Fine-Tuning Cross-Lingual LLMs for POS Tagging in Code-Switched Contexts

Shayaan Absar
School of Informatics
University of Edinburgh, United Kingdom
m.s.absar@sms.ed.ac.uk

Abstract

Code-switching (CS) involves speakers switching between two (or potentially more) languages during conversation and is a common phenomenon in bilingual communities. The majority of NLP research has been devoted to mono-lingual language modelling. Consequentially, most models perform poorly on code-switched data. This paper investigates the effectiveness of Cross-Lingual Large Language Models on the task of POS (Part-of-Speech) tagging in code-switched contexts, once they have undergone a fine-tuning process. The models are trained on code-switched combinations of Indian languages and English. This paper also seeks to investigate whether fine-tuned models are able to generalise and POS tag code-switched combinations that were not a part of the fine-tuning dataset.

Additionally, this paper presents a new metric, the S-index (Switching-Index), for measuring the level of code-switching within an utterance.

1 Introduction

1.1 Background

At present, approximately half of the world's population is bilingual and increased globalisation and migration is creating more multilingual communities. (Stavans and Porat, 2019). Consequently, code-switching is becoming an increasingly common form of communication, especially in online media.

Code-switching in digital and face-to-face communication can arise for a multitude of reasons including quoting someone, excluding a particular person or group from a conversation and emphasising group identity (Grosjean, 1997).

1.2 Code Switching

Code Switching is not simply alternating between two languages. Instead, it involves the fusion of two different languages which gives rise to unique grammatical constructs that are not present in either of the original languages (Attia and Elkahky, 2019). This means that mono-lingual models cannot simply be combined to produce models that are capable of dealing with CS. Additionally, CS can occur at the level of individual morphemes within a single word. This can result in frequent out-of-vocabulary words.

Often in CS, asymmetry arises (Joshi, 1982) whereby one language is more dominant compared to the other. The dominant language is referred to as the Matrix Language (ML) and the other as the Embedded Language (EL). It has been proposed (Joshi, 1982) that CS can be modelled with two grammars representing the ML and EL where a mechanism can be used to shift control from the ML to EL but not vice-versa (Martinez, 2020).

Alternatively, CS between two specific languages can be modelled as its own language (Çetinoğlu et al., 2016). For inter-sentential CS, the model can be trained on mono-lingual data from both languages. For intra-sentential CS, specific CS datasets must be obtained as the language of tokens may change within a sentence.

1.3 POS Tagging

POS (Part-of-Speech) Tagging is the task of predicting the part-of-speech of a word given its context. Complexity arises due to the fact that the same token can have different meanings and different parts of speech when used in different contexts.

This paper uses the base version of XLM-RoBERTa (Conneau et al., 2020), a cross-lingual language model trained on data from 100 different languages. The model was fine-tuned to predict part-of-speech tags. Previous attempts at this idea (Maksutov et al., 2021) involve modelling the task as a sequence-to-sequence task to generate a tag for each word in the input sequence. It is important to note here that the output vocabulary for the transformer is incredibly small compared to the

Proceedings of the Third Workshop on Resources and Representations for Under-Resourced Languages and Domains (RESOURCEFUL 2025), pages 7–12
March 2, 2025 ©2025 Association for Computational Linguistics

input vocabulary. The output vocabulary is the set of possible part of speech tags, whereas the input vocabulary is the set of all words that appear in the training dataset.

The BERT architecture (Devlin et al., 2019) is highly appropriate for this as the Masked Language Model objective used during pre-training, allows the model to learn bi-directional context. This should enable the model to more easily understand the sequences passed to it.

2 Measuring Code-Switching

2.1 Current Metrics

As previously mentioned, the Matrix Language is the dominant language in a code-switched text.

$$L_{matrix}(s) = \arg\max_{L_i \in \mathbb{L}} \{t_{L_i}\}(s) \tag{1}$$

($L_i \in \mathbb{L}$ iterates through each language in the corpus, $\{t_{L_i}\}(s)$ returns the number of tokens of language L_i in sequence s, L_{matrix} is the matrix language)

The Code-Mixing Index metric (Gambäck and Das, 2016) can be used to measure the level of code switching in a sequence s-

$$CMI(s) = \frac{\lambda(N(s) - \{t_{L_{matrix}}\}(s)) + \mu P(s)}{N(s)} \tag{2}$$

($N(s)$ is the number of tokens in the sequence, $\{t_{L_{matrix}}\}(s)$ is the number of tokens in the matrix language, $P(s)$ is the number of code alteration points and λ and μ are weights that sum to 1)

If a sequence has a high number of tokens not in the matrix language, it has a high amount of code-switching. The sequence also has a high amount of code-switching if there are a large number of alteration points. This measurement manages to capture both of these metrics.

