Transfer Learning for Code-Mixed Data: Do Pretraining Languages Matter?

Kushal Tatariya[1] **Heather Lent**[2] **Miryam de Lhoneux**[1]
[1] Department of Computer Science, KU Leuven, Belgium
[2] Department of Computer Science, Aalborg University, Denmark
{kushaljayesh.tatariya, miryam.delhoneux}@kuleuven.be
hcle@cs.aau.dk

Abstract

Monolinguals make up a minority of the world's speakers, and yet most language technologies lag behind in handling linguistic behaviours produced by bilingual and multilingual speakers. A commonly observed phenomenon in such communities is code-mixing, which is prevalent on social media, and thus requires attention in NLP research. In this work, we look into the ability of pretrained language models to handle code-mixed data, with a focus on the impact of languages present in pretraining on the downstream performance of the model as measured on the task of sentiment analysis. Ultimately, we find that the pretraining language has little effect on performance when the model sees code-mixed data during downstream finetuning. We also evaluate the models on code-mixed data in a zero-shot setting, after task-specific finetuning on a monolingual dataset. We find that this brings out differences in model performance that can be attributed to the pretraining languages. We present a thorough analysis of these findings that also looks at model performance based on the composition of participating languages in the code-mixed datasets.

1 Introduction

In multilingual societies, contact between multiple languages has resulted in a plethora of linguistic phenomena that have long been the subject of study in linguistics, and more recently in NLP. One such phenomenon is code-switching, or code-mixing[1], in which speakers use material from two or more languages within the same conversation (Thomason, 2001).

Code-mixing typically occurs in informal registers and casual conversations, permitted or constrained by different sociolinguistic factors (Doğruöz et al., 2021). The typical lack of formality surrounding the use of code-mixing contributes to difficulties in data collection, as code-mixing is less likely to occur in official documents by governments and organizations, which have been reliable resources for the creation of many datasets (Sitaram et al., 2019). In contrast, social media has been a particularly fruitful domain for sourcing code-mixed data, useful in a wide variety of downstream tasks (Barman et al., 2014; Banerjee et al., 2016; Chakma and Das, 2016; Vijay et al., 2018; Patra et al., 2018a; Bohra et al., 2018). Among these tasks, sentiment analysis and offensive language detection stand out in particular, as Agarwal et al. (2017) have demonstrated that multilingual speakers are likely to utilize code-mixing to express their emotions, especially when cursing. Thus, improving methodologies for working with intricate code-mixed data is highly relevant to the study of sentiment analysis, and social media at large.

The advent of pretrained language models (PLMs) has tangibly shaped the norms for working with most languages, yet the implications for code-mixed data are much less clear. PLMs have so far largely operated under monolingual assumptions and biases (Ramesh et al., 2023; Talat et al., 2022). Most PLMs, including the massively multilingual ones, are trained on large web corpora, and studies have shown that the quality filters and data selection methodologies for these data sources tend to exclude text with dialectal nuances, such as text with non-standard varieties of English like African American English, or Hispanic-aligned English. (Dodge et al., 2021; Gururangan et al., 2022). Attempts have been made at language modelling for code-mixed data (Gupta, 2019; Nayak and Joshi, 2022), but an interesting question remains about how much the languages used in the pretraining of PLMs interact with each other to impact their performance on code-mixed data. A better understanding of this would enable targeted resource allocation to code-mixed NLP, and also potentially help understand how PLMs process language. PLMs

[1]Although distinctions between the two terms are made, we use them interchangeably.

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 365–378
July 14, 2023 ©2023 Association for Computational Linguistics

that have been pretrained on many high- and low-resource languages are now widely available and accessible, which provides a fertile ground for such analyses (Wolf et al., 2020). To shape the focus of this study, we introduce our hypothesis below.

Hypothesis: *PLMs trained exclusively on data from relevant languages would demonstrate better performance than those that contain other extraneous languages and/or are only trained on one language.*

At the same time, the "curse of multilinguality", coined by Conneau et al. (2019), refers to the trade-off between adding more languages to increase cross-lingual capabilities, and the consequences of adding too many which can ultimately lead to loss of performance across the board in monolingual and cross-lingual benchmarks. Massively multilingual models can be susceptible to this, and therefore we presume that models trained on data from relevant language families would be at an advantage. To this end, we test the performance of 7 pretrained language models on the task of sentiment analysis for different code-mixed datasets, which cover 6 languages.

2 Background

2.1 Code-Mixed NLP

In recent years, research in code-mixed NLP has steadily increased, resulting in the release of benchmark datasets like GLUE-CoS (Khanuja et al., 2020) and LinCE (Aguilar et al., 2020), organized shared tasks (Aguilar et al., 2018; Solorio et al., 2020, 2021), and several survey papers (Sitaram et al., 2019; Doğruöz et al., 2021; Winata et al., 2022). Although most code-mixing datasets include at least one high-resource language like English, progress in code-mixed NLP still lags behind as there exist additional challenges not present within the scope of monolingual work. Firstly, detecting or predicting when and where code-mixing will occur is non-trivial for a wide variety of linguistic reasons (Doğruöz et al., 2021). Most language identification approaches operate on the document or sentence level, rather than token level, and thus do not perform well for code-mixed data (Caswell et al., 2020). Moreover, some code-mixed data includes the use of multiple scripts, which can further complicate matters. Therefore, it is not surprising that, as Khanuja et al. (2020) found with mBERT, performance over code-mixed data is typically worse than monolingual counterparts, calling for further studies on the capabilities of PLMs on code-mixed data.

Studies in code-mixed sentiment analysis have demonstrated the strong relationship between a speaker's language choice and the sentiment they wish to convey. For example, Rudra et al. (2016) found that bilingual Hindi-English speakers preferred to express negative sentiments in Hindi. Similarly, Ndubuisi-Obi et al. (2019) found that Naija was used for expressing any kind of sentiment (i.e. high-emotion settings), in lieu of English for matter-of-fact statements. While this makes code-mixing relevant to studies in sentiment analysis, Zaharia et al. (2020) have noted that current methods in this space cannot cope when two languages come together to express one sentiment. Thus, improved methods for code-mixed NLP are also important for sentiment analysis in general, in a world where most people are bilingual or multilingual.

2.2 Transfer Learning

Transfer learning is the capacity of a model to take knowledge acquired from one language or domain and effectively apply it towards another. Thus, without enough data to create PLMs tailored to code-mixed language, transfer learning will undoubtedly play an important role in processing code-mixed text. PLMs have shown promising transfer learning abilities across languages that are similar (Pires et al., 2019; Lin et al., 2019; de Vries et al., 2022). Pires et al. (2019) demonstrated that successful cross-lingual transfer can lead to multilingual representations that are able to incorporate information from multiple languages, and even generalise across observed scripts, ultimately leading to increased performance on code-mixed data. PLMs have also been proven to have zero-shot transfer capabilities (Wu and Dredze, 2020), which can then be further enhanced by fine-tuning on limited instances from the target languages (Lauscher et al., 2020; de Vries et al., 2021). However, other work has shown that transfer learning is not always trivial. In the context of Creole NLP, Lent et al. (2022) found that even pretraining on languages with direct genealogical ties to the target Creoles failed to result in useful PLMs for those languages. Thus, further investigation of the mechanisms of pretraining data on the performance of PLMs is required.

