Exploring the Capabilities of Pre-Trained Language Models in Code-Switching Sentiment Analysis

Jih-Kang Hsieh*, Chia-Hua Wu†, and Hsin-Min Wang†
* Department of Computer Science and Information Engineering, National Taiwan University, Taiwan
E-mail: r13922a03@csie.ntu.edu.tw
† Institute of Information Science, Academia Sinica, Taipei, Taiwan
E-mail: {maxwu, whm}@iis.sinica.edu.tw

***Abstract*—Pre-trained BERT-type language models have greatly boosted the accuracy of various natural language processing (NLP) tasks in recent years. However, their effectiveness in code-switching NLP tasks remains unclear. This paper focuses on code-switching sentiment analysis. We compare five BERT-type multilingual language models and evaluate the effect of data augmentation using bilingual and synthetic code-switching data. In addition, we analyze whether the language alignment considered in pre-training affects the performance of pre-trained multilingual language models in code-switching sentiment analysis tasks. Experimental results on the LinCE Spanish-English dataset show that LaBSE performs best, while mBERT performs worst. The code and models are available at GitHub[1].**

***Index Terms*—Code-switching, sentiment analysis, pre-trained language models, data augmentation, language alignment.**

I. Introduction

Code-switching is a linguistic phenomenon in which multilingual speakers use multiple languages simultaneously in a sentence. It is common in multilingual regions such as Mexico, India, and Singapore, and is also common in social media, professional fields, and spoken language. However, due to the lack of training data, most models tend to perform poorly on code-switching tasks. Even large language models with multilingual capabilities suffer from reduced performance when faced with code-switching [1].

One approach to addressing the code-switching problem is to utilize large amounts of monolingual data to assist in the code-switching task [2], [3]. As a result, many researchers have worked on synthesizing code-switching data from monolingual data, including rule-based generation [4], [5], random word translation [6], [7], modifying model architectures specifically for code-switching [8], [9], [10], leveraging generative models [11], [12], [13], [14], and even directly prompting large language models to generate synthetic data [15]. However, these approaches typically require large amounts of parallel data in two or more languages [16], [17], [18] or generate data of varying quality that require manual inspection.

In addition to the work on data synthesis, the study in [19] showed that bringing the embeddings of two languages closer in the representation space can benefit code-switching language modeling using only monolingual data. Unfortunately, the potential benefits of language alignment for code-switching tasks have not received enough attention since then.

[1]https://github.com/stephenjkhsieh/ECLM-CSSA

In recent years, pre-trained language models have greatly improved the accuracy of various natural language processing (NLP) tasks. One of the most successful models is BERT [20], which is pre-trained using masked language modeling (MLM) and next sentence prediction (NSP). Although more advanced multilingual language models than BERT have been proposed, such as multilingual BERT [20], XLM-RoBERTa [21], LaBSE [22], and MPNet [23], their effectiveness in code-switching NLP tasks remains unclear. Therefore, we study their effectiveness in the code-switching sentiment analysis (SA) task.

We are interested in how models with different pre-training methods perform on the code-switching SA task. Specifically, we study the impact of language-independent sentence embeddings on the code-switching domain. In addition, we also study the effect of augmenting the training data for the code-switching SA task with bilingual and synthetic code-switching data in both reasonable-resource and low-resource settings.

Our contributions are as follows.

- We evaluate the performance of multilingual language models with different pre-training methods on the code-switching SA task.
- We analyze the effectiveness of language-independent sentence embeddings of pre-trained multilingual language models in fine-tuning these models to the code-switching sentiment analysis task.
- We provide visualizations of feature representations to explain the experimental results.

II. Methodology

A. Pre-trained Language Models

In this study, we investigate the performance of five pre-trained language models in the form of Transformer-based encoders [24] on the code-switching SA task, including the vanilla multilingual language models multilingual BERT (mBERT) [20] and XLM-RoBERTa (XLM-R) [21], as well as the language-aligned multilingual language models LaBSE [22], smaller-LaBSE ($\text{LaBSE}_{\text{smaller}}$) [25] and MPNet-SBERT[2].

[2]https://huggingface.co/sentence-transformers/paraphrase-multilingual-mpnet-base-v2

Table I lists the number of parameters and the number of supported languages for each model. Note that the five pre-trained language models are of the same size. The differences in the number of model parameters come from the different vocabulary sizes.