This metric can exaggerate the level of code-switching in short sequences since it divides by the length of the sequence. This is particularly prominent in sequences with a single word followed by punctuation. This arises since punctuation is often listed as a language of its own (e.g. 'universal'). Therefore a sequence such as 'What?' is calculated as having a high-level of code-switching since there is one alteration point and one token not in the matrix language in a sequence with only two tokens.

2.2 Proposed New Metric

To solve this problem, this paper introduces the S-index measure ($\mathcal{S}$) using the same two metrics as the CMI.

$$\mathcal{S}(s) = \lambda \tanh(\mu P(s)) \times \log\left(\frac{N(s)}{\{t_{L_{matrix}}\}(s)}\right) \tag{3}$$

(λ and μ are arbitrary constants. The values in this paper use $\lambda = 1$ and $\mu = 0.5$)

Since this metric does not divide by the number of tokens in the sequence and a logarithm is applied to the ratio of tokens to tokens in the matrix language, the exaggeration for short sequences is prevented. The use of the hyperbolic tangent, limits the influence of $P(s)$ for very long sequences (preventing the opposite form of exaggeration), since it naturally saturates for large values. The constants λ and μ can be used to adjust when and to what value the $P(s)$ term saturates.

Token	**Language**
Matlab	Hindi
?	Universal
Translation	Meaning?
$N(s)$	2
$\{t_{L_{max}}\}(s)$	1
$P(s)$	1
$CMI(s)$	0.5
$\mathcal{S}(s)$	0.32

Table 1: CMI exaggerates the level of code-switching here.

It is clear that the sequence in Table 2 has a higher level of code-switching than the sequence in Table 1. However, the CMI metric fails to capture this but the S-index does.

3 Training

3.1 Dataset

We utilise a dataset consisting of code-switched social media posts and messages in three different language combinations (Jamatia et al., 2015) that was used for the ICON 2016 shared NLP task. Table 3 details the make-up of the dataset and the Code-Mixing Index and S-Index for each language pair. For the entire dataset, Pearson's Correlation Coefficient (r) (Lee Rodgers and Nicewander, 1988) between the CM-Index and S-Index was 0.85. This indicates that there is a generally strong positive

Token	Language
I	English
mean	English
.	Universal
Ye	Hindi
bol	Hindi
ri	Hindi
thi	Hindi
ki	Hindi
unki	Hindi
pics	English
do	Hindi
Translation	I mean. She was saying to give her pictures.
$N(s)$	11
$\{t_{L_{max}}\}(s)$	7
$P(s)$	3
$CMI(s)$	0.36
$\mathcal{S}(s)$	0.41

Table 2: CMI undervalues the level of code-switching here.

correlation between the two measures, yet also shows that there is significant cases where they differ and where we believe the S-index resolves some of the flaws of the CM-Index.

3.2 POS-Tagging with BERT models

The tokenizers used by BERT models (and many other Large Language Models) often produce multiple tokens per word (Schuster and Nakajima, 2012). This means that when assigning POS-tags, complexity arises, as each POS-tag can be associated with multiple tokens. Some simple solutions to this problem (Saidi et al., 2021) include assigning the POS tag to the first sub-word token of each word and assigning the same POS tag to each sub-word token. The solution implemented here is to pass each sub-word token into the model, producing a context-aware embedding for each sub-word token. These are then re-aligned to the word level by taking the average embedding for words that consist of more than one token (Lauren, 2022).

The use of sub-word tokenizers can be viewed as a benefit in the case of code-switching as it enables the model to more effectively deal with out-of-vocabulary words (Nayak et al., 2020).

Here, the POS-tagging task is modelled as a sequence-to-sequence task. Upon passing a sequence to the model, a tag is generated for each token in the input sequence.

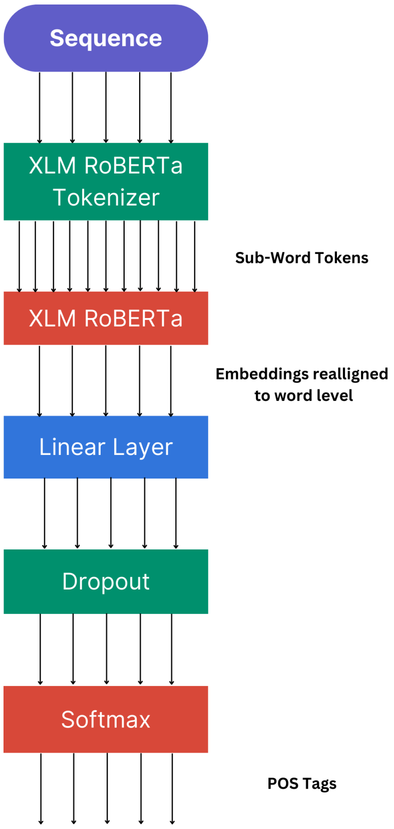

Figure 1: Model Architecture

3.3 Model

The sequences are tokenized using the XLM-RoBERTa tokenizer and then passed into XLM-RoBERTa which produces a high-dimensional embedding of each token in the input sequence. This embedding passes through a linear layer and finally, a softmax operation to transform it into a low-dimensional probability distribution, indicating the likelihood of each token belonging to different part-of-speech tags.

