Modeling Code-Switch Language

Aryan Gulati
University of Southern California
Department of Computer Science
aryangul@usc.edu

Jordan Cahoon
University of Southern California
Department of Computer Science
jcahoon@usc.edu

Nicolas Perez
University of Southern California
Department of Computer Science
nicolagp@usc.edu

Abstract

Code-Switch (CS) language presents a unique challenge to modern natural language processing systems. Although CS language is frequently used in cross-lingual social contexts, it remains an understudied area. In this work, we explore fine-tuning pre-trained large language models to perform well on CS language for a variety of natural language processing tasks, including language modeling and sentiment analysis. In this report, we focus on the Spanish-English CS, known colloquially as Spanglish. We evaluated the performance of pre-trained auto-regressive and encoder-only language models that were fine-tuned on monolingual, multi-lingual, and CS data on CS language. We find that fine-tuning models on CS data enable models to better handle CS language, as shown by the improvement in the language modeling and sentiment analysis tasks. We also analyze the embedded space of these fine-tuned models and find that fine-tuning models on CS text improve language-agnostic embeddings compared to pre-trained models.

1 Introduction

Code-switching is a common linguistic phenomenon where an individual fluently incorporates multiple languages into a sentence. People often use code-switch (CS) language to communicate in multi-lingual, global settings, including social media, politics, and science. Yet, CS language is still a relatively new topic in natural language processing despite its frequency in spoken and written contexts. As large-language models become increasingly prevalent in our lives, we must develop models that perform well on CS input. These models may enhance the applicability and generalizability in many downstream tasks, including automatic speech recognition (ASR) and machine translation. Our project aims to explore CS input in large language models. In particular, we will examine CS in sentiment analysis.

2 Related Work

Numerous examples in the literature attempt to model CS language. The most basic task is identifying when language switch points occur [16], which has also been done in low-resource language-pairs [6].

There are efforts to use transfer learning to fine-tune general language models to the multilingual task [5][7], however, other approaches specifically target the CS task [2] by using a special CS corpus. This is also seen for downstream tasks like machine translation [12][17] and ASR [10]. A common problem with this approach is the lack of data, but there is work on the generation of artificial CS data for the task of machine translation [17]

Another interesting approach is to learn alignment for CS language without the need for CS data [8]. This can be done thanks to advances in cross-lingual word embedding models [15] [9].

3 Hypothesis

We aim to train a language model to model Spanish and English (Spanglish) CS using the Linguistic Code-switching Evaluation (LinCE) corpus. We will also use this data for sentiment analysis. Specifically, we will fine-tune pre-trained language models on CS input and evaluate the model on perplexity. We will compare the original pre-trained models as well as our fine-tuned models on the tasks of language modeling and sentiment analysis. We expect all fine-tuned models to have lower perplexity than the baseline. Furthermore, we expect the biggest performance gain from fine-tuning the code-switch data.

4 Datasets

CS language can occur in many different settings, for example, in social media, politics, or science. We specifically chose datasets containing Twitter data to consistently analyze models trained in different languages. We also restricted our search to datasets that contained sentiment labels in three classes: negative, neutral, and positive. In addition to the sentiment analysis datasets, we selected a dataset with parallel sentences to analyze the cross-lingual embedding spaces for each trained model.

The following are the chosen datasets for each language used in training.

4.1 Code-Switched Spanish/English

The LinCE corpus [1] is a centralized benchmark that contains data scraped from Twitter on four code-switch language pairs for four downstream tasks. We use the Spanish-English sentiment analysis dataset, which contains 18.8k CS sentences. They are categorized into three classes: positive (56.2%), negative (16.3%), neutral (27.6%). Additionally, every word in each sentence is labeled based on which language it belongs (lang1, lang2, other, and ambiguous).

4.2 English

The Massive text embedding benchmark (MTEB) corpus [13] contains 56 datasets on up to 112 languages and 8 tasks. We picked the dataset containing tweets in English, for the task of sentiment classification. This dataset contains 31k total tweets, categorized into positive (31.2%), negative (28.3%) and neutral (40.5%).