TABLE I: Specifications of the five pre-trained Transformer-based language models used in this paper. **P** denotes the total number of parameters, $\mathbf{P_E}$ refers to the number of parameters in the Transformer encoder, and **Languages** indicates the number of supported languages.

Model	**P**	$\mathbf{P_E}$	**Languages**
mBERT	178M	85M	104
$\text{LaBSE}_{\text{smaller}}$	219M	85M	15
XLM-R	278M	85M	94
MPNet-SBERT	278M	85M	53
LaBSE	471M	85M	109

Among language-aligned models, MPNet-SBERT adopts a student-teacher framework, where the student model aligns multilingual sentence embeddings to a monolingual teacher (typically English), effectively mapping all languages into the English embedding space. In contrast, LaBSE uses contrastive learning to pull parallel sentences across languages closer and push non-parallel ones apart, constructing a shared multilingual embedding space independently, without relying on English as an anchor. MPNet-SBERT is recognized as one of the best-performing models within the student-teacher approach.

B. Sentiment Classification Models

To enable a pre-trained BERT-type language model to perform SA tasks, its output [CLS] token (which provides aggregate sequence representation) is integrated into a linear layer. The model training process involves optimizing the parameters of the pre-trained BERT-type model and the linear layer to ensure that the entire network can effectively distinguish the three sentiment classes. The training process uses cross-entropy loss and updates the model parameters with the AdamW optimizer.

III. Datasets

A. Code-Switching Data

LinCE [26] is a Linguistic Code-switching Evaluation benchmark. We conducted experiments using LinCE's Sentiment Analysis (SA) dataset [26]. It contains 18,800 (12,200 for training, 1,800 for development, and 4,740 for testing) Spanish-English code-switched tweets labeled negative, neutral, or positive. Since the test set labels are not publicly available, we use the development set to evaluate the model performance.

In addition, we used GPT-4 [27] to translate the test set into a single language, thereby building a parallel corpus of code-switched and monolingual data for visualization and alignment analysis.

B. Bilingual Data

We sampled 38,000 (30,000 for training, 4,000 for development, and 4,000 for testing) Spanish and English instances from the open dataset for sentiment analysis[3] to construct a bilingual dataset, excluding instances labeled "none" and ensuring a balanced distribution across the three SA categories.

C. Synthetic Code-Switching Data

We also synthesized code-switching data from the bilingual dataset. We used a simple but effective method: for each word, there was a 0.3 probability of replacing it with a translation in the MUSE [28] dictionary. If the word did not exist in the dictionary, it remained unchanged.

IV. Experimental Setup

A. Training Data

To train the SA models, we used several data configurations. The baseline setting contained 12,000 code-switched utterances. In one configuration, we augmented it with 30,000 bilingual utterances. Another setting combined the same code-switching data with 15,000 bilingual utterances and 15,000 synthetic code-switching utterances. We also tested a setting where the code-switching data was supplemented with 30,000 synthetic code-switching utterances. In addition to the reasonable-resource scenario, we also simulated a low-resource condition, where only 10% of the code-switching data (1,200 utterances) was used for training.

B. Training Configuration

For each experimental setting, we trained five models for each pre-trained language model, using different random seeds to initialize the classifier and different shuffles of the training data. Each model was trained for 8 epochs, and the best performance on the development set was recorded. For each setting, we report the average performance of five runs. The learning rate was linearly decayed from 2e-5 to 0, and the batch size was fixed to 16. All experiments were performed using NVIDIA 3090 or NVIDIA 2080Ti GPUs.

