others. In multilingual societies, CM reflects an important aspect of these interactions, reflecting social dynamics and identity. Sentiment analysis (SA) is crucial for understanding these dynamics, as it captures the emotional nuances embedded in multilingual interactions. Furthermore, SA has become a primary CM task due to its need for complex semantic understanding and its implications for social media (Drus and Khalid, 2019), where CM is commonly present (Srinivasan and Subalalitha, 2023). By accurately analyzing sentiment in code-mixed text, SI systems enhance their ability to interpret user intent and emotional states, enabling more meaningful interactions addressing the more diverse environments in which SI occurs. Since multilingual speakers bridge information on social media (Li and Murray, 2022), machines must also accurately analyze CM text to capture public opinion and disseminate news. However, current approaches fall short in handling code-mixed settings (Doğruöz et al., 2021; Aguilar et al., 2020) due to data scarcity.

Beyond the CM domain, few-shot learning has shown promise in overcoming data scarcity, as

Proceedings of the Second Workshop on Social Influence in Conversations (SICon 2024), pages 85–101
November 16, 2024 ©2024 Association for Computational Linguistics

Large Language Models (LLMs) trained on diverse tasks generalize to new ones with minimal training (Brown et al., 2020; Lin et al., 2022; Winata et al., 2021). LLMs are used for data augmentation (Ding et al., 2024; Whitehouse et al., 2023; Yoo et al., 2021; Dai et al., 2023), training data generation (Yu et al., 2023), and knowledge distillation (Xu et al., 2024; Phuong and Lampert, 2021), particularly in low-resource settings (Ding et al., 2024). However, this approach remains underexplored in the CM domain, which presents unique challenges (Zhang et al., 2023).

In this work, we bring LLM-powered data augmentation to the task of code-mixed sentiment analysis. We use few-shot prompting to generate labeled CM SA data in Spanish-English and low-resource Malayalam-English. Following Li and Murray (2023); Whitehouse et al. (2023); Tareq et al. (2023), we quantify the performance gains by fine-tuning multilingual pre-trained language models (PLMs) on the LLM-generated data. We investigate if these synthetic data samples can reflect natural code-mixing patterns and nuances compared to other data augmentation techniques and verify the synthetic data quality through human evaluation.

Figure 1 displays our overall system workflow with examples of natural and synthetic data. We summarize our contributions as follows:

- We introduce LLMs for CM data augmentation as a simple, cost-effective way to improve sentiment analysis models with natural-sounding sentences;
- We surpass past baselines, achieving third on the LinCE benchmark (Aguilar et al., 2020) in Spanish-English and outperforming the highest published benchmark by 4.85% on the low-resource MalayalamMixSentiment dataset (Chakravarthi et al., 2020);
- We thoroughly analyze the efficacy of our data augmentation approach in comparison to other techniques and with human evaluation;
- We release the synthetic data and code on Github[1] for public use and reproducibility.

2 Related Work

2.1 Data Augmentation for Code-Mixing

Existing attempts at generating synthetic CM data focus on using linguistics theory or converting monolingual data to CM data.

For instance, Pratapa et al. (2018) use Equivalence Constraint Theory to align the parse trees of Hindi and English and replace words in one language with their corresponding words in the second language. Lee et al. (2019) apply Matrix Language Frame theory to convert parallel data to CM data, and Gregorius and Okadome (2022) use a dependency tree which predicts code-switching points and a machine translator to convert monolingual sentences to CM. While these methods consider the intention behind code-switching points (Solorio and Liu, 2008), they require expert linguistic knowledge, assume languages pairs can be parsed by the same parse tree, and rely on the accuracy of the parsers employed.

Other approaches convert monolingual data into CM using machine translation systems (Vu et al., 2012; Li and Murray, 2022; Tarunesh et al., 2021), word dictionaries (Tareq et al., 2023), or parallel corpora (Winata et al., 2019; Whitehouse et al., 2022). For instance, Winata et al. (2019) employ a sequence-to-sequence model to learn language-switching points while Chang et al. (2019) use generative-adversarial networks. Li and Murray (2022) introduce language-agnostic masks in a monolingual SA corpus to train models on recognizing the patterns of CM, and Tareq et al. (2023) utilize word dictionaries to map monolingual data into CM SA data. Although some of these techniques account for code-switching points, they do not consistently produce natural sentences. Moreover, their effectiveness relies on the quality of the underlying systems and the assumption that large datasets with distributions similar to real CM data are available.

Unlike conversion-based methods, our approach generates CM sentences from scratch. By leveraging LLMs' multilingual pre-training and generalization capabilities, we aim to produce synthetic data that more accurately reflects the natural patterns and nuances of human-generated CM language.

2.2 Large Language Models for Code-Mixing

To our knowledge, LLMs have not yet been used for CM data augmentation. The closest related works are by Yong et al. (2023), who explore LLMs in South Asian CM dialects through prompting experiments, and Zhang et al. (2023), who assess LLMs' zero-shot performance on various CM tasks, including SA. Both studies find that LLMs need

[1] https://github.com/lindazeng979/LLM-CMSA

significant improvement on *zero-shot* CM tasks but do not explore if LLM-generated data can help *task-specific models* improve their training, despite sub-optimal LLM zero-shot performance. Notably, both studies found that GPT-3.5 (Brown et al., 2020) shows superior performance among LLMs and do not evaluate the more advanced GPT-4 (Achiam et al., 2024). Our research builds on their findings by using GPT-4 for data generation and fine-tuning task-specific models in addition to evaluating zero-shot performance.

In contrast to the findings of Yong et al. (2023) and Zhang et al. (2023), Whitehouse et al. (2023) report improvements using GPT-4 for data augmentation in cross-lingual commonsense reasoning tasks. While cross-lingual tasks involve separate languages, code-mixed tasks involve language switching within sentences. Nonetheless, the success reported by Whitehouse et al. (2023) supports the feasibility of our approach.

3 Methods

In this section, we introduce our data, the synthetic generation process, and our fine-tuning methods.

3.1 Natural Data

We conducted experiments using two human-labeled datasets which we call our natural data. The first is the Spanish-English SA dataset from the LinCE Benchmark (Aguilar et al., 2020), containing 18,789 CM tweets with code-mixing between English and Spanish. The second dataset is the Malayalam-English SA dataset from the MalayalamMixSentiment dataset (Chakravarthi et al., 2020), containing 5,452 CM YouTube movie review comments with code-mixing between English and Malayalam, a low-resource Dravidian language. The mean sentence lengths for both datasets are shown in Table 1.

Both datasets feature colloquial CM social media comments with diverse code-mixing patterns, presenting significant challenges to NLP models. They include sentiment categories: *Positive*, *Negative*, or *Neutral*. For preprocessing, we filtered out comments labeled "non-Malayalam" or "unknown" from the Malayalam-English dataset and adjusted the data splits. Both datasets were cleaned to remove empty strings, hashtags, URLs, and symbols, with emojis replaced by English descriptions using the emoji library.[2]

[2] https://pypi.org/project/emoji/

Language	Natural	LLM-Generated	Random Translation
Sp-En	13.0 ±7.4	14.7 ±4.0	23.1 ±30.4
Ma-En	8.2 ±3.1	8.4 ±1.7	N/A

Table 1: Mean sentence length and standard deviation, measured in words, of natural and synthetic data for each language.