4.3 Spanish

The Sentiment Analysis at SEPLN (TASS) 2018 workshop [11] contains a variety of tasks with tweet data in different varieties of Spanish, from Spain, Peru, and Costa Rica. We aggregated the different varieties and ended up with a corpus consisting of 2k tweets. The tweets were also paired with sentiment labels in the three classes.

4.4 Parallel Sentences

To analyze the cross-lingual embedding spaces learned by our model, we took the XNLI: The Cross-Lingual NLI Corpus [3]. It provides parallel sentences in multiple languages, and we chose a subset of the 2490 English and Spanish examples for our study.

5 Experiments

5.1 Language Modeling

We used the GPT-2 [14] 1.5 billion parameter generative model to form a language modeling baseline. We selected GPT-2 for several reasons. It is an open-source large language model that is available at a manageable size, which enables us to iterate experiments at a reasonable pace, while still having the benefits of a large model. The model is pre-trained on 40GB of text, primarily comprised of English and code.

Parameter	Value
Learning rate	5.00×10^{-4}
Train steps	200
Sentence length	40
Batch size	32

Table 1: **Hyperparameters for GPT-2 Language Modeling.**

We assess the zero-shot performance of the pre-trained model on CS data as a baseline of how well the pre-trained model does on CS data in terms of perplexity. We then fine-tune GPT-2 on the monolingual datasets in English and Spanish, multilingual datasets sampling both English and Spanish, and the LinCE CS data. Since we only have about 2k training examples for Spanish, we limit the amount of data across all datasets to this number. This is done to ensure a fair comparison of the effects of fine-tuning caused by adding different data. In particular, the Spanish+English mix was achieved by selecting 1000 examples at random from each of the Spanish and English datasets.

5.2 Sentiment Analysis

5.2.1 Motivation

In addition to developing a general language model, we also wanted to evaluate approaches to using transformer models for downstream tasks that involve CS data. We chose the sentiment analysis task because the LinCe corpus [1] provides labeled sentiment analysis data.

To motivate the need for models capable of handling CS input, we examine the zero-shot performance of mBERT [5] and XLM-RoBERTa [4]. Specifically, we use the kNN-based approach described below and extract representations from the pre-trained checkpoints of both models. We test their performance on English, Spanish and CS inputs using data from the sources described in section 4. Table 2 shows that both models have comparable performance when presented with English or Spanish data, but suffer from drops in perfor-

	Code-Switch		English		Spanish	
	Accuracy	F1 Score	Accuracy	F1 Score	Accuracy	F1 Score
XLM-RoBERTa	39.12	36.75	47.43	47.42	47.25	44.20
mBERT	39.51	35.09	47.47	46.89	45.68	42.66

Table 2: **Zero-shot sentiment analysis performance of mBERT and XLM-RoBERTa on CS, Spanish, and English data.** Both models underperform when presented with CS input

mance when given CS input. This validates the necessity of developing approaches that improve these models' performance on CS data.

5.2.2 Fine-Tuning

To improve downstream task performance on code-switched inputs, we propose fine-tuning a large pre-trained model for the sentiment analysis task. We evaluate various fine-tuning approaches by varying the language of the data provided to the model. In all cases, we fine-tune the XLM-RoBERTa [4] model on a sentiment analysis dataset, and all models are trained on the same number of training instances. Additionally, the classes were balanced and downsized to 5.4k instances for each dataset. Specifically, we train the following models:

RoBERTweet fine-tuned on English-only data using the MTEB twitter sentiment analysis dataset [13].

RoBERTito fine-tuned on Spanish-only data from the TASS 2018 Twitter dataset [11].

RoBERTwito fine-tuned on a randomly chosen subset of the English and Spanish datasets, such that half of the training data is Spanish-only and the other half is English-only.

RoBERTinho fine-tuned on the code-switched Spanish/English sentiment analysis dataset from the LinCe corpus [1].

RoBERTinho-Plus fine-tuned on *all* of the code-switch data available. This model is the only one from the list that has a larger training data size of approximately 12,000 sentences.

5.2.3 Classification Approaches

After fine-tuning the models, we evaluate two different approaches for classification.