3 Languages and Datasets

The datasets used in this study are mainly comprised of text scraped from Twitter, Facebook and YouTube. Details are summarised in Table 1. All datasets from this work can be found in our github repository[2].

Dataset	Language	Train / Dev
AfriSenti	pcm	5.1K / 1.2K
NaijaVader	pcm	9.8K / 1.4K
SAIL	hin-eng	10K / 1.2K
IIITH-CodeMix	hin-eng	2.7K / 388
TamilMixSentiment	tam-eng	110K / 1.2
MalayalamMixSentiment	mal-eng	4.2K / 480
DravidianCodeMix	tam-eng	33K / 4.2K
DravidianCodeMix	mal-eng	14K / 1.8K
DravidianCodeMix	kan-eng	5.2K / 656

Table 1: Details about the datasets in the study. The first four datasets have 3 labels - 'positive', 'negative' and 'neutral', and the latter five datasets have 4 labels - 'positive', 'negative', 'mixed_feelings' and 'unknown_state'.

3.1 Code-Switching in India

With the multitude of languages being spoken in India, and the plethora of bilingual and multilingual speakers, code-switching is a commonly observed phenomenon (Barnali, 2017). With the dominance of English in Indian society, educational institutions and official communications, there are millions of English speakers in India who can also be fluent in at least one other native Indian language. Thus, speakers can frequently switch between English and their other native language for ease of communication. Very commonly observed is Hindi-English code-switching, more popularly known as Hinglish, which refers to mixing of Hindi and English lexicon, phrases and syntax. In the written form, it is normally seen in Latin script. This paper looks at Hinglish, along with the mixing of English with Dravidian languages like Malayalam, Tamil and Kannada.

Hinglish Data For Hinglish we use the datasets curated by Joshi et al. (2016) (hereafter referred to as IIITH-CodeMix) and Patra et al. (2018b) (hereafter referred to as SAIL). The IIITH-CodeMix dataset consists of user comments from popular Indian Facebook pages, with comments not written in the Roman script, or comments completely in English being removed. The SAIL dataset, included in the GLUECoS benchmark, on the other hand, is Twitter data, again with only romanized instances of Hindi.

Dravidian Data For south Indian languages in the Dravidian language family, we use 5 datasets in 3 languages - Tamil, Malayalam and Kannada. The dataset introduced in Chakravarthi et al. (2020b) is referred to as TamilMixSentiment, with Tamil-English data, and (Chakravarthi et al., 2020a) is called MalayalamMixSentiment, containing Malayalam-English data. The remaining 3, in Tamil, Malayalam and Kannada, come from Chakravarthi et al. (2021), following a similar annotation scheme as the previous ones, hereafter referred to as DravidianCodeMix. All five datasets have been created from scraping YouTube comments.

The Dravidian datasets, unlike the others, contain text that is not in the Latin script. For this study, however, we transliterated all the non-Latin characters into Latin script to make fair comparisons between monolingual models that have not been trained on non-Latin script and the multilingual ones that have. Moreover, Moosa et al. (2023) found that transliteration helps improve multilingual model performance and cross-lingual representations. We used the transliteration library for Indic languages created by Madhani et al. (2022), trained on the Aksharantar dataset. Additionally, the original datasets contain 5 labels - 'positive', 'negative', 'mixed_feelings', 'unknown_state' and 'not_target_language'. All examples labeled 'not_target_language' were removed from the datasets since they contained non-Indic scripts that the transliteration model has not seen, and language identification falls outside the scope of this study.

3.2 Code-Switching in Nigeria

Nigerian Pidgin, commonly referred to as Naija, is the unofficial lingua franca in Nigeria (Ekundayo, 2022). It is an English-based Creole, which arose from language contact between English and local Nigerian languages such as Hausa, Yoruba, Igbo, and others. Despite the significant influence of English on the language, Naija is a fully independent language of its own, with aspects of morphology, syntax, and semantics that are detached from English (Agbo and Plag, 2020; Odiegwu, 2022). Code-mixing in Nigeria can often occur between English, Naija, and a given speaker's mother

[2] https://github.com/kushaltatariya/Sentiment-Analysis-for-Code-Mixed-Data

		Is It Present?								
		Monolingual		Multilingual		Indic		African		Code-mixed
Language	**Script**	BERT	RoBERTa	mBERT	XLM-R	IndicBERT	MuRIL	AfriBERTa	AfroXLMR	HingMBERT
English (eng)	Latin	✓	✓	✓	✓	✓	✓		✓	✓
Naija (pcm)	Latin							✓	✓	
Hinglish	Latin									✓
Hindi (hin)	Latin				✓		✓			
	Devanagari			✓	✓	✓	✓			✓
Malayalam (mal)	Latin						✓			
	Malayalam			✓	✓	✓	✓			✓
Tamil (tam)	Latin				✓		✓			
	Tamil			✓	✓	✓	✓			✓
Kannada (kan)	Latin						✓			
	Kannada			✓	✓	✓	✓			✓

Table 2: Languages present in the pretraining of each PLM.

tongue (Mensah and Ndimele, 2014; Akande and Salami, 2021; Sarah and Oladayo, 2021). However, the boundaries between Naija and code-mixing may not always be straightforward to diagnose, as Naija is amenable to immense variation from one speaker to another (Lent et al., 2022). While most datasets in Naija are not designed for studying code-mixing (with the exception of Ndubuisi-Obi et al. (2019)), we surmise that some code-mixing may be present in Naija text, as a result of Naija's flexibility for speakers' individual linguistic backgrounds. Therefore, we include Naija in our analysis to gain a perspective on how language models perform on code-mixing within a language in its own right. This choice is also in line with previous work, which acknowledges the propensity for code-mixing in Naija and other African Creoles (Adebara et al., 2022).

Naija Data We use two datasets for Naija. The first one was introduced by Oyewusi et al. (2020) (hereafter referred to as NaijaVader) within the VADER Sentiment Analysis framework (Hutto and Gilbert, 2014), containing tweets. The authors did not release official splits of the data, so we created our own train-dev-test splits. The second dataset (hereafter referred to as AfriSenti), is part of Muhammad et al. (2023), a Twitter sentiment analysis benchmark for African languages. They used a location and vocabulary based setup to collect tweets in each respective language.

4 Models

The PLMs compared in this study can be classified into four categories based on their pretraining data: monolingual, multilingual, Indic and African, presented in Table 2. We used the base version of each model for our experiments, without performing any additional pretraining.