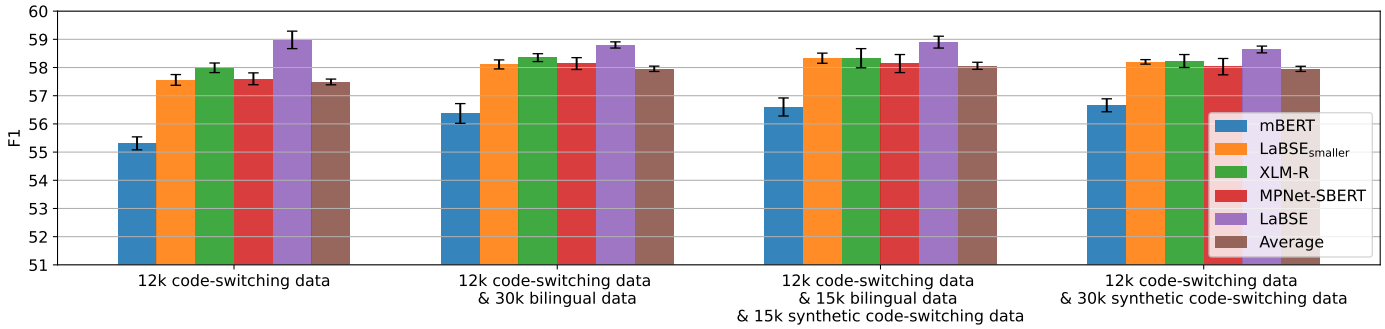
C. Language Representation Alignment Analysis

To evaluate the alignment between code-switching and monolingual sentence representations, we sampled 800 code-switching sentences from the LinCE test set and used GPT-4 to generate their corresponding Spanish monolingual translations with a temperature of 0.2. For each sentence pair $(s_i, \hat{s}_i)$, we calculated the cosine similarity of their corresponding representations $\mathbf{h}_i$ and $\hat{\mathbf{h}}_i$.

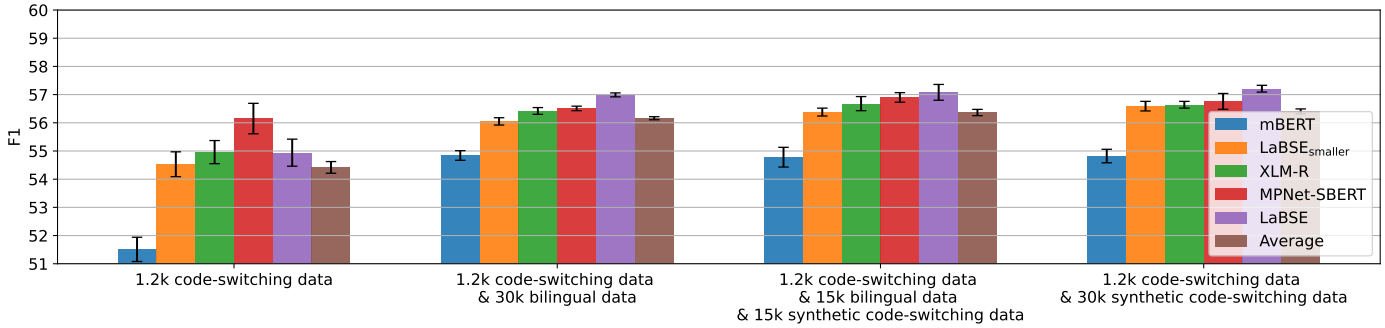
To address potential distortion caused by anisotropy in the representation space [29], all 1600 representations were normalized to have zero mean before computing the cosine similarity:

$$sim_i = \frac{\mathbf{h}_i^{norm} \cdot \hat{\mathbf{h}}_i^{norm}}{\max\{\|\mathbf{h}_i^{norm}\| \cdot \|\hat{\mathbf{h}}_i^{norm}\|, \epsilon\}}, \quad (1)$$

[3]https://github.com/charlesmalafosse/open-dataset-for-sentiment-analysis



(a) Models trained using 12k code-switching data and different data augmentation settings (reasonable-resource scenario).



(b) Models trained using 1.2k code-switching data and different data augmentation settings (low-resource scenario).

Fig. 1: Weighted F1 performance of different SA models under different training data settings, reported as the mean over five random seeds with error bars indicating ±1 standard deviation.

where $\mathbf{h}_i^{norm}$ and $\hat{\mathbf{h}}_i^{norm}$ denote the normalized representations, and ϵ is a small constant to avoid division by zero.

We computed the average cosine similarity across all 800 sentence pairs for each model and training configuration. The results are shown in Figure 2.