Linear Classification Layer All models are fine-tuned end-to-end with a softmax classification layer added after the final layer of the pre-trained model. This added classification layer is then used to generate predictions at test time.

KNN Classification In this method, we train the model end-to-end with a classification layer as above. However, to generate predictions after training, we mean-pool the representations from the penultimate layer (i.e. the input to the classifier) of each token in our sequence. These mean-pooled representations are then used as features for a k-Nearest-Neighbors classifier. This setting allows for zero-shot evaluation, as we do not need to fine-tune the model to generate meaningful representations.

5.3 Cross-Lingual Embedding Spaces

After fine-tuning the XLM-RoBERTa model [4] on each of our chosen datasets for sentiment analysis, we analyze the embedding spaces of each model. We hypothesize that fine-tuning on CS input may lead to more language-agnostic embeddings. To achieve this goal, we examine how close embeddings are for sentences with same semantic meaning in both Spanish and English, taken from the parallel sentences corpus XLNI [3].

In particular, we collect the embeddings generated by the model at its last hidden state for each token in the input text. These token embeddings are then mean-pooled to get a sentence-level embedding. The similarity of sentences is evaluated by computing cosine-similarity on the raw sentence-level embeddings.

The parallel sentences chosen for analysis are in Spanish and English. The baseline is established by the pre-trained embeddings for parallel sentences. Then, we compare the effect of fine-tuning on different kinds of data (monolingual, multilingual, and CS) on the similarity (cosine-similarity) between these sentences.

Additionally, we use principal component analysis (PCA) to analyze the reduced embedding space. We assess the cosine similarity and the normalized Euclidean distance between embeddings for parallel sentences for the first five principal components. We expect the cosine similarity to increase and normalized euclidean distance to decrease with fine-tuning on average.

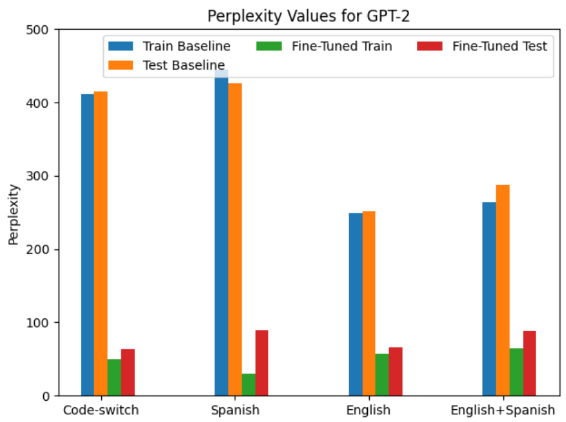


Figure 1: **Perplexity values for language modeling.** Fine-tuning decreases perplexity on training and test set for each dataset.

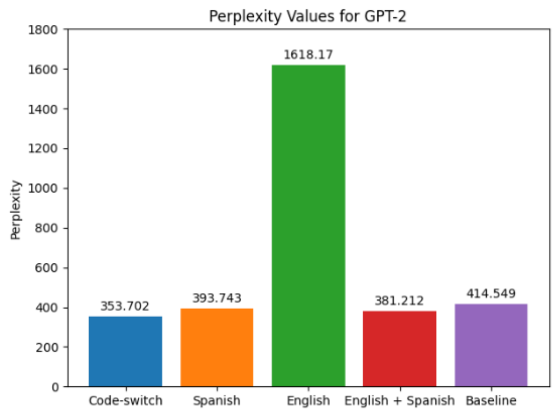


Figure 2: **Perplexity of fine-tuned models on CS Data.** Fine-tuning on code-switch achieves the lowest perplexity, although notably the English+Spanish model achieves similar performance.

6 Results

6.1 Language Modeling

Perplexity Baseline We calculated the perplexity using GPT-2 on each sentence in the dataset. As expected, perplexity is quite high, likely due to a combination of the lack of Spanish in the corpus used for pre-training and the small contexts given the short nature of tweets. The Train and Test Baseline depicts these splits in Figure 1.