Monolingual Models For this study, we focus mainly on standard English monolingual PLMs, namely BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019). The datasets contain code-mixing of various languages with English. Thus, English construes a large part of, and is a common thread in, language data that we analyse. Both these models also have multilingual versions, mentioned below, which serves us well for comparability.

Massively Multilingual Models The multilingual BERT model (mBERT) (Devlin et al., 2018) is a transformer model pretrained on the Wikipedias of 104 languages including some Indic and African languages. XLM-RoBERTa (XLM-R) (Conneau et al., 2020) is the multilingual version of RoBERTa, pretrained on 100 languages from the CommonCrawl corpus. The Hindi included in the pretraining is romanized Hindi, instead of Devanagari Hindi, which is notable for our purposes since we only have romanized Hindi in our Hinglish code-mixed datasets. XLM-R specialises in cross-lingual representations. Both PLMs were chosen based on their competitive performance on low-resource languages.

Indic Language Models Introduced by Doddapaneni et al. (2022), IndicBERT v2 is a PLM incorporating 24 Indian languages, including English. It is a standard BERT model pretrained on IndicCorp v2, introduced in the same paper, with the Masked Language Modelling (MLM) objective function. While there are different flavours of the model available that are trained on an additional Translation Language Modelling (TLM) objective, we use the standard MLM-only model since we found marginal differences in the scores when we tested

Dataset	IndicBERT	MuRIL	AfriBERTa	Afroxlmr	mBERT	XLM-R	BERT	RoBERTa	HingMBERT
AfriSenti	-	-	0.75	**0.78**	0.76	0.77	0.77	0.76	0.77
NaijaVader	-	-	0.72	0.74	0.73	**0.74**	0.74	0.73	0.73
SAIL	0.62	0.62	-	-	0.60	0.64	0.60	0.61	**0.66**
IIITH-CodeMix	0.69	0.73	-	-	0.69	0.71	0.70	0.70	**0.74**
TamilMixSentiment	**0.71**	0.70	-	-	0.70	0.71	0.70	0.71	0.71
DravidianCodeMix (tam)	0.64	0.64	-	-	0.65	**0.66**	0.65	0.65	0.66
MalayalamMixSentiment	0.73	0.73	-	-	0.73	**0.74**	0.71	0.74	0.73
DravidianCodeMix (mal)	0.76	**0.77**	-	-	0.75	0.76	0.76	0.75	0.77
DravidianCodeMix (kan)	**0.71**	0.70	-	-	0.70	0.67	0.66	0.70	0.70

Table 3: Accuracy scores on the validation sets. Bold indicates best result for a dataset. The first two datasets are in Naija, next two in Hinglish, then Tamil-English, Malayalam-English, and the final single dataset is for Kannada-English code-mixing.

both the models on our datasets.

MuRIL (Khanuja et al., 2021) contains 16 Indian languages and English, from the Common Crawl OSCAR corpus, Wikipedia, PMINDIA corpus and the Dakshina Dataset, trained on the MLM and TLM objective functions. The TLM objective leverages both translated and transliterated data, to account for code-mixing.

African Language Models For the Naija datasets, we compare two language models trained on African languages, and the only models in our roster that include Naija in the pretraining.

AfriBERTa (Ogueji et al., 2021) is a transformer-based language model pretrained on 11 low-resourced African languages, with data sourced from the BBC news and the Common Crawl Corpus. It is trained with the standard MLM objective.

AfroXLMR (Alabi et al., 2022) is currently the largest available PLM for African languages. This model results from applying multilingual adaptive finetuning on XLM-R, with language adaption being performed on 17 African languages, and 3 other high resource languages spoken on the continent, including English sourced from the mt5 pretraining corpus, the BBC and other news websites.

Code-mixed Language Model We also include HingMBERT (Nayak and Joshi, 2022), a PLM containing Hinglish data in the pretraining. It is a multilingual BERT model that has been further pretrained on the L3Cube-HingCorpus. In the same work, the HingCorpus consists of code-mixed tweets - both in Latin script and transliterated into Devanagari. While there is a version of the model that has been pretrained on both Latin and Devanagari script, we use HingMBERT pretrained only on the latinized corpus to match our data.

In summary, each of the above PLMs selected for this work included training data for at least one language relevant to the target code-mixed data. Thus, we refine our hypothesis:

Refined Hypothesis: *Indic language models would perform better on the Indic datasets, and the African language models would perform better on the Naija datasets, than the monolingual or multilingual language models. Additionally, the code-mixed language model would perform better on the Hinglish datasets than the other PLMs.*

5 Experiments

We used the Massive Choice Ample Tasks (MaChAmp) (van der Goot et al., 2021) codebase for the experiments. MaChAmp provides an efficient and effective way to finetune PLMs on downstream tasks.

5.1 Finetuning

We finetuned the models on the training data from the code-mixed datasets. For the Indic datasets we finetuned the monolingual, multingual, codemix and Indic language models, while for the Naija datasets we finetuned the monolingual, multilingual, code-mixed and African models. We ran the experiments for 50 epochs, maintaining the same hyperparameters across all the models and datasets, and chose the model with the best performance on the validation set.

Finetuning Results We report the validation scores of each model-dataset combination in Table 3. Contrary to the hypothesis, there is not a very tangible difference observed between the performance of each model on the datasets. Models trained on relevant languages in some cases do have the best performance, like Afroxlmr with AfriSenti, which as seen in Table 2 contains Naija in the pretraining. Similarly with HingMBERT and the Hinglish datasets, and MuRil and IndicBERT with DravidianCodeMix (kan) and TamilMixSentiment, but this difference is very marginal. MuRil, trained on Indic languages, outperforms monolingual BERT on DravidianCodeMix (mal) by just one accuracy point. So does Afroxlmr with AfriSenti, where BERT is just one point behind.

On the other hand, for the datasets NaijaVader, MalayalamMixSentiment and DravidianCodeMix (tam), where the PLMs trained on relevant language families do not outperform the other models, XLM-R comes on top, but again with minimal difference. For NaijaVader, three categories of PLMs have very similar accuracy scores - BERT from the monolingual category, Afroxlmr from the African category and XLM-R from the multilingual category.

5.2 Other Tasks

Results from the above section raise the question whether models perform fairly similarly because the models are able to learn simple spurious correlations to classify sentiment, rather than relying on the PLM's capacity to understand the code-mixed data. To rule out this possibility, we performed similar experiments with Named Entity Recognition (NER), sarcasm detection and universal dependency parsing (UDPoS) datasets. If PLM performance on these tasks yield similar results to the sentiment analysis tasks, we can conclude that our findings thus far are pertinent to the capabilities of PLMs on code-mixed data, generally.