V. Experimental Results

A. Performance of SA Models under Different Training Data Settings

Figure 1 shows the performance of five SA models under different training data conditions. In the reasonable-resource scenario, data augmentation with bilingual or synthetic code-switching data generally improves the performance of all models except LaBSE. In the low-resource scenario, data augmentation significantly improves the performance of all models, with synthetic code-switching data providing the most significant improvement. However, in the reasonable-resource scenario, data augmentation with synthetic code-switching data does not bring significant improvement, indicating that sufficient code-switching data reduces the incremental benefit of synthetic data. Overall, LaBSE-based models perform best, while mBERT-based models lag behind. Considering both model size and performance, LaBSE$_{\text{smaller}}$ is recommended because it can effectively utilize synthetic code-switching data. Furthermore, MPNet-SBERT shows excellent performance without any external data augmentation in the low-resource scenario. Therefore, MPNet-SBERT is also recommended in the extremely low-resource scenario.

B. Comparison of XLM-R, MPNet-SBERT, and LaBSE

We first compare XLM-R and MPNet-SBERT. MPNet-SBERT incorporates language alignment during pre-training, while XLM-R does not. As shown in Figure 1, MPNet-SBERT with language alignment outperforms XLM-R in the low-resource scenario. However, XLM-R outperforms MPNet-SBERT with sufficient code-switching data. This suggests a disconnect between general NLP capabilities and code-switching handling, likely influenced by language alignment. XLM-R excels in general NLP tasks, but struggles with code switching due to the lack of language alignment during pre-training; MPNet-SBERT incorporates language alignment in the pre-training stage, so it performs better in code-switching tasks, but its NLP capabilities are limited.

LaBSE stands out, especially when provided with sufficient code-switching data or when augmented with bilingual or synthetic code-switching data. As reported in [30], while both LaBSE and MPNet-SBERT are powerful language-independent sentence embeddings models, MPNet-SBERT focuses more on semantic representation. We argue that LaBSE achieves more subtle alignments than MPNet-SBERT, suggesting that its code-switching knowledge is deeper, but its NLP orientation is weaker due to contrastive learning. In the low-resource scenario, this lack of NLP knowledge leads to inferior performance. However, as the amount of available data improves its NLP knowledge, LaBSE's code-switching expertise enables it to outperform other models.