We utilise dropout layers between the output of XLM-RoBERTa and the linear layer to reduce the effects of overfitting during training.

3.4 Fine-Tuning

We fine-tuned four XLM-RoBERTa models on different language pair combinations: (1) HI-EN, TE-EN, and BE-EN; (2) HI-EN and TE-EN; (3) HI-EN and BE-EN; and (4) TE-EN and BE-EN. The purpose of this was to investigate whether the models were capable of generalising the POS-tagging process to language combinations that were not present in the dataset used for fine-tuning. Previous studies (Blum, 2022) have evaluated the effectiveness of

Languages	Mean CM-Index	Mean S-Index	Count
English-Hindi	0.405	0.583	1867
English-Bengali	0.507	0.776	625
English-Telugu	0.503	0.792	1487
Overall	0.458	0.692	3979

Table 3: Statistics for each language combination in the dataset.

fine-tuned multilingual language models for POS tagging in languages that were absent from the fine-tuning dataset, specifically in contexts without code-switching.

We employ the use of a learning rate scheduler and the AdamW (Loshchilov and Hutter, 2019) optimiser during the fine-tuning process.

3.5 Performance on Unseen Combinations during Fine-Tuning

Figure 3 shows the performance of the models on the hidden language combination during training. Despite the fine-tuning dataset containing no data from the respective languages, it is clear that the performance improves significantly during the fine-tuning process.

One cause of this property is the overlap between the subword tokens found in the training dataset and the hidden language datasets. Therefore, the model is still indirectly exposed to some of the same tokens, improving its performance. Experiments (Pires et al., 2019) show that when tested in this way, fine-tuned multilingual models do not solely rely on an overlap between tokens (which would indicate learning through simple vocabulary memorisation) and that the pretraining process has enabled more robust multilingual representations.

However, the loss values for the hidden languages do not reach as low as the validation loss (only containing the language combinations visible in the fine-tuning process) as shown in Figure 2. It is unclear whether this is due to the small size of the model and the lack of data (Kaplan et al., 2020) or if there is a hypothetical limit on the performance on hidden languages when models are fine-tuned in this way.

A cause of this limit could be catastrophic forgetting (McCloskey and Cohen, 1989) whereby the model loses some of its ability to understand the languages that appeared during pre-training when fine-tuned on the other languages.

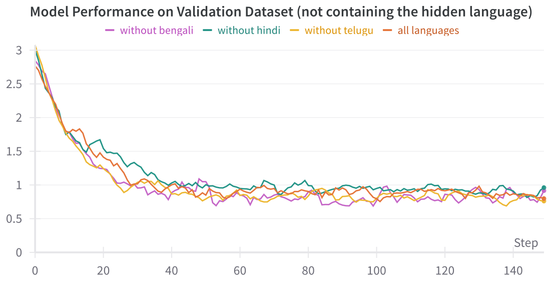

Figure 2: The performance of the model on the validation dataset (containing data from the languages the model is fine-tuned on) during the fine-tuning process.

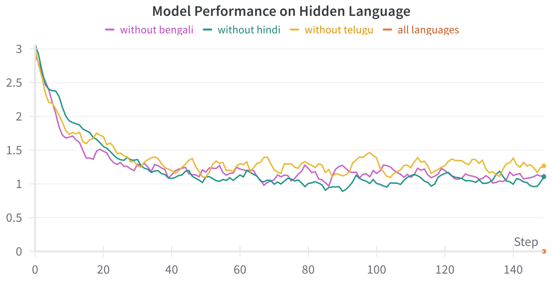

Figure 3: The performance of the model on the data from the language that is not contained in the fine-tuning dataset.

4 Results

The fine-tuned models were tested on a portion of the dataset. The results are shown in Table 4. The testing shows that the models were able to predict POS-tags with a reasonable degree of accuracy. We feel that the performance of the models is highly promising given that the language model used only has 279 million trainable parameters and only a small dataset was used.

4.1 Performance on Unseen Code-Switched Combinations

The testing shows that the models are capable of predicting POS-tags in unseen language combinations to a similar level of accuracy as to when these combinations are included in the fine-tuning dataset.