NER For NER, we use the dataset introduced by Singh et al. (2018), which is also part of the GlueCoS benchmark. It is a Hinglish dataset of code-mixed tweets annotated with BIO labels for persons, organisations and locations. The authors did not release official train-dev-test splits for the data, so we created our own, resulting in 50k tokens in the training set, and 7k in the validation. We then finetuned the monolingual, multilingual, code-mixed and Indic models on the training data. We also ran a similar experiment with the monolingual, multilingual and African models on the Naija part of MasakhaNER (Adelani et al., 2021), which showed similar results as discussed for Hinglish below. However, since MasakhaNER is sourced from BBC Pidgin, and owing to the formality of the register is less likely to contain code-switching, we report the results for it in Appendix A.1.

Sarcasm Detection For sarcasm detection, we use the dataset curated by Shah and Maurya (2021), consisting of 144k tweets in Hinglish. They are annotated based on the presence of hashtags, where all tweets with #sarcasm, #sarcastic, #irony, #humor were labelled as positive, and others with general hashtags like #politics, #food, #movie were labelled as negative for sarcasm. We used the splits released by the authors, and finetuned the monolingual, multilingual, code-mixed and Indic models on the training data consisting of 115K examples.

	NER	Sarcasm	UDPoS
IndicBERT	0.77	0.89	-
MuRIL	0.77	0.90	-
AfriBERTa	-	-	0.99
AfroXLMR	-	-	**0.99**
mBERT	0.78	0.89	0.99
XLM-R	0.77	0.90	0.99
BERT	0.76	0.89	0.99
RoBERTa	0.76	0.89	0.99
HingMBERT	**0.78**	**0.90**	-

Table 4: NER span-f1 and accuracy scores for sarcasm detection and UDPoS on validation sets.

UDPoS For UDPoS we use the Naija dataset introduced by Caron et al. (2019), consisting of 140k words. While it is not a social media dataset, it contains transcriptions of spoken Naija from different domains like speeches, free conversations, comments about current affairs, radio programs etc. Spoken data such as the kind included in this dataset contains a similar informality to social media, and thus likely to also contain code-switching. We used the official splits released by the authors and finetuned the monolingual, multilingual and African models on the training data.

Other Results The scores for sequence labelling with NER and UDPoS, and classification with sarcasm detection, presented in Table 4, show similar trends to that of sentiment analysis. All the models perform equally well, with the difference between the best and the worst being 2 percentage points in NER, 1 percentage point in sarcasm detection and less than 1 percentage point in UDPoS.

5.3 Zero-shot

Since there were only slight differences observed between the models when finetuning on code-mixed data, we evaluated the models on the code-mixed data in a zero-shot setting. In this scenario, there was no code-mixed data present in the downstream finetuning of the models, before testing on code-mixed data. We performed the zero-shot experiments with the Hinglish datasets and thus, we used monolingual Hindi and English sentiment analysis datasets for downstream finetuning of the monolingual, multilingual, code-mixed and Indic models. This could potentially bring out differences in model performance, if any, that arise from differences in pretraining data.

For the Hindi data, we used the sentiment analysis dataset created by Akhtar et al. (2016), which is also included in the IndicGLUE benchmark (Kakwani et al., 2020). It contains two indivdual datasets from two different domains - movie reviews and product reviews. While the movie reviews contain entire reviews that can potentially span one or two paragraphs as individual data points, the product reviews contain one or two sentences. Thus, to match the structure of the code-mixed datasets, we only use the product review dataset for downstream finetuning in Hindi. This dataset is in the Devanagari script, so we first transliterated it into Latin script for comparability.

For the English data, we used a reduced version of the SST-2 dataset (Socher et al., 2013), from the GLUE benchmark (Wang et al., 2018), reduced to match the size of the Hindi dataset to eliminate size as a potential factor in the results. We then evaluated these models on the validation sets from SAIL and IIITH-CodeMix. Moreover, the English and the Hindi datasets only have two sentiment labels - 'negative' and 'positive'. Thus, we removed the instances labelled 'neutral' from the Hinglish validation sets for this scenario.

Zero-shot Results Scores from the zero-shot experiments are in Table 5. Pretraining data here seems to make a drastic difference in the relative performance of the models. For both datasets, HingMBERT outperforms other models by a substantial margin, in both English and Hindi settings. When comparing models that do not contain code-mixed data in the pretraining, in the English setting, RoBERTa performs the best on both the datasets. On the other hand, MuRIL shows a very drastic decline in accuracy, being the worst on both datasets. This is reversed in the Hindi setting, where MuRIL outperforms the others, and RoBERTa is the least accurate by a large margin.

	SAIL		IIITH-CodeMix	
	Hindi	**English**	**Hindi**	**English**
IndicBERT	0.62	0.61	0.60	0.56
MuRIL	0.64	0.57	0.74	0.43
mBERT	0.57	0.56	0.64	0.47
XLM-R	0.63	0.62	0.70	0.46
BERT	0.61	0.62	0.63	0.57
RoBERTa	0.61	0.66	0.55	0.73
HingMBERT	**0.72**	**0.69**	**0.78**	**0.77**

Table 5: Zero-shot scores on Hinglish validation sets with Hindi and English task-specific finetuning.

	IIITH-Codemix	NaijaVader
IndicBERT	0.69	0.74
MuRIL	0.73	0.72
AfriBERTa	0.68	0.72
Afroxlmr	0.70	0.74
Best Model	0.74	0.74

Table 6: Accuracy scores of Indic models on a Naija dataset and African models on a Hinglish dataset, along with the best scores for each dataset from Table 3.

6 Analysis

It can be inferred from the above results that for code-mixed datasets, when finetuning a PLM on the code-mixed language, the languages seen in the pretraining may not substantially impact the performance of the model. We further confirmed this by finetuning the African models on IIITH-CodeMix, and the Indic models on NaijaVader. The results are in Table 6.

IndicBERT on NaijaVader is on par with the best performing model, and the African models do not demonstrate a drastic decline in performance on IIITH-CodeMix as compared to the Indic models. On the other hand, the pretraining languages of a PLM greatly influence performance scores when testing on code-mixed data in a zero-shot setting.

6.1 Language Identification and Composition

To understand these scores further, we looked at the composition of each participating language in the datasets, and compared the predictions of each model to see, whether despite overall accuracy being similar in the finetuning scenario, the models were performing better on one language than the other.

To this end, we ran a language identification (LID) model for code-mixed data on the Hinglish validation sets, using the CodeSwitch (Sarkar, 2020) tool, trained on data from the LinCE benchmark. The LID model takes in a code-mixed sentence, tokenizes it into subwords and outputs a language score for each subword. There were instances where the model assigned different languages for subwords from the same word. In these cases we picked the language assigned to the first subword. We manually verified the accuracy of LID on a sample from the IIITH-CodeMix dataset, and with a 95% accuracy, found it suitable enough for our purposes.