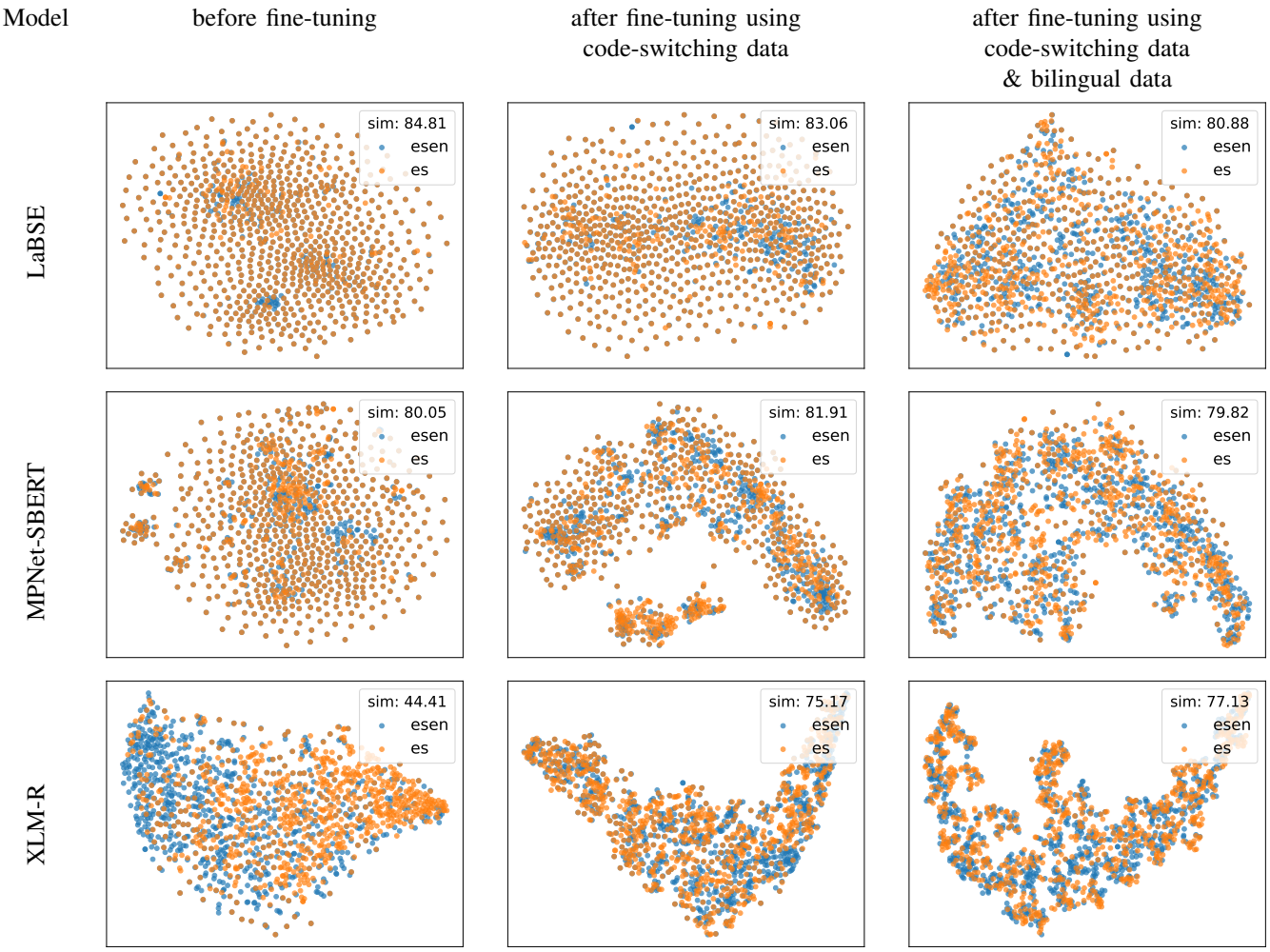


Fig. 2: T-SNE visualization of the representation space of Spanish-English code-switching sentences (esen: blue dots) and their corresponding GPT-4-generated Spanish translations (es: orange dots) under different training data and model settings. Since the orange dots are superimposed on the blue dots with semi-transparency, higher density of blue dots and higher saturation of orange dots indicate poorer alignment in the representation space. The legend shows the average cosine similarity for each setting to quantify the degree of alignment.

C. Visualization and Alignment Analysis

Figure 2 shows that before fine-tuning, LaBSE and mMPNet outperform XLM-R in alignment, ranking as follows: LaBSE $>$ mMPNet $>$ XLM-R. After fine-tuning on code-switching data, XLM-R and mMPNet form distinct clusters for the three classes (positive, negative, and neutral), but this pattern is less obvious for LaBSE. However, with the addition of bilingual data, LaBSE also forms a three-cluster pattern after fine-tuning. This pattern may represent task-oriented characteristics. This supports our hypothesis that mMPNet is more NLP-oriented than LaBSE, but less so than XLM-R. We also observe that fine-tuning disrupts the initial alignment properties, although LaBSE maintains its alignment better than mMPNet. This can be attributed to the more subtle alignment of LaBSE during pre-training, which helps preserve language alignment knowledge after fine-tuning.

TABLE II: Comparison of LaBSE and state-of-the-art models on the LinCE test set benchmark. * [31], † [32], ‡ [33], § [34].

Model	Weighted F1
GPT-$4_{\text{zero-shot}}$ *	54.60
Char2subword†	59.07
XLM-$T_{\text{LLM-Generated}}$‡	60.28
RoBERTuito§	60.57
LaBSE (our)	61.62
Anonymous	62.21

D. Comparison with State-of-the-Art Models

As shown in Table II, despite only simple fine-tuning, LaBSE still outperforms most other state-of-the-art models that rely on more complex techniques. Notably, it ranks second on the LinCE benchmark (https://ritual.uh.edu/lince/leaderboard),

even surpassing the well-known GPT-4 model, and only behind an anonymously submitted model. These results highlight the effectiveness of language-aligned models like LaBSE in the task of code-switching sentiment analysis.