3.2 Data Augmentation Methods

Our primary data augmentation method involves prompting LLM with task demonstrations to generate synthetic CM training data. As a secondary method to use for comparison, we implement the more traditional technique of translating monolingual sentences into CM.

3.2.1 LLM Prompting

We use GPT-4 as our LLM, as many past studies (Whitehouse et al., 2023; Yong et al., 2023; Zhang et al., 2023) have found high CM performance in GPT-based models. We construct instructions for GPT-4 based on previously successful CM generation prompts (Whitehouse et al., 2023; Yong et al., 2023) and empirical observations of the data. Additionally, we provide task demonstrations randomly sampled from the natural pre-processed training dataset, which may again appear in the SA fine-tuning phase, with an equal amount of demonstrations for each class. Since LLM requires few task demonstrations, this data augmentation approach is not contingent on having a large dataset, and synthetic data generation utilized 15 to 50 examples. The prompt refinement process, our final prompt, and data generation implementation details can be viewed in Appendix B.1.

Our final synthetic data sizes were ~53,000 in Spanish-English and ~24,000 in Malayalam-English. Shown in Table 1, LLM-generated sentences effectively resembled natural CM sentences in mean sentence length. However, LLM-generated sentences tended to vary less in sentence length, indicated by consistently lower standard deviation values.

3.2.2 Random Translation

Our secondary technique, Random Translation, converts a monolingual SA corpus into a CM SA corpus using machine translation. Similar to Li and Murray (2022); Tareq et al. (2023); Tarunesh et al. (2021), we used Stanford's Sentiment140 dataset (Go et al., 2009) and SemEval's Sentiment Analysis in Twitter dataset (Rosenthal et al., 2017) as monolingual corpora and randomly translated parts of English tweets into Spanish through Marian NMT (Junczys-Dowmunt et al., 2018). We did not use this technique for Malayalam-English due to the lack of reliable machine translation systems supporting Malayalam.

The resulting synthetic corpus consisted of 49,560 data samples. As shown in Table 1, the randomly translated data exhibited a significantly higher mean sentence length compared to LLM-generated synthetic data, due to constraints imposed by the statistics of the selected monolingual dataset. This highlights the limited flexibility of using pre-existing datasets for CM data augmentation.

3.3 Fine-tuning Sentiment Analysis

We fine-tuned multilingual BERT (mBERT), which was most commonly used in past benchmarks (Chakravarthi et al., 2020; Aguilar et al., 2020), and XLM-T, which is a XLM-R (Conneau et al., 2020) model pre-trained on millions of social media tweets from over thirty languages including Spanish and Malayalam. For each language, we trained both models on three datasets: only natural data, only synthetic data, and a combined dataset of natural and synthetic data. We also introduced a lower-resource experimental setup for Spanish-English, where we reduced the natural data to a 3,000-sample subset to align with Li and Murray (2022). Table 2 summarize the data sizes used in the full Spanish-English, subset of Spanish-English, and Malayalam-English experimental setups. For the full 12.2k Spanish-English data setup, we repeated experiments using both LLM-generated and randomly-translated synthetic data to compare the two techniques. In all, we hypothesized that training on both natural and synthetic data would lead to the highest performance, as it benefited from both natural data, which had a similar distribution and style as the natural test data, and synthetic data, which increased the number of examples for models to learn CM features.

In all Spanish-English experiments, when training on a combination of synthetic and natural data, we adopted the gradual fine-tuning method proposed by Xu et al. (2021) and applied to CM data augmentation by Li and Murray (2022). Treating the synthetic CM data as out-of-domain data, we

Language	**Train**		**Val**	**Test**
	Natural	*Synthetic*		
Sp-En	12,194	50,000	1,859	4,736
Sp-En	3,000	50,000	1,859	4,736
Ma-En	3,452	15,000	1,000	1,000

Table 2: Training, validation, and test data sizes for each round of experiments. Each row included training on natural data, synthetic data, and the combined (natural + synthetic) data, repeated for mBERT and XLM-T across different types of synthetic data.

fine-tuned the model for five stages, gradually decreasing the amount of synthetic data from 50,000 to {25000, 15000, 5000, 0} for subsequent training stages while keeping natural data constant. As a result, the model gradually fit closer to natural data, which it would be tested on. In Malayalam-English, we retained one stage of training due to higher performance after preliminary experimentation. Fine-tuning hyperparameters and the impact of gradual fine-tuning are included in Appendix B.2 and Appendix C, respectively.

4 Results

This section evaluates overall model performance and then quantifies relative percent improvements contributed by data augmentation.

4.1 Overall Performance

Table 3 presents the overall F1 scores achieved for the Spanish-English and Malayalam-English CM SA datasets in the full 12.2k and 3.5k data setup, respectively, compared to zero-shot scores, baseline scores, and current benchmarks. All Spanish-English models were evaluated using the same test dataset as the LinCE benchmark. However, the Malayalam-English models used adjusted train-test splits in comparison to benchmarks, due to the removal of extraneous labels (see Section 3.1).

4.1.1 Baselines

To provide reference points, GPT-4, mBERT, and XLM-T were evaluated using a zero-shot approach, where no additional training or fine-tuning was applied. For GPT-4, we generated predictions by providing a prompt with no examples and parsing the generated outputs directly as the model's predictions. For mBERT and XLM-T, we loaded in the pre-trained models with an extra classification

Method	Model	Natural Data	Synthetic Data	Spanish-English F1	Malayalam-English F1
Zero-shot	GPT-4			0.546	0.524
No Training	mBERT			0.045	0.131
No Training	XLM-T			0.543	0.354
Dataset Baseline	mBERT	✓		0.564	0.750
Our Baseline	XLM-T	✓		0.588	**0.843**
Random Translation	XLM-T		✓	0.491	
LLM-Generated	XLM-T		✓	0.544	0.595
Random Translation	XLM-T	✓	✓	0.563	
LLM-Generated	XLM-T	✓	✓	**0.603**	0.763
Top Score				0.622	0.804

Table 3: Summary of weighted F1 scores on the full 12k Spanish-English and 3.5k Malayalam-English datasets with comparisons to other baselines. Scores in bold indicate our highest performance on each dataset. The top score for Spanish-English is anonymous on the LinCE benchmark, and the top score for Malayalam-English is Bai et al. (2021).

layer and proceeded directly to evaluation without further training. Results are shown in the first section of Table 3.

Our zero-shot analysis reveals three main findings. First, consistent with Zhang et al. (2023), large language models like GPT-4 are still not sufficiently adept for zero-shot tasks like Spanish-English and Malayalam-English sentiment analysis, as they perform below dataset benchmarks (Aguilar et al., 2020; Chakravarthi et al., 2020). However, GPT-4's zero-shot performance on Malayalam-English is still surprisingly high considering the language is low-resource. Second, the size of an LLM does not necessarily equate to better performance. XLM-T, with its task-specific pre-training on code-mixed data from Common Crawl and Twitter (Li and Murray, 2022), demonstrates that a smaller, specialized model can be nearly as effective as a much larger general-purpose model in Spanish-English, aligning with Zhang et al. (2023). Lastly, XLM-T shows a significant zero-shot performance boost over mBERT for both Spanish-English and Malayalam-English, demonstrating the importance of task-specific pre-training.