We assigned a majority language to each instance in the dataset, where if the instance had more than 50% words in English, it was categorised as *mostly-English*, and *mostly-Hindi* otherwise. Thus, we looked at the predictions of each model for the *mostly-English* and *mostly-Hindi* sentences to see whether, for example, the Indic or code-mixed PLMs were outperforming on the *mostly-Hindi* sentences, and failing on the *mostly-English*.

6.2 Implications of Language Composition: The Finetuning Scenario

Figure 1 illustrates the results. For IIITH-CodeMix, all models perform similarly on the *mostly-Hindi* examples, with MuRIL and HingMBERT performing slightly better. There are slightly larger differences in performance with the *mostly-English* examples, with the monolingual and code-mixed PLMs performing better than the multilingual and Indic PLMs. For the SAIL dataset, there is also a difference seen in performance on the *mostly-Hindi* examples, where the code-mixed PLM is able to handle them the best, followed closely by multilingual XLM-R. Not surprisingly, the monolingual models trail behind, with almost a 10 percentage point difference between HingMBERT and BERT. The *mostly-English* examples have similar performances across the models, with monolingual RoBERTa slightly ahead. All models perform better on *mostly-English* than on *mostly-Hindi* examples, with the pretraining language of the PLM potentially accounting for how big that difference is. The difference is larger in monolingual models compared to the others.

Another notable observation is that for SAIL, HingMBERT performs almost equally on *mostly-English* and *mostly-Hindi* examples. This could be attributed to the language composition of each dataset, where about 40% of the SAIL dataset is *mostly-English*, while the IIITH-CodeMix dataset only has about 14% *mostly-English*. Thus the distribution of the parent languages is more even in SAIL and heavily skewed towards Hindi in IIITH-CodeMix. Therefore, it can be argued that the code-mixed language model also learns the distribution of the participating languages in the dataset during training, and that reflects on the predictions of the model.

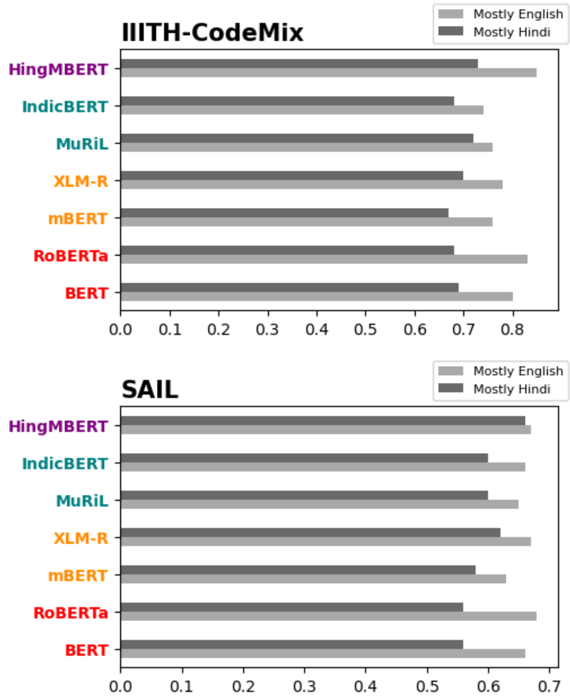

Figure 1: PLM performance relative to LID. The IIITH-CodeMix dev set was 14% *mostly-English* utterances, while the SAIL dev set was 40% *mostly-English* utterances.

We also looked at the distribution of sentiment labels for the *mostly-English* and *mostly-Hindi* examples, and compared model predictions to see if the models showed any bias toward a particular label for a language, but we saw no difference.

Since there are no such LID tools available for the other languages in our roster, we tested the CodeSwitch LID tool on samples from the other datasets as well. We found that the model is able to identify the English words in the samples satisfactorily, if not the other participating languages. So we ran the LID model on all the validation sets from the rest of the Indic and Naija datasets, and conducted similar analyses. The results confirmed the findings from the Hinglish datasets, but since the tool is not very reliable for these languages, we only report the results in Appendix A.2.

6.3 Implications of Language Composition: The Zero-Shot Scenario

The scenario described in the previous section takes a turn when evaluating the models in a zero-shot setting. From the results in Table 5, we find that pretraining has a major impact on the model performance, along with the composition of the parent languages in the dataset. As mentioned before, SAIL has a much more even composition of *mostly-Hindi* and *mostly-English* examples than IIITH-CodeMix.

This reflects in the performance of the models with respect to the finetuning language. While the code-mixed PLM does not show much difference in both scenarios on both datasets, the multilingual models suffer more with English finetuning than Hindi on IIITH-CodeMix, but do not show much difference in SAIL. Interestingly, BERT seems to suffer with English finetuning on IIITH-CodeMix, while RoBERTa has a jump in performance, even though they are both monolingual models pretrained on English data, and IIITH-CodeMix has more Hindi than English text. RoBERTa, in fact, suffers from Hindi finetuning on both the datasets. Conversely, MuRIL always suffers from English finetuning, more on IIITH-CodeMix than SAIL, which can be attributed to parent language composition of the datasets.

When comparing IndicBERT and MuRIL, differences in pretraining also reflect on the scores. MuRIL has seen romanized Hindi, with the TLM objective leveraging transliterated data as well, while the IndicBERT model we used has not. Thus, when finetuning with romanized Hindi, MuRIL has a significant bump in performance, in both cases performing better than IndicBERT. This could also be seen as a drawback for MuRIL when finetuning with English since it performs worse than IndicBERT on both SAIL and IIITH-CodeMix.

7 Summary

We summarise the findings of the paper in this section to answer the main underlying question of this work - do pretraining languages matter? We approach this question for code-mixed data in two transfer learning settings: with in-language finetuning, and zero-shot.

- When finetuning a PLM on a code-mixed dataset, the effects of the pretraining languages of the PLMs do not reflect in the performance scores substantially.

- In the finetuning setting when looking at PLM performance relative to language ID, all the PLMs perform better on the *mostly-English* sentences, than on *mostly-Hindi*, with the pretraining languages of the PLM and the language composition of the dataset potentially accounting for how big that difference is.

- In a zero-shot setting, the pretraining languages of the PLM do matter for performance.

- The language used to finetune a PLM greatly affects performance in the zero-shot setting. MuRIL is the best performing model with Hindi finetuning and RoBERTa has the highest score with English finetuning. The language composition of the dataset also potentially affects how much the score of the best performing model differs from the least performing model.

8 Conclusion

In this study, we found that the pretraining languages do not matter much for performance when downstream finetuning a PLM on code-mixed data. The finetuning process, to an extent, negates the effects of the pretraining languages in the PLMs and generates even performance across the board. On the other hand, the pretraining language of the models and the language composition of the data, both seem to be factors in model performance in a zero-shot setting. Overall, it can be better to use a PLM with pretraining on code-mixed languages like Hinglish, but this may not be possible for all types of code-mixed languages. Moreover, it does not seem to prove advantageous when it comes to Naija. Thus, this study can be used as a starting point for further interpretability analysis of PLMs, to understand exactly why in some settings the pretraining languages matter, and in some settings they don't.