Fine-tuning task per dataset First, we fine-tuned GPT-2 on each chosen dataset. In this case, we train and evaluate on the same dataset, for each of [English, Spanish, English+Spanish, Code-Switch]. We found that fine-tuning does reduce perplexity on each individual dataset. It's clear from Figure 1 that the Train and Test splits have both considerably lower perplexity than the baseline (before fine-tuning). An interesting point is that both the Code-Switch and the Spanish datasets experience the biggest reduction in perplexity after fine-tuning. This seems in line with our hypothesis since the pre-training data contained only a small number of Spanish examples.

Evaluation on CS Next, we took the fine-tuned models and evaluated them on the Code-Switch Test set. We can see from Figure 2 that fine-tuning on all datasets, excluding English, performed better on the CS Test set than the Baseline, which is the pre-trained GPT-2. Additionally, Code-Switch and English+Spanish fine-tuning seemed to do the best on the task. This result supports the idea that fine-tuning a mixed language corpus improves model performance on code-switch language modeling. The model fine-tuned on English did considerably worse due to overfitting – it seems that adding more English data is not beneficial for the base model GPT-2 when evaluating the Code-Switch language modeling task.

6.2 Sentiment Analysis

After examining all the fine-tuning methods described above in both evaluation settings (KNN vs. Linear), we found that any fine-tuning objective outperforms the XLM-RoBERTa baseline. However, monolingual fine-tuning on Spanish-only or English-only data leads to the best classification performance when using a linear classifier. This may be a result of the lack of CS data in the pre-training corpus of XLM-RoBERTa. Nonetheless, when extracting representations from our models and classifying with kNN, we find that models trained on CS-only data perform the best. This result indicates that training on CS data leads to the most meaningful learnt representations for CS text, and is further supported by our analysis of the embedded spaces of these models.

As described above, RoBERTinho-Plus is trained on all of the available CS sentiment analysis data. The LinCe dataset includes a very significant class imbalance, and when training on the entire dataset, the linear classifier overfits and only predicts the majority class. This leads to the high accuracy listed but a steep drop in F1 score. Nonetheless, using representations from RoBERTinho-Plus with a kNN classifier leads to competitive performance. This indicates that the representations it learns may still be meaningful, even if the final classification layer overfits.

6.3 Cross-Lingual Embeddings

We examined three metrics to quantify how the cross-lingual embedding spaces change after fine-tuning on CS input. Table 4 shows fine-tuning on CS data decreases the average cosine-similarity between parallel sentences for all datasets. This decrease suggests that training RoBERTa for the sentiment analysis task increases differences between embeddings and the raw embeddings may not be informative for examining the cross-lingual embedding space. Table 5 shows the similarity of parallel embeddings after PCA, as measured by cosine-similarity and Euclidian distance. Notably, fine-tuning on mono-lingual and CS input increases the cosine similarity and decreases the normalized distance compared to the pre-trained baseline, with RoBERTito and RoBERTinho achieving the best performance for each metric, respectively. Figure 3 demonstrates that fine-tuning on CS sentences disrupts the global structure of the XLM-RoBERTa embedding space, which is driven by language type. The clustering after fine-tuning is less distinct, and the parallel embeddings are closer together in the PCA space, which suggests that fine-tuning on CS input may improve language-agnostic embeddings.

7 Limitations

One important limitation of our work was the amount of data available. Since the Spanish dataset used had significantly smaller amounts of sentences for training, we limited the size of the other datasets to the same number. This was done so the comparison between the models fine-tuned on different datasets could be compared fairly in both the language modeling and sentiment analysis tasks. Furthermore, it's also important to notice that our best result in the language modeling task was achieved through training on the CS dataset, and the evaluation was also done on this dataset (though on unseen data). Even though the results are positive, we acknowledge that the test data was collected in the same methodology as the CS train set, so results might be skewed toward lower perplexity.

Another limitation we would like to point out is that the model RoBERTa is designed for word-level embeddings, but we use it to analyze sentence-level embeddings. Losses in information might occur when averaging out the word embeddings to get a sentence embedding.