The second section of Table 3 shows results after fine-tuning XLM-T on the full natural data. XLM-T consistently outperforms mBERT in both languages, similar to its zero-shot performance. Our Spanish-English baseline with XLM-T surpasses the LinCE Organizers' baseline using mBERT, and our Malayalam-English baseline achieves the highest score on this dataset, exceeding the previous top score by Bai et al. (2021).

4.1.2 Performance with Synthetic Data

The third and fourth sections of Table 3 display results when fine-tuning XLM-T on solely synthetic data and on a combination of natural and synthetic data, respectively.

When fine-tuning XLM-T on solely synthetic Spanish-English data, LLM-generated data slightly improves performance compared to no training, whereas randomly-translated data decrease performance below zero-shot levels.

Combining random-translated data with the full natural Spanish-English data similarly degrades performance relative to our baseline, highlighting its less effective representation of code-mixing. On the other hand, combining natural and LLM-generated synthetic data yields our highest Spanish-English score of 0.603 F1, ranking third on the LinCE benchmark. This demonstrates that LLM-generated data can mitigate overfitting and enhance task-specific model performance beyond LLM's own zero-shot capabilities in Spanish-English.

For Malayalam-English, training on either synthetic or natural data significantly improves performance compared to zero-shot results. LLM-generated synthetic data nearly double XLM-T's performance, and natural data more than double it, achieving higher scores than Spanish-English. Training with both natural and synthetic data averages their individual performances, suggesting that there exists a performance threshold past which synthetic data can no longer help. Nonetheless, the combination surpasses the dataset benchmark (Chakravarthi et al., 2020).

4.2 Contribution of Data Augmentation

Table 4 displays the relative improvements from data augmentation techniques on the three data setups: the full Spanish-English dataset, the subset of the Spanish-English dataset, and the Malayalam-English dataset. Unlike absolute scores, which can vary with training conditions, percent improvements provide a consistent measure for comparing models trained with and without synthetic data.

4.2.1 Full Spanish-English Dataset

The contrast in relative improvements between the LLM-Generated technique and the Random Translation technique, which are shown in the first section of Table 4, can be attributed to two factors: First, the monolingual corpora used for Random Translation did not closely match the distribution of natural CM data, and second, the code-switching points in the synthetic data were randomly generated. Since LLM-generated data did not experience the same performance losses, it mitigated these issues by producing sentences that more accurately reflected natural data distributions and incorporated intentional code-switching rather than random occurrences.

4.2.2 Subset of Spanish-English Dataset

In the subset of the Spanish-English dataset, where the training set was reduced to 3,000 samples, LLM-generated data showed a more substantial improvement for both models than on the full Spanish-English dataset, displayed in the second section of Table 4. These improvements outperformed the results obtained by Li and Murray (2022), indicating that LLM-generated data samples are particularly effective in a Spanish-English low-resource setting.

4.2.3 Malayalam-English Dataset

Displayed in the third section of Table 4, the high baseline accuracy of XLM-T in Malayalam-English led to a performance drop with synthetic data, while mBERT's performance improved slightly. In comparison, Li and Murray (2022) cite large improvements using their language-agnostic method, which reduces the focus on Malayalam's particular language features and emphasizes learning CM patterns. Nonetheless, this method also improves on a lower baseline score. These disparities suggest that the utility of synthetic data may diminish when the model's baseline performance is already high.

4.2.4 Cross-Dataset Analysis

Across all datasets, synthetic data generally enhances performance up to a certain threshold. Models with lower initial baselines, such as those trained on the limited Spanish-English subset, show greater percent improvements with synthetic data, reaching almost the same performance as models with quadruple the amount of natural data. This performance stability suggests that LLM-powered data can effectively boost performance for relatively small datasets. Conversely, models with high initial baselines, like XLM-T in Malayalam-English, may experience a decrease in accuracy when synthetic data samples are added, as synthetic data maintain performance at a similar threshold.

Overall, LLM-powered data augmentation proves effective in improving five of six models for CM SA, with our Spanish-English system achieving a notable 9.32% relative percent improvement, surpassing other methods such as Li and Murray (2022) under similar conditions.

5 Analysis

This section details results from human evaluation, subsequent empirical data analysis, and discussion about the trade-offs of generating synthetic data.

5.1 Human Evaluation

To gain insight on the quality of LLM-generated data, we asked native speakers to evaluate Spanish-English and Malayalam-English sentences from both the original dataset and the LLM-generated dataset on the grounds of *Code-Mixing Naturalness, Label Accuracy,* and if the sentences are *Human or Machine-Generated.* 400 Malayalam-English sentences were labeled by one annotator, and 200 Spanish-English sentences were labeled by two annotators, all of whom were balanced bilinguals with C1-C2 proficiency in the languages they annotated, according to the Common European Framework of Reference for Languages (CEFR). Detailed instructions for evaluators and descriptions of each label are elaborated in Appendix D. In this study, our human evaluation was constrained due to limited resources. While this is a limitation, it is worth noting that other studies, such as Whitehouse et al. (2023), have worked with even smaller sample sizes.

As shown in the first graph of Figure 2, annotators rated LLM-generated sentences similarly to human-generated sentences in terms of naturalness for both datasets. This suggests that

Dataset	Method	Model	Baseline	+Synthetic	% Change
Full Spanish-English$_{12.2k}$	LLM-Generated	XLM-T	0.588	0.603	2.55%
	LLM-Generated	mBERT	0.503	0.533	**5.96%**
	Random Translation	XLM-T	0.588	0.491	-16.5%
	Random Translation	mBERT	0.503	0.512	1.79%
Subset of Spanish-English$_{3k}$	LLM-Generated	XLM-T	0.547	0.598	**9.32%**
	LLM-Generated	mBERT	0.487	0.526	8.01%
	Li and Murray (2022)	XLM-T	0.649	0.660	1.68%
	Li and Murray (2022)	mBERT	0.495	0.506	2.12%
Malayalam-English$_{3.5k}$	LLM-Generated	XLM-T	0.843	0.763	-9.84%
	LLM-Generated	mBERT	0.737	0.745	1.09%
	Li and Murray (2022)	mBERT	0.670	0.722	**7.73%**

Table 4: A comparison of relative percent improvements achieved by different data augmentation methods on our three datasets for XLM-T and mBERT, with the largest improvements highlighted in bold. F1 scores are also provided from fine-tuning on natural data and on a combination of natural and synthetic data.