9 Limitations

A large limitation of this work is the ubiquity of English. With the exception of the AfriBERTa (which has seen Naija), the remaining PLMs in this study all included English in the pretraining data. As a result, it is difficult to disentangle the benefits of including relevant languages in the pretraining data, from the general benefits of including *English* in the pretraining data, for processing code-mixed text. To this effect, future work in examining the capacity of PLMs for code-mixed language would benefit from examining commonly code-mixed language pairs, that do not involve English (e.g. Turkish-German).

In a similar vein, our work is limited in that we did not try other non-English monolingual PLMs. For the Indic languages, this is because monolingual Indic PLMs typically use the Devanagari script, but the datasets in this paper are constrained to using the Latin script. For Naija, we likewise did not experiment with monolingual models for the other relevant Nigerian languages; to our knowledge, most publicly available PLMs for Hausa, Yoruba, and Igbo seem to be created through continued pretraining with monolingual data over existing multilingual PLMs. Thus, experimenting with these models still does not strictly control for English and other languages.

Beyond PLMs, another limitation of this work pertains to the error analysis, which hinges upon currently available LID technologies. As explored in detail by Caswell et al. (2020), most LID technologies operate on a document level, and thus intra-utterance LID is still an open problem. For code-mixed language, the lack of robust LID puts limits us to coarser-grained analysis of the data (e.g. partitioning samples by *mostly-English* or *mostly-Hindi*). Ideally, a finer-grained partition of the data could be useful in determining the extent to which a PLM's knowledge of English enables performance on downstream tasks.

10 Acknowledgements

The computational resources and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation - Flanders (FWO) and the Flemish Government - department EWI (for Kushal Tatariya and Miryam de Lhoneux). This work is also funded by the Carlsberg Foundation under an Accelerate career grant entitled "Multilingual Modelling for Resource-Poor Languages", grant code CF21-0454 (for Heather Lent).

References

Ife Adebara, AbdelRahim Elmadany, Muhammad Abdul-Mageed, and Alcides Alcoba Inciarte. 2022. Afrolid: A neural language identification tool for african languages. In *Conference on Empirical Methods in Natural Language Processing*.

David Ifeoluwa Adelani, Jade Abbott, Graham Neubig, Daniel D'souza, Julia Kreutzer, Constantine Lignos, Chester Palen-Michel, Happy Buzaaba, Shruti Rijhwani, Sebastian Ruder, Stephen Mayhew, Israel Abebe Azime, Shamsuddeen H. Muhammad, Chris Chinenye Emezue, Joyce Nakatumba-Nabende, Perez Ogayo, Aremu Anuoluwapo, Catherine Gitau, Derguene Mbaye, Jesujoba Alabi, Seid Muhie Yimam, Tajuddeen Rabiu Gwadabe, Ignatius Ezeani, Rubungo Andre Niyongabo, Jonathan Mukiibi, Verrah Otiende, Iroro Orife, Davis David, Samba Ngom, Tosin Adewumi, Paul Rayson, Mofetoluwa Adeyemi, Gerald Muriuki, Emmanuel Anebi, Chiamaka Chukwuneke, Nkiruka Odu, Eric Peter Wairagala, Samuel Oyerinde, Clemencia Siro, Tobius Saul Bateesa, Temilola Oloyede, Yvonne Wambui, Victor Akinode, Deborah Nabagereka, Maurice Katusiime, Ayodele Awokoya, Mouhamadane MBOUP, Dibora Gebreyohannes, Henok Tilaye, Kelechi Nwaike, Degaga Wolde, Abdoulaye Faye, Blessing Sibanda, Orevaoghene Ahia, Bonaventure F. P. Dossou, Kelechi Ogueji, Thierno Ibrahima DIOP, Abdoulaye Diallo, Adewale Akinfaderin, Tendai Marengereke, and Salomey Osei. 2021. MasakhaNER: Named entity recognition for African languages. *Transactions of the Association for Computational Linguistics*, 9:1116–1131.

Prabhat Agarwal, Ashish Sharma, Jeenu Grover, Mayank Sikka, Koustav Rudra, and Monojit Choudhury. 2017. I may talk in english but gaali toh hindi mein hi denge : A study of english-hindi code-switching and swearing pattern on social networks. In *2017 9th International Conference on Communication Systems and Networks (COMSNETS)*, pages 554–557.

Ogechi Florence Agbo and Ingo Plag. 2020. The relationship of nigerian english and nigerian pidgin in nigeria: Evidence from copula constructions in ice-nigeria. *Journal of Language Contact*.

Gustavo Aguilar, Fahad AlGhamdi, Victor Soto, Thamar Solorio, Mona Diab, and Julia Hirschberg, editors. 2018. *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*. Association for Computational Linguistics, Melbourne, Australia.

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.

Akinmade T. Akande and Oladipo Salami, editors. 2021. *Current Trends in Nigerian Pidgin English A Sociolinguistic Perspective*. De Gruyter Mouton, Berlin, Boston.

Md Shad Akhtar, Ayush Kumar, Asif Ekbal, and Pushpak Bhattacharyya. 2016. A hybrid deep learning architecture for sentiment analysis. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 482–493, Osaka, Japan. The COLING 2016 Organizing Committee.

Jesujoba O. Alabi, David Ifeoluwa Adelani, Marius Mosbach, and Dietrich Klakow. 2022. Adapting pre-trained language models to African languages via multilingual adaptive fine-tuning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4336–4349, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Somnath Banerjee, Sudip Kumar Naskar, Paolo Rosso, and Sivaji Bandyopadhyay. 2016. The first cross-script code-mixed question answering corpus. In *MultiLingMine@ECIR*.

Utsab Barman, Amitava Das, Joachim Wagner, and Jennifer Foster. 2014. Code mixing: A challenge for language identification in the language of social media. In *CodeSwitch@EMNLP*.

Chatterjee Barnali. 2017. Code-switching and mixing in communication a study on language contact in indian media. *Social Science Research Network*.

Aditya Bohra, Deepanshu Vijay, Vinay Singh, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. A dataset of hindi-english code-mixed social media text for hate speech detection. In *PEOPLES@NAACL-HTL*.

Bernard Caron, Marine Courtin, Kim Gerdes, and Sylvain Kahane. 2019. A surface-syntactic UD treebank for Naija. In *Proceedings of the 18th International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2019)*, pages 13–24, Paris, France. Association for Computational Linguistics.

Isaac Caswell, Theresa Breiner, Daan van Esch, and Ankur Bapna. 2020. Language ID in the wild: Unexpected challenges on the path to a thousand-language web text corpus. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6588–6608, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Kunal Chakma and Amitava Das. 2016. Cmir: A corpus for evaluation of code mixed information retrieval of hindi-english tweets. *Computación y Sistemas*, 20:425–434.