The fact that Bengali, Hindi, and Telugu are all

	% of tokens correctly predicted			
Combinations Trained On	**HI-EN**	**TE-EN**	**BE-EN**	**Overall**
HI-EN, TE-EN, BE-EN	76.54	71.86	73.75	74.53
HI-EN, TE-EN	78.67	74.32	67.68	70.28
HI-EN, BE-EN	77.80	67.90	75.32	69.60
TE-EN, BE-EN	72.14	73.15	77.90	72.40

Table 4: The % of tokens in the test dataset that each model correctly predicted.

Indian languages with shared grammatical features likely contributes to this ability. Moreover, the consistent subject-object-verb (SOV) word order across these languages helps in POS tagging by providing a similar syntactic structure.

However, Telugu belongs to a different language family (Dravidian) than Bengali and Hindi (Indo-European) which introduces some variance. This would suggest that the models are capable of learning more general syntactic patterns that appear across different languages. To determine whether this ability persists in other code-switched language combinations would require further experiments. Unfortunately, the current lack of suitable datasets presents a challenge to conducting such investigations.

When the HI-EN data is removed, the performance on this language combination improves significantly compared to when other language pairs are removed. This is likely because Hindi, being the most widely spoken language in India, is often mixed into other language pairs. This pattern was observed by the creators of the dataset[1].

5 Conclusion and Future Work

Although the performance of the models trained here is not comparable to those of today's state-of-the-art POS taggers, we feel that our models are highly promising.

The ability of models to POS-tag in unseen code-switched combinations is evident and more research needs to be performed to analyse whether this property extends to other code-switched language combinations that are not so closely related.

Additionally, the ability of multilingual models to be fine-tuned to perform other NLP tasks such as Sentiment Analysis and Named Entity Recognition is also an area that needs to be researched.

Limitations

This study was limited to a small number of code-switched combinations between English and three Indian languages, due to a lack of widely available datasets.

Furthermore, we noted a small yet significant discrepancy between the performance of the models on code-switched combinations that were included in the fine-tuning dataset and those that were not. We feel that more research needs to be done on the causes of this discrepancy and how they can be limited.

References

Mohammed Attia and Ali Elkahky. 2019. Segmentation for domain adaptation in Arabic. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 119–129, Florence, Italy. Association for Computational Linguistics.

Frederic Blum. 2022. Evaluating zero-shot transfers and multilingual models for dependency parsing and POS tagging within the low-resource language family tupían. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 1–9, Dublin, Ireland. Association for Computational Linguistics.

Özlem Çetinoğlu, Sarah Schulz, and Ngoc Thang Vu. 2016. Challenges of computational processing of code-switching. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 1–11, Austin, Texas. Association for Computational Linguistics.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of

[1] `https://amitavadas.com/Code-Mixing.html`

deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Björn Gambäck and Amitava Das. 2016. Comparing the level of code-switching in corpora. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1850–1855, Portorož, Slovenia. European Language Resources Association (ELRA).

François Grosjean. 1997. The bilingual individual. *Interpreting*, 2:163–187.

Anupam Jamatia, Björn Gambäck, and Amitava Das. 2015. Part-of-speech tagging for code-mixed English-Hindi Twitter and Facebook chat messages. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 239–248, Hissar, Bulgaria. INCOMA Ltd. Shoumen, BULGARIA.

Aravind K. Joshi. 1982. Processing of sentences with intra-sentential code-switching. In *Coling 1982: Proceedings of the Ninth International Conference on Computational Linguistics*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models.

Paula Lauren. 2022. Reconstructing word representations from pre-trained subword embeddings. In *2022 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 35–40.

Joseph Lee Rodgers and W Alan Nicewander. 1988. Thirteen ways to look at the correlation coefficient. *The American Statistician*, 42(1):59–66.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization.

Artem A. Maksutov, Vladimir I. Zamyatovskiy, Viacheslav O. Morozov, and Sviatoslav O. Dmitriev. 2021. The transformer neural network architecture for part-of-speech tagging. In *2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)*, pages 536–540.

Victor Soto Martinez. 2020. *Identifying and modeling code-switched language*. Columbia University.

Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.

Anmol Nayak, Hariprasad Timmapathini, Karthikeyan Ponnalagu, and Vijendran Gopalan Venkoparao. 2020. Domain adaptation challenges of BERT in tokenization and sub-word representations of out-of-vocabulary words. In *Proceedings of the First Workshop on Insights from Negative Results in NLP*, pages 1–5, Online. Association for Computational Linguistics.

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.

Rakia Saidi, Fethi Jarray, and Mahmud Mansour. 2021. A bert based approach for arabic pos tagging. In *Advances in Computational Intelligence: 16th International Work-Conference on Artificial Neural Networks, IWANN 2021, Virtual Event, June 16–18, 2021, Proceedings, Part I 16*, pages 311–321. Springer.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152.

Anat Stavans and Ronit Porat. 2019. Codeswitching in multilingual communities. *Multidisciplinary Perspectives on Multilingualism: The Fundamentals*, 19:123.