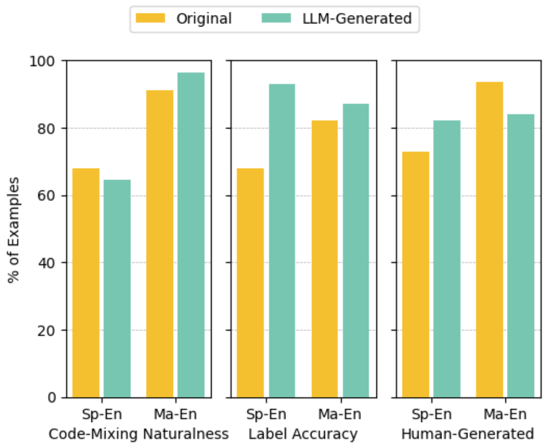

Figure 2: Human evaluation on Spanish-English and Malayalam-English sentences from the original datasets and the LLM-Generated datasets.

LLM-generated sentences did not appear unnatural when compared to human sentences. Notably, our Malayalam-English annotator labeled 5.5% more synthetic sentences as natural compared to human sentences. Since we define CM naturalness as the fluency of a sentence such that it can be recognized and accepted as authentic CM in real-life contexts, this finding indicates that, despite the differences in appearance between LLM-generated and natural data, both forms may be perceived as valid representations of CM in the real world. Furthermore, while there is a slight increase in the rating of synthetic sentences in Malayalam-English, the difference is relatively small and may not represent a significant divergence between LLM-generated and human sentences in terms of perceived naturalness.

Consistent across both datasets, LLM-generated data exhibited significantly higher sentiment label accuracy compared to human-generated data, shown in the second graph of Figure 2. This finding suggests that LLM-generated samples are less ambiguous, likely because we explicitly prompt GPT-4 to generate sentences for the sentiment analysis task. In contrast, real-world social media tweets, created without this directive, may exhibit greater semantic variability. These results highlight potential label ambiguity issues in the original datasets, particularly for Spanish-English, and demonstrate the utility of synthetic sentences to mitigate these issues by providing clearer examples during training. However, less ambiguous synthetic data may also lead to models that are less robust to natural complexities in human expression.

When predicting whether a sentence was human- or machine-generated, annotators faced significant challenges in distinguishing between LLM-generated and human sentences, shown in Figure 2. For Spanish-English, annotators mistakenly identified more LLM-generated sentences as human-produced than actual human sentences. In Malayalam-English, while annotators more accurately identified human sentences, a substantial margin of error persisted. Consequently, even though annotators tended to rate certain groups with higher naturalness or label accuracy, they lacked a clear understanding and identifiable cues indicating the sentences' original sources.

Ultimately, inter-annotator agreement was low for Spanish-English ($\kappa < 0.3$). While our findings

Sentences	Label	Prediction
Happy Friday #elvacilondelaGatita	neutral	positive
#elvacilondelagatita #quotes #friday	positive	neutral
Sentences	**Label**	**Correction**
Get your outfit now! Escoge tus prendas favoritas y haz tu pedido Blusa morada $20.00 #ilovesalhuaclothing	neutral	positive
Como me encabrona enterarme de quien se va en The Bachelor sin haber visto el episodio Angry Face	positive	negative

Table 5: Examples of sentences from the natural Spanish-English dataset, including their true labels, XLM-T's predicted labels, and the proposed corrections by human evaluators. Translations of the CM sentences into English are provided in Table 7 in Appendix A.

offer a qualitative perspective to the quantitative fine-tuning results, we encourage more comprehensive studies dedicated to human evaluation in the future.

5.2 Empirical Data Analysis

When observing natural and synthetic data, we focus on explaining two questions: (1) Why did the Malayalam-English baseline perform better than Spanish-English despite less training data? (2) Why did synthetic data improve Spanish-English performance while decreasing Malayalam-English performance in XLM-T? We find that the challenges in the dataset, task, and the training background of LLM best answer these questions.

5.2.1 Dataset Challenge

Aligning with the results of human evaluation, we found significant label ambiguity in the human-labeled Spanish-English dataset due to both the inherent ambivalence of human speech and the various interpretations that can be made by human annotators.

In Table 5, the first two examples highlight annotation ambiguity. Despite conveying similar ideas of anticipating Friday and listening to the Hispanic radio morning show "El Vacilón de la Gatita," they are labeled differently. Notably, the use of "Happy" in the first sentence seems to imply a positive sentiment but is labeled as neutral.

The subsequent examples illustrate disagreements between human evaluators and true labels. One example, a clothing ad with a seemingly positive connotation, could be interpreted as neutral due to its advertising context. Conversely, the second example, discussing hearing a spoiler for "The Bachelor," seems to clearly warrant a negative rather than positive label.

In contrast, the Malayalam-English dataset contains cleaner, more consistently phrased examples. A significant portion of negatively labeled sentences include the word "Dislike," simplifying the sentiment analysis task. This consistency likely contributes to Malayalam-English's high performance compared to Spanish-English. We provide further analysis of the challenges of CM sentiment analysis in Appendix E.

5.2.2 Data Parallels

We discovered many parallels between natural and synthetic Spanish-English data both semantically and syntactically. Shown in Table 6, both natural and synthetic sentences discuss common ideas, such as replaying a song, and use Spanish-English hybrid words like "textear." LLM's ability to adapt to the topics discussed in the Spanish-English data and to capture these CM nuances supports the high performance gains synthetic data provide.

Natural	Synthetic
Something came up algo surgió un problema **sorry something came up** and I cann't make it to the party	Can't believe I got stood up... Mi date dijo **"sorry, something came up"** like for real?!
Ojos verdes on **replayyy**	This song me tiene in my feels, **replay** x100
Estaba pensando en **textear**le a mi hermana y al minuto me llega un mensaje de ella #sisterlyconnection	Cuando te voy a **textear** y apareces typing, call it telepathy or just buena onda
Se me olvidaron todos los **passwords** del Hospital **y no podia entrar a** ningun lado #PerksDeLosFinales	UGH, olvidé mi **password** again **y no puedo entrar a** mi cuenta... FML
Deslike adicha ella punnara makkalkum nanni	Plot had potential, but execution polilla , disappointed.

Table 6: Comparisons of natural and synthetic sentences in Spanish-English (red) and Malayalam-English (yellow). Overlapping words or phrases are highlighted in bold. Translations of the CM sentences into English are provided in in Table 8 in Appendix A.

While the Spanish-English natural data frequently featured *alternational* CM patterns, where sentences alternated between languages, Malayalam-English natural data primarily exhibited *insertional* CM, where English words were occasionally inserted into predominantly Malayalam sentences. LLMs often generated alternational CM in Spanish-English and insertional CM with English as the dominant language in Malayalam-English. As a result, they improved performance in Spanish-English but did not align well with the

Malayalam-English natural dataset, where Malayalam was the dominant language. Even though LLM-generated Malayalam-English data sounded natural according to human evaluators, it reflected a real-world insertional CM pattern not present in our particular human-labeled dataset. As a result, this discrepancy highlights the inherent complexity of CM tasks for ML models due to the diverse nature of CM cultural practices.

A key challenge remains in controlling the type of CM—whether alternational or insertional—that LLMs produce. While LLMs handle alternation between English and Spanish with relative ease due to extensive training data, balancing languages like Malayalam and English remains a significant challenge. Consequently, the effectiveness of data augmentation is contingent not only the model's initial task performance but also the similarity between the CM patterns in natural and synthetic datasets.