VI. Conclusions and Future Work

In this paper, we have studied the effectiveness of five multilingual BERT-type pre-trained language models on the code-switching sentiment analysis task. We have also evaluated the effect of data augmentation using bilingual and synthetic code-switching data. Overall, LaBSE performs the best among the five models compared, while mBERT performs the worst. Considering the trade-off between model size and performance, $\text{LaBSE}_{\text{smaller}}$ is a good choice because it performs well when trained with data augmentation using synthetic code-switching data and its model size is the second smallest among the five models. We have also analyzed whether the language alignment considered in pre-training affects the model performance on the code-switching sentiment analysis task.

Potential future directions include exploring representation space mapping methods for implementing language-aligned models [35], and combining language-aligned and task-oriented losses during fine-tuning to encourage models to exhibit both alignment and task-specific capabilities. In addition, using techniques such as task vectors [36] can enable models to have both alignment properties and strong performance on code-switching NLP tasks.

Limitations

This study explores the effectiveness of five pre-trained multilingual language models for code-switching sentiment analysis, focusing on encoder-based models due to their wider availability and diversity in multilingual tasks where sentence-level embeddings are applied. While we include several state-of-the-art models (e.g., zero-shot GPT-4) in our evaluation to broaden the scope of our study, we acknowledge that our analysis does not comprehensively cover decoder-based architectures or other large-scale, instruction-tuned models (e.g., LLaMA and PaLM).

In addition, the performance of multilingual language models on code-switching tasks may be affected by various factors not fully explored in this study, such as domain-specific pre-training, fine-tuning strategies, or language-specific tokenization techniques. Our focus on the LinCE Spanish-English dataset also limits the generalizability of our findings to other language pairs or code-switching contexts. Therefore, while our results provide a useful benchmark, future research should explore a wider range of models across different datasets and tasks to better evaluate their applicability to code-switching NLP tasks.

References

[1] R. Zhang, S. Cahyawijaya, J. C. B. Cruz, et al., "Multilingual Large Language Models Are Not (Yet) Code-Switchers," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2023.

[2] Ö. Çetinoğlu, S. Schulz, and N. T. Vu, "Challenges of Computational Processing of Code-Switching," in *Proceedings of the Workshop on Computational Approaches to Code Switching*, 2016.

[3] T. Nguyen, N. Tran, L. Deng, et al., *Optimizing Bilingual Neural Transducer with Synthetic Code-switching Text Generation*, arXiv:2210.12214, 2022.

[4] A. Pratapa, G. Bhat, M. Choudhury, et al., "Language Modeling for Code-Mixing: The Role of Linguistic Theory based Synthetic Data," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2018.

[5] K. Gupta, C. Prachaseree, T. N. Ho, et al., "Singaporean Conversational English-Malay Code-Switching Speech: An Analysis Based on Code-switching Points and Part -of-Speech," in *Proceedings of the International Conference on Asian Language Processing*, 2023.

[6] F. Wang, K.-H. Huang, K.-W. Chang, et al., "Self-Augmentation Improves Zero-Shot Cross-Lingual Transfer," in *Proceedings of the International Joint Conference on Natural Language Processing and the Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, 2023.

[7] C. Du, H. Li, Y. Lu, et al., "Data Augmentation for end-to-end Code-Switching Speech Recognition," in *Proceedings of the IEEE Spoken Language Technology Workshop*, 2021.

[8] G. I. Winata, S. Cahyawijaya, Z. Liu, et al., "Are Multilingual Models Effective in Code-Switching?" In *Proceedings of the Workshop on Computational Approaches to Linguistic Code-Switching*, 2021.

[9] G. Aguilar and T. Solorio, "From English to Code-Switching: Transfer Learning with Strong Morphological Clues," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2020.

[10] R. Das, S. Ranjan, S. Pathak, et al., "Improving Pretraining Techniques for Code-Switched NLP," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2023.

[11] I. Tarunesh, S. Kumar, and P. Jyothi, "From Machine Translation to Code-Switching: Generating High-Quality Code-Switched Text," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, 2021.

[12] K. R. Chandu and A. W. Black, "Style Variation as a Vantage Point for Code-Switching," in *Proceedings of the Annual Conference of the International Speech Communication Association*, 2020.

[13] Y. Gao, J. Feng, Y. Liu, et al., "Code-Switching Sentence Generation by Bert and Generative Adversarial Networks," in *Proceedings of the Annual Conference of the International Speech Communication Association*, 2019.