8 Conclusion

Our experiments indicate that fine-tuning pre-trained models for CS language yields better results in a variety of tasks. Language modeling perplexity is improved the most by fine-tuning on this kind of data, but we also find that using a mixture of languages without code-switching may also be effective in developing CS Models. We also saw a similar trend on the downstream task of sentiment analysis – fine-tuning on tweet data generally improved model performance. Fine-tuning solely on CS data achieved comparable performance to models fine-tuned on multi-lingual and mono-lingual models. However, when examining the reduced cross-lingual embedding space, we find fine-tuning on CS may produce better language agnostic embeddings. Notably, XLM-RoBERTa was trained significantly more in mono-lingual instances than the few thousand CS instances used to fine-tune the model in the work shown here. Future work may explore if fine-tuning on larger sample sizes of CS data more drastically improves embedding quality for downstream tasks.

Because of these positive results, we believe it is worthwhile to explore the problem further by training larger models on a larger amount of data.

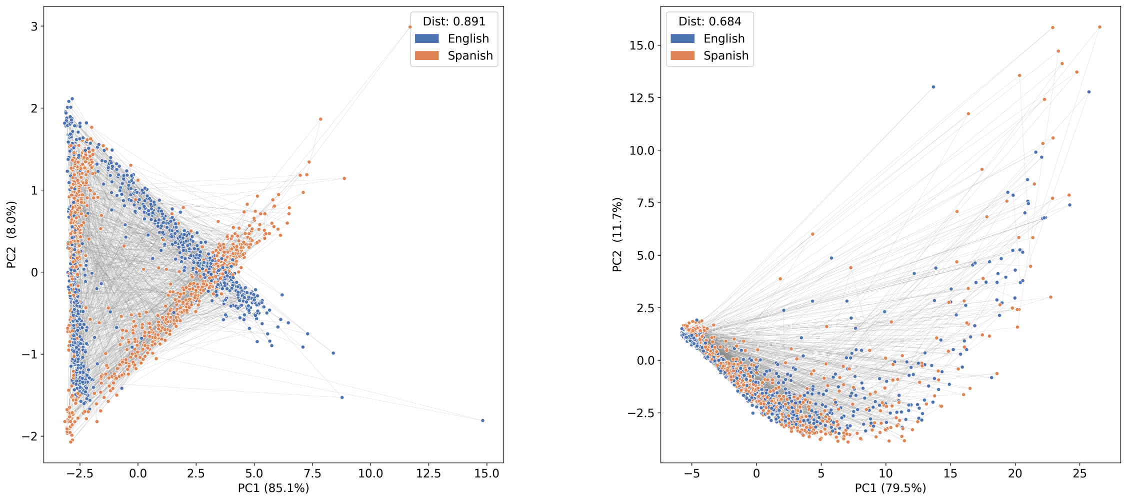
As CS language modeling improves, it could also have important consequences on other language tasks. An important example is Automatic Speech Recognition (ASR). Much of the interaction with personal assistant systems is done through voice, so enhancing these models could yield a much better experience for users who communicate in CS. Another application could be automatic captioning of videos that might include CS dialogue, or technical talks in other languages that employ terms in multiple foreign languages.

References

[1] Gustavo Aguilar, Sudipta Kar, and Thamar Solorio. "LinCE: A Centralized Benchmark for Linguistic Code-switching Evaluation". English. In: *Proceedings of the Twelfth Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, May 2020, pp. 1803–1813. ISBN: 979-10-95546-34-4.

	KNN		Linear	
	Accuracy	F1 Score	Accuracy	F1 Score
RoBERTinho	42.35	39.95	49.12	43.94
RoBERTweet	41.58	39.25	**50.20**	**49.23**
RoBERTito	42.66	39.69	**50.20**	44.02
RoBERTwito	44.50	**41.23**	49.27	47.58
RoBERTinho-Plus	**51.96**	39.46	56.26	24.00
XLM-RoBERTa	39.12	36.75	–	–

Table 3: **Performance of Fine-tuned CS Sentiment Models.** Fine-tuning improves sentiment analysis performance over the baseline. Fine-tuning with monolingual data and using a linear classifier leads to the best results, but the KNN performance implies that fine-tuning on CS leads to the best-learned representations



(a) XLM-RoBERTa

(b) RoBERTinho (fine-tuned)

Figure 3: **PCA of Cross-Lingual Embedding Space.** Fine-training on CS sentences decreases distance in PCA space for embeddings of parallel sentences and decreases clustering within language types.