Bharathi Raja Chakravarthi, Navya Jose, Shardul Suryawanshi, Elizabeth Sherly, and John Philip McCrae. 2020a. 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.

Bharathi Raja Chakravarthi, Vigneshwaran Muralidaran, Ruba Priyadharshini, and John Philip McCrae. 2020b. Corpus creation for sentiment analysis in code-mixed Tamil-English text. 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 202–210, Marseille, France. European Language Resources association.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Vigneshwaran Muralidaran, Navya Jose, Shardul Suryawanshi, Elizabeth Sherly, and John P. McCrae. 2021. Dravidiancodemix: Sentiment analysis and offensive language identification dataset for dravidian languages in code-mixed text. *CoRR*, abs/2106.09460.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

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

Wietse de Vries, Martijn Bartelds, Malvina Nissim, and Martijn Wieling. 2021. Adapting monolingual models: Data can be scarce when language similarity is high. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4901–4907, Online. Association for Computational Linguistics.

Wietse de Vries, Martijn Wieling, and Malvina Nissim. 2022. Make the best of cross-lingual transfer: Evidence from POS tagging with over 100 languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7676–7685, Dublin, Ireland. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Sumanth Doddapaneni, Rahul Aralikatte, Gowtham Ramesh, Shreya Goyal, Mitesh M. Khapra, Anoop Kunchukuttan, and Pratyush Kumar. 2022. Indicxtreme: A multi-task benchmark for evaluating indic languages.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. 2021. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1286–1305, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Omowumi Olabode Steven Ekundayo. 2022. Naija: The cinderella for nigerian and west african national language, unity and identity. *Journal of General Education and Humanities*.

Vivek Kumar Gupta. 2019. "hinglish" language - modeling a messy code-mixed language. *CoRR*, abs/1912.13109.

Suchin Gururangan, Dallas Card, Sarah K. Dreier, Emily K. Gade, Leroy Z. Wang, Zeyu Wang, Luke Zettlemoyer, and Noah A. Smith. 2022. Whose language counts as high quality? measuring language ideologies in text data selection. *CoRR*, abs/2201.10474.

Clayton J. Hutto and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. *Proceedings of the International AAAI Conference on Web and Social Media*.

Aditya Joshi, Ameya Prabhu, Manish Shrivastava, and Vasudeva Varma. 2016. Towards sub-word level compositions for sentiment analysis of Hindi-English code mixed text. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2482–2491, Osaka, Japan. The COLING 2016 Organizing Committee.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961, Online. Association for Computational Linguistics.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, Shruti Gupta, Subhash Chandra Bose Gali, Vish Subramanian, and Partha Talukdar. 2021. Muril: Multilingual representations for indian languages.

Simran Khanuja, Sandipan Dandapat, Anirudh Srinivasan, Sunayana Sitaram, and Monojit Choudhury. 2020. GLUECoS: An evaluation benchmark for code-switched NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3575–3585, Online. Association for Computational Linguistics.

Anne Lauscher, Vinit Ravishankar, Ivan Vulic, and Goran Glavas. 2020. From zero to hero: On the limitations of zero-shot cross-lingual transfer with multilingual transformers. *CoRR*, abs/2005.00633.

Heather Lent, Kelechi Ogueji, Miryam de Lhoneux, Orevaoghene Ahia, and Anders Søgaard. 2022. What a creole wants, what a creole needs. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6439–6449, Marseille, France. European Language Resources Association.

Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, Antonios Anastasopoulos, Patrick Littell, and Graham Neubig. 2019. Choosing transfer languages for cross-lingual learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3125–3135, Florence, Italy. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Yash Madhani, Sushane Parthan, Priyanka A. Bedekar, Ruchi Khapra, Vivek Seshadri, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh M. Khapra. 2022. Aksharantar: Towards building open transliteration tools for the next billion users. *ArXiv*, abs/2205.03018.

Eyo O. Mensah and Roseline Ihuoma Ndimele. 2014. Linguistic creativity in nigerian pidgin advertising. *Sociolinguistic Studies*, 7:321–344.

Ibraheem Muhammad Moosa, Mahmud Elahi Akhter, and Ashfia Binte Habib. 2023. Does transliteration help multilingual language modeling?

Shamsuddeen Hassan Muhammad, Idris Abdulmumin, Abinew Ali Ayele, Nedjma Ousidhoum, David Ifeoluwa Adelani, Seid Muhie Yimam, Ibrahim Sa'id Ahmad, Meriem Beloucif, Saif Mohammad, Sebastian Ruder, Oumaima Hourrane, Pavel Brazdil, Felermino Dário Mário António Ali, Davis Davis, Salomey Osei, Bello Shehu Bello, Falalu Ibrahim, Tajuddeen Gwadabe, Samuel Rutunda, Tadesse Belay, Wendimu Baye Messelle, Hailu Beshada Balcha, Sisay Adugna Chala, Hagos Tesfahun Gebremichael, Bernard Opoku, and Steven Arthur. 2023. Afrisenti: A twitter sentiment analysis benchmark for african languages.

Ravindra Nayak and Raviraj Joshi. 2022. L3cube-hingcorpus and hingbert: A code mixed hindi-english dataset and bert language models.

Innocent Ndubuisi-Obi, Sayan Ghosh, and David Jurgens. 2019. Wetin dey with these comments? modeling sociolinguistic factors affecting code-switching behavior in Nigerian online discussions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6204–6214, Florence, Italy. Association for Computational Linguistics.

Nancy Chiagolum Odiegwu. 2022. Review of current trends in nigerian pidgin english. a sociolinguistic perspective. *Corpus Pragmatics*, 6:89 – 93.

Kelechi Ogueji, Yuxin Zhu, and Jimmy Lin. 2021. Small data? no problem! exploring the viability of pretrained multilingual language models for low-resourced languages. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 116–126, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wuraola Fisayo Oyewusi, Olubayo Adekanmbi, and Olalekan Akinsande. 2020. Semantic enrichment of nigerian pidgin english for contextual sentiment classification. *CoRR*, abs/2003.12450.

Braja Gopal Patra, Dipankar Das, and Amitava Das. 2018a. Sentiment analysis of code-mixed indian languages: An overview of sail_code-mixed shared task @icon-2017. *ArXiv*, abs/1803.06745.

Braja Gopal Patra, Dipankar Das, and Amitava Das. 2018b. Sentiment analysis of code-mixed indian languages: An overview of sail_code-mixed shared task @icon-2017. *CoRR*, abs/1803.06745.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Krithika Ramesh, Sunayana Sitaram, and Monojit Choudhury. 2023. Fairness in language models beyond english: Gaps and challenges.