5.3 Trade-offs with Using Synthetic Data

While our research demonstrates that LLMs can effectively generate CM training data, the key question is why we should prefer LLM-generated data over human-labeled data.

Collecting high-quality natural CM data is resource-intensive, involving web scraping, human annotation, and rigorous quality control. For instance, to create the Spanish-English SA dataset, Patwa et al. (2020) scraped CM data from Twitter, employed three Amazon Mechanical Turk[3] workers to label 18,789 tweets, and conducted manual reviews to correct errors. The estimated cost for annotating these tweets was approximately $3,054 USD, based on the minimum rate for Spanish-speaking workers.[4] A detailed cost breakdown is available in Appendix G.

Comparing the baseline scores on the full Spanish-English dataset to the subset in Section 4.2, adding ~9,000 human-labeled sentences to a baseline of 3,000 resulted in a **7.49%** improvement. According to the procedure above, the cost of these sentences was approximately **$1,495 USD**, and the annotation process likely took several weeks.

In contrast, generating synthetic data using GPT-4 for both Spanish-English and Malayalam-English, including preliminary experiments, cost $376.54 USD in total. Adding 50,000 synthetic sentences to the same baseline of 3,000 resulted in a **9.32%** improvement. These sentences were generated in hours and cost only **$37.92 USD**, making synthetic data generation 40 times cheaper than manual annotation of a corpus one-fifth the synthetic size.

While a larger volume of synthetic sentences is needed to achieve the same performance gains as a smaller set of human-labeled sentences, synthetic data generation is significantly more cost-effective and faster. Moreover, adding a large amount of synthetic data to natural data yields greater performance improvements (9.32%) than adding a smaller set of human-labeled data (7.49%).

6 Conclusion and Future Work

To address CM data scarcity, we propose using few-shot prompting with LLMs to generate synthetic, labeled CM data for SA. We tested this approach by training mBERT and XLM-T on natural, synthetic, and combined datasets for Spanish-English and Malayalam-English. In Spanish-English, our method improved sentiment classification by 9.32% for the 3k training setup and achieved third place on the LinCE benchmark for the 12k training setup. Human evaluations confirmed that our synthetic data closely mimic natural data and are indistinguishable from human-labeled examples. For Malayalam-English, our baseline system exceeded the highest published benchmark with an F1 score of 0.847, though further improvements with additional data were limited. Our findings indicate that LLM-generated synthetic data are most effective for enhancing models with low baseline performance, particularly when the languages are evenly represented as well as for resource-constrained scenarios. Overall, LLM-powered data augmentation offers a cost-effective alternative to human annotation, producing high-quality, natural-sounding sentences with minimal label ambiguity.

To improve performance in Malayalam-English, we intend to apply our observations of synthetic data to refine our LLM prompt and regenerate data. In addition, we aim to extend our research to encompass a broader range of LLMs and dialects, including those without English as a base and those primarily written in non-Latin scripts. Ultimately, our findings offer a promising avenue for CM data augmentation, and we encourage further exploration with LLMs in CM, an area which presents technical challenge and valuable social impact.

[3] https://www.mturk.com/

[4] Minimum rates for workers with premium qualifications are detailed here: https://requester.mturk.com/pricing

7 Limitations

The findings may not generalize across all types of data or tasks. While we find that results are generalizable across different PLMs such as mBERT and XLM-T and that LLMs typically generate natural-sounding sentences, the effectiveness of the data augmentation method may vary depending on the specific characteristics of the dataset, the resource level of the language, or the nature of the natural language processing task. Our experiments focused on Spanish-English and Malayalam-English for sentiment analysis, and we encourage future research to explore this method in other languages and tasks.

Furthermore, the effectiveness of this data augmentation method is limited by the baseline performance on natural data. If performance on natural data is already higher than the threshold synthetic data can raise results to, then further improvements are difficult to achieve. To mitigate this issue, an option is to regenerate synthetic data with an improved prompt, resulting in more natural synthetic data that can raise performance to an even higher threshold.

However, quickly quantifying the effectiveness of a prompt or strategy is challenging because it necessitates repeatedly generating large datasets and retraining models to measure performance improvements, which may become resource intensive if repeated numerous times. Furthermore, human evaluation was constrained to 200 and 400 data samples due to limited resources. In the future, developing a metric to quantify synthetic data quality without fine-tuning a separate model or using human evaluation would help streamline the development process and provide more direct insights.

Notably, there are data augmentation methods for CM SA other than Li and Murray (2022) and similar to our implementation of Random Translation, including Tareq et al. (2023), who convert a monolingual English corpus into Bangla-English using a word embedding algorithm, and Ma et al. (2020), who also randomly translate parts of a monolingual English corpus into Spanish-English. However, they either use different datasets, do not provide all baseline scores to be able to compare, do not detail their exact experiments, or do not release their code, so we were not able to directly compare our techniques with theirs.

8 Ethics Statement

Like most data augmentation methods, LLM-powered synthetic data generation raises ethical concerns because of its potential to magnify biases within datasets. Since multilingual NLP and CM are interlaced with people's identities, cultures, and heritages, it is important that LLMs do not misrepresent peoples' cultures and languages in offensive or inaccurate ways. As a result, we acknowledge the importance of working alongside qualified CM experts and including speakers familiar with the languages in CM patterns in the research process. Before deploying models to the public, it is vital that generated data is verified and CM language models are thoroughly tested.

9 References

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

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.

Yang Bai, Bangyuan Zhang, Wanli Chen, Yongjie Gu, Tongfeng Guan, and Qisong Shi. 2021. Automatic detecting the sentiment of code-mixed text by pre-training model. *Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Bharathi Raja Chakravarthi, Navya Jose, Shardul Suryawanshi, Elizabeth Sherly, and John Philip McCrae. 2020. A sentiment analysis dataset for code-mixed Malayalam-English. In *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*, pages 177–184, Marseille, France. European Language Resources association.

Ching-Ting Chang, Shun-Po Chuang, and Hung-Yi Lee. 2019. Code-switching sentence generation by generative adversarial networks and its application to data augmentation. *Preprint*, arXiv:1811.02356.

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. *Preprint*, arXiv:1911.02116.

Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Yihan Cao, Zihao Wu, Lin Zhao, Shaochen Xu, Wei Liu, Ninghao Liu, Sheng Li, Dajiang Zhu, Hongmin Cai, Lichao Sun, Quanzheng Li, Dinggang Shen, Tianming Liu, and Xiang Li. 2023. Auggpt: Leveraging chatgpt for text data augmentation. *Preprint*, arXiv:2302.13007.

Bosheng Ding, Chengwei Qin, Ruochen Zhao, Tianze Luo, Xinze Li, Guizhen Chen, Wenhan Xia, Junjie Hu, Anh Tuan Luu, and Shafiq Joty. 2024. Data augmentation using llms: Data perspectives, learning paradigms and challenges. *Preprint*, arXiv:2403.02990.

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.

Zulfadzli Drus and Haliyana Khalid. 2019. Sentiment analysis in social media and its application: Systematic literature review. *Procedia Comput. Sci.*, 161(C):707–714.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12).

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.

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.