[14] C.-T. Chang, S.-P. Chuang, and H.-Y. Lee, "Code-Switching Sentence Generation by Generative Adversarial Networks and its Application to Data Augmentation," in *Proceedings of the Annual Conference of the International Speech Communication Association*, 2019.

[15] Z. X. Yong, R. Zhang, J. Forde, et al., "Prompting Multilingual Large Language Models to Generate Code-Mixed Texts: The Case of South East Asian Languages," in *Proceedings of the Workshop on Computational Approaches to Linguistic Code-Switching*, 2023.

[16] G. Lee and H. Li, "Modeling Code-Switch Languages Using Bilingual Parallel Corpus," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2020.

[17] G. I. Winata, A. Madotto, C.-S. Wu, et al., *Learn to Code-Switch: Data Augmentation using Copy Mechanism on Language Modeling*, arXiv:1810.10254, 2018.

[18] G. I. Winata, A. Madotto, C.-S. Wu, et al., "Code-Switched Language Models Using Neural Based Synthetic Data from Parallel Sentences," in *Proceedings of the Conference on Computational Natural Language Learning*, 2019.

[19] S.-P. Chuang, T.-W. Sung, and H.-y. Lee, "Training Code-Switching Language Model with Monolingual Data," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2020.

[20] J. Devlin, M.-W. Chang, K. Lee, et al., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019.

[21] A. Conneau, K. Khandelwal, N. Goyal, et al., "Unsupervised Cross-lingual Representation Learning at Scale," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2020.

[22] F. Feng, Y. Yang, D. Cer, et al., "Language-agnostic BERT Sentence Embedding," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2022.

[23] K. Song, X. Tan, T. Qin, et al., "MPNet: Masked and Permuted Pre-training for Language Understanding," in *Proceedings of Advances in Neural Information Processing Systems*, 2020.

[24] A. Vaswani, N. Shazeer, N. Parmar, et al., "Attention is All you Need," in *Proceedings of Advances in Neural Information Processing Systems*, 2017.

[25] A. Abdaoui, C. Pradel, and G. Sigel, "Load What You Need: Smaller Versions of Multililingual BERT," in *Proceedings of the Workshop on Simple and Efficient Natural Language Processing*, 2020.

[26] G. Aguilar, S. Kar, and T. Solorio, "LinCE: A Centralized Benchmark for Linguistic Code-switching Evaluation," in *Proceedings of the Language Resources and Evaluation Conference*, 2020.

[27] J. Achiam, S. Adler, S. Agarwal, et al., *GPT-4 Technical Report*, arXiv:2303.08774, Mar. 2024.

[28] G. Lample, A. Conneau, M. Ranzato, et al., "Word translation without parallel data," in *Proceedings of the International Conference on Learning Representations*, 2018.

[29] J. Gao, D. He, X. Tan, et al., "Representation Degeneration Problem in Training Natural Language Generation Models," in *Proceedings of the International Conference on Learning Representations*, 2019.

[30] N. Reimers and I. Gurevych, "Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2020.

[31] L. Zeng, "Leveraging Large Language Models for Code-Mixed Data Augmentation in Sentiment Analysis," in *Proceedings of the Workshop on Social Influence in Conversations*, 2024.

[32] G. Aguilar, B. McCann, T. Niu, et al., "Char2Subword: Extending the Subword Embedding Space Using Robust Character Compositionality," in *Findings of the Association for Computational Linguistics: Conference on Empirical Methods in Natural Language Processing*, Nov. 2021.

[33] F. Barbieri, L. Espinosa Anke, and J. Camacho-Collados, "XLM-T: Multilingual Language Models in Twitter for Sentiment Analysis and Beyond," in *Proceedings of the Language Resources and Evaluation Conference*, 2022.

[34] J. M. Pérez, D. A. Furman, L. Alonso Alemany, et al., "RoBERTuito: A pre-trained language model for social media text in Spanish," in *Proceedings of the Language Resources and Evaluation Conference*, 2022.

[35] Z. Xie, H. Zhao, T. Yu, et al., "Discovering Low-rank Subspaces for Language-agnostic Multilingual Representations," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2022.

[36] G. Ilharco, M. T. Ribeiro, M. Wortsman, et al., "Editing models with task arithmetic," in *Proceedings of the International Conference on Learning Representations*, 2023.