Koustav Rudra, Shruti Rijhwani, Rafiya Begum, Kalika Bali, Monojit Choudhury, and Niloy Ganguly. 2016. Understanding language preference for expression of opinion and sentiment: What do Hindi-English speakers do on Twitter? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1131–1141, Austin, Texas. Association for Computational Linguistics.

Balogun Sarah and Murana Muniru Oladayo. 2021. Code-switching and code mixing in the selected tracks of the hip hop music of flavour and 9ice. *International Journal of English and Comparative Literary Studies*, 2(3):55–70.

Sagor Sarkar. 2020. Code switch.

Aditya Shah and Chandresh Maurya. 2021. How effective is incongruity? implications for code-mixed sarcasm detection. In *Proceedings of the 18th International Conference on Natural Language Processing (ICON)*, pages 271–276, National Institute of Technology Silchar, Silchar, India. NLP Association of India (NLPAI).

Vinay Singh, Deepanshu Vijay, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. Named entity recognition for Hindi-English code-mixed social media text. In *Proceedings of the Seventh Named Entities Workshop*, pages 27–35, Melbourne, Australia. Association for Computational Linguistics.

Sunayana Sitaram, Khyathi Raghavi Chandu, Sai Krishna Rallabandi, and Alan W. Black. 2019. A survey of code-switched speech and language processing. *CoRR*, abs/1904.00784.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Thamar Solorio, Shuguang Chen, Alan W. Black, Mona Diab, Sunayana Sitaram, Victor Soto, Emre Yilmaz, and Anirudh Srinivasan, editors. 2021. *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*. Association for Computational Linguistics, Online.

Thamar Solorio, Monojit Choudhury, Kalika Bali, Sunayana Sitaram, Amitava Das, and Mona Diab, editors. 2020. *Proceedings of the The 4th Workshop on Computational Approaches to Code Switching*. European Language Resources Association, Marseille, France.

Zeerak Talat, Aurélie Névéol, Stella Biderman, Miruna Clinciu, Manan Dey, Shayne Longpre, Sasha Luccioni, Maraim Masoud, Margaret Mitchell, Dragomir Radev, Shanya Sharma, Arjun Subramonian, Jaesung Tae, Samson Tan, Deepak Tunuguntla, and Oskar Van Der Wal. 2022. You reap what you sow: On the challenges of bias evaluation under multilingual settings. In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pages 26–41, virtual+Dublin. Association for Computational Linguistics.

Sarah G. Thomason. 2001. *Language Contact*. Edinburgh University Press, Edinburgh.

Rob van der Goot, Ahmet Üstün, Alan Ramponi, Ibrahim Sharaf, and Barbara Plank. 2021. Massive choice, ample tasks (MaChAmp): A toolkit for multi-task learning in NLP. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 176–197, Online. Association for Computational Linguistics.

Deepanshu Vijay, Aditya Bohra, Vinay Singh, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. A dataset for detecting irony in hindi-english code-mixed social media text. In *EMSASW@ESWC*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Genta Indra Winata, Alham Fikri Aji, Zheng Xin Yong, and Thamar Solorio. 2022. The decades progress on code-switching research in nlp: A systematic survey on trends and challenges. *ArXiv*, abs/2212.09660.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing.

Shijie Wu and Mark Dredze. 2020. Are all languages created equal in multilingual bert? *CoRR*, abs/2005.09093.

George-Eduard Zaharia, George-Alexandru Vlad, Dumitru-Clementin Cercel, Traian Rebedea, and Costin Chiru. 2020. UPB at SemEval-2020 task 9: Identifying sentiment in code-mixed social media texts using transformers and multi-task learning. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1322–1330, Barcelona (online). International Committee for Computational Linguistics.

A Appendix

A.1 MasakhaNER Results

MasakhaNER	
Model	**Score**
BERT	0.89
RoBERTa	0.89
mBERT	0.90
XLM-R	**0.91**
AfriBERTa	0.89
AfroXLMR	0.90

Table 7: Span-f1 scores for MasakhaNER Naija. These results are consistent with those reported in Section 5.2.

A.2 Language ID Results for Other Datasets

Language ID results for the other datasets are reported here. The tables below contain the percentage of *mostly English* and *mostly Not-English* examples that each PLM correctly classified.

AfriSenti		
	Mostly English	**Mostly Not-Eng**
Proportion	97.58%	2.42%
BERT	76.80%	90.32%
RoBERTa	76.32%	80.65%
mBERT	75.52%	80.65%
XLM-R	77.44%	77.42%
AfriBERTa	74.88%	83.87%
AfroXLMR	77.92%	74.19%
NaijaVader		
	Mostly English	**Mostly Not-Eng**
Proportion	91.79%	8.21%
BERT	72.61%	86.09%
RoBERTa	72.14%	84.35%
mBERT	72.22%	80.00%
XLM-R	72.61%	86.96%
AfriBERTa	70.97%	80.87%
AfroXLMR	72.68%	83.48%
TamilCodeMix		
	Mostly English	**Mostly Not-Eng**
Proportion	35.99%	64.01%
BERT	71.56%	69.54%
RoBERTa	72.69%	69.54%
mBERT	72.46%	69.04%
XLM-R	72.46%	69.16%
MuRiL	72.23%	68.65%
IndicBERT	72.91%	69.42%
MalayalamCodeMix		
	Mostly English	**Mostly Not-Eng**
Proportion	19.58%	80.42%
BERT	75.53%	70.21%
RoBERTa	76.60%	72.80%
mBERT	79.79%	71.50%
XLM-R	81.91%	72.02%
MuRiL	80.85%	70.98%
IndicBERT	78.72%	71.76%
DravidianCodeMix (Kannada)		
	Mostly English	**Mostly Not-Eng**
Proportion	31.40%	68.60%
BERT	69.90%	64.44%
RoBERTa	73.30%	68.44%
mBERT	68.93%	70.00%
XLM-R	66.99%	67.33%
MuRiL	70.39%	70.22%
IndicBERT	74.75%	69.11%
DravidianCodeMix (Tamil)		
	Mostly English	**Mostly Not-Eng**
Proportion	26.73%	73.27%
BERT	71.29%	63.00%
RoBERTa	71.38%	62.42%
mBERT	69.69%	63.07%
XLM-R	70.93%	63.75%
MuRiL	68.18%	62.35%
IndicBERT	68.53%	62.87%
DravidianCodeMix (Malayalam)		
	Mostly English	**Mostly Not-Eng**
Proportion	13.47%	86.53%
BERT	80.24%	75.52%
RoBERTa	80.24%	73.89%
mBERT	78.63%	74.58%
XLM-R	80.65%	74.89%
MuRiL	82.66%	75.58%
IndicBERT	84.27%	74.51%

Table 8: Proportion of *mostly English* and *mostly Not-Eng* examples in the dev sets, and the proportion of correctly classified examples by the models for each dev set.