Grandee Lee, Xianghu Yue, and Haizhou Li. 2019. Linguistically Motivated Parallel Data Augmentation for Code-Switch Language Modeling. In *Proc. Interspeech 2019*, pages 3730–3734.

Shuyue Stella Li and Kenton Murray. 2022. Language agnostic code-mixing data augmentation by predicting linguistic patterns. *Preprint*, arXiv:2211.07628.

Tianjian Li and Kenton Murray. 2023. Why does zero-shot cross-lingual generation fail? an explanation and a solution. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12461–12476, Toronto, Canada. Association for Computational Linguistics.

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.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *Preprint*, arXiv:1711.05101.

Yili Ma, Liang Zhao, and Jie Hao. 2020. XLP at SemEval-2020 task 9: Cross-lingual models with focal loss for sentiment analysis of code-mixing language. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 975–980, Barcelona (online). International Committee for Computational Linguistics.

Rana D. Parshad, Suman Bhowmick, Vineeta Chand Chand, Nitu Kumari, and Neha Sinha. 2016. What is india speaking? exploring the "hinglish" invasion. *Physica A: Statistical Mechanics and its Applications*, 449:375–389.

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.

Mary Phuong and Christoph H. Lampert. 2021. Towards understanding knowledge distillation. *Preprint*, arXiv:2105.13093.

Adithya Pratapa, Monojit Choudhury, and Sunayana Sitaram. 2018. Word embeddings for code-mixed language processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3067–3072, Brussels, Belgium. Association for Computational Linguistics.

Shruti Rijhwani, Royal Sequiera, Monojit Choudhury, Kalika Bali, and Chandra Shekhar Maddila. 2017. Estimating code-switching on Twitter with a novel generalized word-level language detection technique. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1971–1982, Vancouver, Canada. Association for Computational Linguistics.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. SemEval-2017 task 4: Sentiment analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518, Vancouver, Canada. Association for Computational Linguistics.

Thamar Solorio and Yang Liu. 2008. Learning to predict code-switching points. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 973–981, Honolulu, Hawaii. Association for Computational Linguistics.

R Srinivasan and C N Subalalitha. 2023. Sentimental analysis from imbalanced code-mixed data using machine learning approaches. *Distributed and Parallel Databases*, 41:37–52.

Mohammad Tareq, Md. Fokhrul Islam, Swakshar Deb, Sejuti Rahman, and Abdullah Al Mahmud. 2023. Data-augmentation for bangla-english code-mixed sentiment analysis: Enhancing cross linguistic contextual understanding. *IEEE Access*, 11:51657–51671.

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.

Ngoc Thang Vu, Dau-Cheng Lyu, Jochen Weiner, Dominic Telaar, Tim Schlippe, Fabian Blaicher, Eng-Siong Chng, Tanja Schultz, and Haizhou Li. 2012. A first speech recognition system for mandarin-english code-switch conversational speech. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4889–4892.

Chenxi Whitehouse, Monojit Choudhury, and Alham Aji. 2023. LLM-powered data augmentation for enhanced cross-lingual performance. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 671–686, Singapore. Association for Computational Linguistics.

Chenxi Whitehouse, Fenia Christopoulou, and Ignacio Iacobacci. 2022. EntityCS: Improving zero-shot cross-lingual transfer with entity-centric code switching. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6698–6714, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Genta Indra Winata, Andrea Madotto, Zhaojiang Lin, Rosanne Liu, Jason Yosinski, and Pascale Fung. 2021. 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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi 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 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.

Haoran Xu, Seth Ebner, Mahsa Yarmohammadi, Aaron Steven White, Benjamin Van Durme, and Kenton Murray. 2021. Gradual fine-tuning for low-resource domain adaptation. In *Proceedings of the Second Workshop on Domain Adaptation for NLP*, pages 214–221, Kyiv, Ukraine. Association for Computational Linguistics.

Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. 2024. A survey on knowledge distillation of large language models. *Preprint*, arXiv:2402.13116.

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 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.

Kang Min Yoo, Dongju Park, Jaewook Kang, Sang-Woo Lee, and Woomyeong Park. 2021. Gpt3mix: Leveraging large-scale language models for text augmentation. *Preprint*, arXiv:2104.08826.

Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. 2023. Large language model as attributed training data generator: A tale of diversity and bias. *Preprint*, arXiv:2306.15895.

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

A Translations of Tables and Figures

This section provides translations of the CM sentences used in Figure 1 and in the tables in Section 5.2. Figure 3 is a translated version of Figure 1, Table 7 is a translated version of Table 5, and Table 8 is a translated version of Table 6.

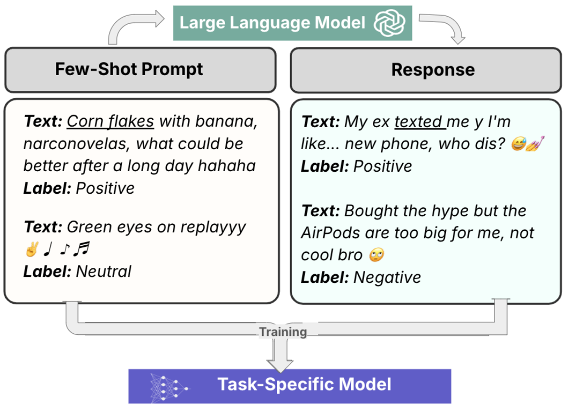

Figure 3: Overall system workflow with translated examples of Spanish-English CM tweets as natural data (left) and synthetic data (right). Underlined words represent Spanish-English hybrid words, examples of the complexities introduced by CM.

Sentences	**Label**	**Prediction**
Happy Friday #thejokeoftheKitten	neutral	positive
#thejokeoftheKitten #quotes #friday	positive	neutral
Sentences	**Label**	**Correction**
Get your outfit now! Choose your favorite garments and place your order Purple blouse $20.00 #ilovesalhuaclothing	neutral	positive
How I find out who's leaving on The Bachelor without having seen the episode Angry Face	positive	negative

Table 7: Translated examples of sentences from the natural Spanish-English dataset, including their true labels, XLM-T's predicted labels, and the proposed corrections by human evaluators.

Natural	**Synthetic**
Something came up something came up a problem **sorry something came up** and I cann't make it to the party	Can't believe I got stood up... Mi date dijo **"sorry, something came up"** like for real?!
Green eyes on **replayyy**	This song has me in my feels, **replay** x100
I was thinking about **textingmy** sister and a minute later I get a message from her #sisterlyconnection	When I'm going to **text** and you show up typing, call it telepathy or just good vibes
I forgot all the **passwords** of the Hospital **and I couldn't enter** anywhere #PerksoftheFinals	UGH, I forgot my **password** again **and I cannot enter** my account... FML
Deslike adicha ella punnara makkalkum nanni	Plot had potential, but execution polilla , disappointed.

Table 8: Comparisons of translated natural and synthetic sentences in Spanish-English (red) and Malayalam-English (yellow). Overlapping words or phrases are highlighted in bold. The Malayalam-English data are not translated due to its low-resource nature and the lack of available translators.