Model	Cosine-Similarity
RoBERTinho (CS)	0.91
RoBERTweet (EN)	0.84
RoBERTito (ES)	0.81
RoBERTwito (EN+ES)	0.76
XLM-RoBERTa (baseline)	**0.98**

Table 4: **Average cosine-similarity for parallel embeddings.** Fine-tuning increases cosine similarity for embeddings.

URL: https://aclanthology.org/2020.lrec-1.223.

[2] Gustavo Aguilar and Thamar Solorio. "From English to Code-Switching: Transfer Learning with Strong Morphological Clues". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 8033–8044. DOI: 10.18653/v1/2020.acl-main.716. URL: https://aclanthology.org/2020.acl-main.716.

[3] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. "Translation artifacts in cross-lingual transfer learning". In: *arXiv preprint arXiv:2004.04721* (2020).

Model	Cosine Similarity	Normalized Distance
RoBERTinho	0.46	**0.68**
RoBERTweet	0.46	0.75
RoBERTito	**0.57**	0.71
RoBERTwito	0.26	0.89
XLM-RoBERTa	0.15	0.89

Table 5: **Average similarity for first five principal components of parallel sentence embeddings.** Fine-tuning increases the cosine similarity, with the Spanish Fine-tuned model performing the best. In contrast, fine-tuning on CS input decreases the Euclidian distances between parallel sentences in the PCA space the most.

[4] Alexis Conneau et al. "Unsupervised Cross-lingual Representation Learning at Scale". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by Dan Jurafsky et al. Online: Association for Computational Linguistics, July 2020, pp. 8440–8451. DOI: 10.18653/v1/2020.acl-main.747. URL: https://aclanthology.org/2020.acl-main.747.

[5] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[6] Jesin James et al. "Language Models for Code-switch Detection of te reo Māori and English in a Low-resource Setting". In: *Findings of the Association for Computational Linguistics: NAACL 2022*. Seattle, United States: Association for Computational Linguistics, July 2022, pp. 650–660. DOI: 10.18653/v1/2022.findings-naacl.49. URL: https://aclanthology.org/2022.findings-naacl.49.

[7] Guillaume Lample and Alexis Conneau. "Cross-lingual language model pretraining". In: *arXiv preprint arXiv:1901.07291* (2019).

[8] Grandee Lee and Haizhou Li. "Modeling code-switch languages using bilingual parallel corpus". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2020, pp. 860–870.

[9] Grandee Lee and Haizhou Li. "Word and class common space embedding for code-switch language modelling". In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2019, pp. 6086–6090.

[10] Guoyu Liu and Lixin Cao. "Code-switch speech rescoring with monolingual data". In: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2021, pp. 6229–6233.

[11] Franco M Luque and Juan Manuel Pérez. "Atalaya at TASS 2018: Sentiment Analysis with Tweet Embeddings and Data Augmentation." In: *Tass@ sepln*. 2018, pp. 29–35.

[12] Mohamed Amine Menacer et al. "Machine translation on a parallel code-switched corpus". In: *Advances in Artificial Intelligence: 32nd Canadian Conference on Artificial Intelligence, Canadian AI 2019, Kingston, ON, Canada, May 28–31, 2019, Proceedings 32*. Springer. 2019, pp. 426–432.

[13] Niklas Muennighoff et al. "MTEB: Massive text embedding benchmark". In: *arXiv preprint arXiv:2210.07316* (2022).

[14] Alec Radford et al. "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8 (2019), p. 9.

[15] Sebastian Ruder, Ivan Vulić, and Anders Søgaard. "A survey of cross-lingual word embedding models". In: *Journal of Artificial Intelligence Research* 65 (2019), pp. 569–631.

[16] Thamar Solorio and Yang Liu. "Learning to predict code-switching points". In: *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. 2008, pp. 973–981.

[17] Jitao Xu and François Yvon. "Can you traducir this? machine translation for code-switched input". In: *arXiv preprint arXiv:2105.04846* (2021).