B Implementation Details

B.1 Data Generation Details

Table 4 displays our prompt-tuning process, where we iteratively improved on our data generation prompt to the LLM. For all experiments, we prompted `gpt-4-1106-preview` with the OpenAI library, with temperature 0.6. For Spanish-English, we varied the number of shots `m` between {15, 50, 150, 500} given in our prompt with the objective to find optimal shot size. To overcome the maximum sequence length, we instructed GPT-4 to generate 50 data points and automatically repeated this process until we reached our desired dataset size. For each iteration, the prompt contained newly randomly-sampled task demonstrations from the training data. We did not post-filter the data due to its size and subjectivity. Our total synthetic data sizes were ~53000 in Spanish-English and ~24000 in Malayalam-English.

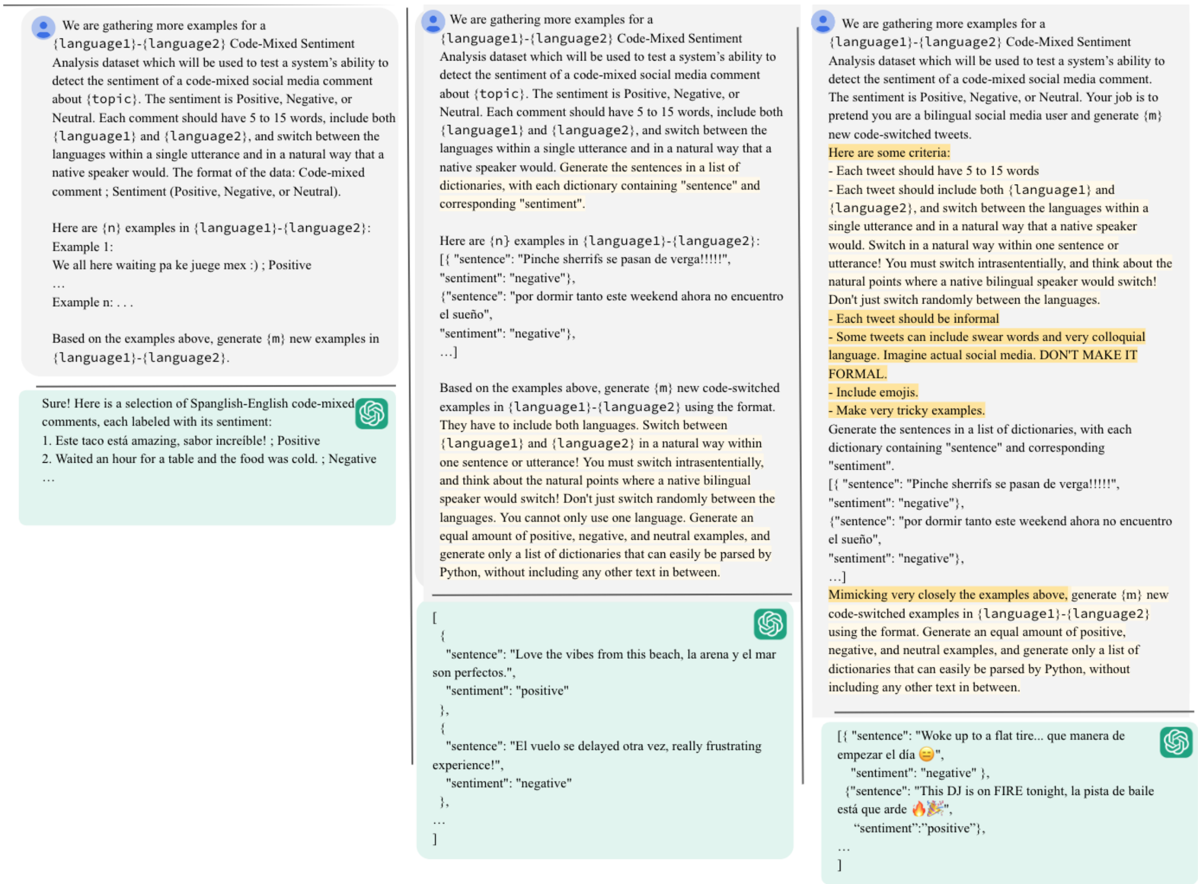

Figure 4: Prompt-tuning process, showing system input in gray, LLM sample output in teal, and iterative improvements made to our prompt highlighted in yellow. Our final prompt is shown to the right.

B.2 Fine-tuning Details

Chosen based on Li and Murray (2022)'s experiments, in our gradual fine-tuning approach, the synthetic data sizes were {50000, 25000, 15000, 5000, 0}, and each stage included 3 epochs. For all experiments, we used the Transformers library (Wolf et al., 2020) to fine-tune XLM-T with a task-specific classification layer using AdamW (Loshchilov and Hutter, 2019) optimizer. According to the hyperparameters of the dataset benchmark (Patwa et al., 2020; Aguilar et al., 2020) and our empirical experiments involving hyperparameter grid search, we set the highest sequence length at 40 tokens, batch size 32, weight decay 0.01, learning rate $5e^{-5}$, and epsilon $1e^{-8}$. For gradual fine-tuning, the learning rates used were $\{1e^{-6}, 2e^{-6}, 2e^{-6}, 4e^{-6}, 2e^{-6}\}$, determined through preliminary experimentation and standard grid search. We also tuned additional hyperparameters including synthetic data size, shot size, and temperature based on a standard grid search. Experiments ran on a 16GB T4 GPU.

Language	Training	F1 Score
Spanish-English	1-Stage	0.595
	5-Stage	0.603
Malayalam-English	1-Stage	0.843
	5-Stage	0.718

Table 9: Comparison of F1 scores when XLM-T is fine-tuned with one stage and with five stages for each language.

C Impact of Gradual Fine-tuning

Table 9 compares F1 score for one stage of training to five stages of training using gradual fine-tuning for Spanish-English and Malayalam-English. Results marginally increase for Spanish-English while decreasing for Malayalam-English. This may be due to less suitable hyperparameters used in five stage training in comparison to one stage.

D Instructions for Human Evaluation

Two native Spanish-English bilingual students, who did not have knowledge of the rest of the experimentation, were each given the same 100 code-mixing texts and corresponding labels, 50 of which were randomly sampled from the natural training data and 50 of which were randomly sampled from the synthetic data. They did not know which were natural or synthetic, as the sentences were scrambled in random order. One native Malayalam-English bilingual speaker was given 400 code-mixing texts and corresponding labels, 200 of which were randomly sampled from natural training data and 200 of which were randomly sampled from synthetic data.

Our first Spanish-English annotator was a balanced bilingual with C2 proficiency in both English (native language) and Spanish (second language). Our second Spanish-English annotator was a balanced bilingual with C2 proficiency in Spanish (native language) and C1 proficiency in English (second language). The Malayalam-English annotator was a balanced bilingual with C2 proficiency in both Malayalam (native language) and English (second language). All annotators reported to use both languages frequently in their daily lives. The initial instructions given were:

> You have been provided with a spreadsheet containing social media comments that are intended to be code-mixed in Spanish and English, though some may not be. Each comment is labeled with a sentiment—'Positive,' 'Negative,' or 'Neutral.' Your task is to evaluate each comment based on the following criteria: Read the Sentence: Carefully review each comment. Fill Out Ratings: Code-mixing Naturalness: Evaluate how naturally the comment switches between Malayalam and English. Label Accuracy: Assess whether the sentiment label ('Positive,' 'Negative,' or 'Neutral') accurately reflects the comment's connotation. If you disagree with the label, you must provide an alternative in the 'If you answered "Disagree", what would you label it?' column. Human or Machine: Determine whether the comment was written by a human or generated artificially by a machine. Additional Comments (Optional): If you have further observations or concerns, please record them in the 'Additional comments' field. Keep in mind: Code-mixing refers to the blending of two or more languages in speech. These comments are sourced from social media, so they may be informal, include emojis, or contain spelling errors. If you are uncertain about your evaluation, choose the most likely option and note your concerns in the comments. Please ensure that your evaluations are accurate and consistent across the dataset.

For "Code-mixing Naturalness," they were given the description:

> Evaluate naturalness on the changing between Spanish and English. Choose between the options: "This sounds natural, like something people would actually type/say," "This sounds a bit strange/could be improved," and "This sounds unnatural/needs to be rewritten." Do not consider naturalness/strangeness of the topics discussed. Do not consider grammar/spelling mistakes unless they are extreme. Do not consider the label.

It is important to note that only the first option for CM naturalness is counted as "natural" while the "strange" and "unnatural" classifications are grouped into an omnibus "unnatural" category. For "Label Accuracy," they were given the description:

> Would you agree with the label associated with each sentence? Is a sentence labeled "positive" actually giving positive connotations? Answer with "Agree" or "Disagree."

For "Human or Machine-Generated," they were given the description:

> Do you think a human wrote this or a machine wrote this? Now you can consider any and all aspects e.g. fluidity, topics, mechanics, anything.

Additionally, evaluators are given the option to correct labels for which they disagreed with and to leave additional comments.

E Case Study on Laughter

We investigated the use of "jajaja," shown in Table 10, the Spanish version of typing laughter, which

Index	Sentences	True Label	Predicted Label
1	jajajaj okay okay ill wait and give them to you on valentines day so it can be your cheat day	positive	positive
2	I can imagine jajaja	positive	positive
3	most likely jajajaj	positive	positive
4	Girrrl I wish I had your self-esteem jaja	neutral	positive
5	Jajajajajajajajajajajajaja ok ok ok	neutral	positive
6	jovanigram's video JAJAJAJAJAJ	neutral	positive
7	tb to your birthday :') jajajaja	neutral	positive
8	Whattt Frowning Face with Open Mouth #forever-riendome jajajjajjajaj	neutral	negative

Table 10: Examples of natural sentences including laughter in the test data, with true labels and predicted labels.

occurred frequently in both natural and synthetic data and can have positive, neutral, or negative connotations.

This case study demonstrates the challenges of CM sentiment analysis in that 1) human labels are sometimes ambiguous, 2) sentences are short, 3) the model predictions may be biased toward the positive label, and 4) emojis and symbols play an important role. Examples of ambiguity are in sentences 1 and 2, which could also be considered neutral, since sentences 5 and 6 are neutral. Sentences 2, 3, and 5 also contain very little information as compared to sentence 1, which the model had correct and shows understanding despite sentence 1's complexity. We also observe almost all positive predictions to the class imbalance as described in Section 5, where it is the model's mistake and there is fairly little ambiguity like sentence 4. For sentence 6, the model may not realize ":')" refers to a crying happy face and errs. On the other hand,for sentence 8, "Frowning Face with Open Mouth" is the English description of the emoji from the original tweet, which likely led the model to respond with negative. The change from emoji to description may also be a factor in performance worth future exploration.

F Generated Sentences about Code-Mixing

Table 11 presents an intriguing observation: when asked to generate code-mixed sentences, many of the sentences ended up being about code-mixing or code-switching. In the CM sentences it was asked to generate, no theme was specified, yet out of 12865 sentences, 9 mention "code-switching," 40 mention "bilingual," 162 mention "Spanish-English," and 5 mention "French," and all discuss being skilled or having fun at code-switching. Perhaps LLM has developed somewhat of a personality, or perhaps this is due to the input instructions.

G Cost Analysis of Data Collection

G.1 Natural Data

To estimate the cost incurred by Patwa et al. (2020) of annotating 18,789 tweets using Amazon Mechanical Turk (MTurk), we first determine the number of HITs (Human Intelligence Tasks) required. Each HIT includes 10 tweets, but only 8 are for annotation purposes, with 2 serving as quality control. Thus, to annotate 18,789 tweets, we need approximately 2,349 HITs. To hire workers fluent in Spanish, HITs are required to be priced at at least $1.00 per HIT.[5] The total cost would then be computed as follows: 2,349 HITs multiplied by $1.00 per HIT results in a total cost of $2,349 USD. This estimate assumes that each HIT is completed by a single annotator and does not account for additional costs related to rejected assignments or quality control beyond the base HIT price.

Estimating the additional costs related to rejected assignments, if 30% of all assignments were rejected and reassigned, the total cost would increase to $3,054 USD.

These calculations use the case of Patwa et al. (2020), but it is important to consider that other works generally require more than one annotator to label each data point. Then, the previously calculated costs would double or triple depending on the number of annotators. Furthermore, Patwa et al. (2020) do not release their exact price per HIT or the number of reassigned assignments, so there is

[5] https://requester.mturk.com/pricing

Sentences	English Translation	Labels
Creo que I finally got the hang of esto de code-switching, it's kinda fun!	I think I finally got the hang of code-switching, it's kinda fun!	positive
¿Does this count como un code-switched tweet? Asking for a friend	Does this count as a code-switched tweet? Asking for a friend	neutral
Ya no sé if I should hablar español o inglés, my brain is too code-switchy today	I still don't know if I should speak Spanish or English, my brain is too code-switchty today	neutral
Random pero I started learning French y ahora mezclo three languages, send help	Random but I started learning French and already mix three languages, send help	neutral

Table 11: Examples of synthetic sentences mentioning CM explicitly, their translations, and their labels. Red text is in Spanish.

high variability. Increased prices per HIT could increase costs significantly.

G.2 Synthetic Data

For generating synthetic data, we made requests to GPT-4 to generate 50 data points at a time. The purpose was to overcome the model's maximum sequence length. In the future, cost can be further reduced due to increasing maximum sequence length in LLMs.

To estimate the cost of generating 50,000 synthetic samples using GPT-4, we first determine the total number of tokens per request. Each request includes a prompt of 330 tokens and 15 data examples, each averaging 20.8 tokens, totaling 642 tokens for the prompt and examples. GPT-4 then generates 50 samples, each averaging 21 tokens, resulting in 1,050 tokens for the generated samples. Therefore, each request utilizes a total of 1,692 tokens. To generate 50,000 samples, we need to make 1,000 requests, resulting in a total of 1,692,000 tokens. Given GPT-4 pricing, which is $10.00 per 1 million input tokens and $30.00 per 1 million output tokens,[6] we can calculate the costs as follows: For the 642,000 input tokens, the cost is $6.42, while for the 1,050,000 output tokens, the cost is $31.50. Thus, the total cost for generating 50,000 samples is approximately $37.92.

[6] https://openai.com/api/pricing/