Switch Point biased Self-Training: Re-purposing Pretrained Models for Code-Switching

Parul Chopra, Sai Krishna Rallabandi, Alan W Black, Khyathi Raghavi Chandu
Language Technologies Institute
Carnegie Mellon University
{parulcho, srallaba, awb, kchandu}@cs.cmu.edu

Abstract

Code-switching (CS), a ubiquitous phenomenon due to the ease of communication it offers in multilingual communities still remains an understudied problem in language processing. The primary reasons behind this are: (1) minimal efforts in leveraging large pretrained multilingual models, and (2) the lack of annotated data. The distinguishing case of low performance of multilingual models in CS is the intra-sentence mixing of languages leading to switch points. We first benchmark two sequence labeling tasks – POS and NER on 4 different language pairs with a suite of pretrained models to identify the problems and select the best performing model, char-BERT, among them (addressing (1)). We then propose a self training method to repurpose the existing pretrained models using a switch-point bias by leveraging unannotated data (addressing (2)). We finally demonstrate that our approach performs well on both tasks by reducing the gap between the switch point performance while retaining the overall performance on two distinct language pairs in both the tasks. We plan to release our models and the code for all our experiments.

1 Introduction

Code-switching (CS) is a phenomenon of switching back and forth between multiple languages and is very common in multilingual communities such as India, Singapore, etc. Understanding mixed language texts has several applications in an increasingly online world like hateful content detection, maintaining engagement with virtual assistants. Despite this pervasive prevalence, CS is often overlooked in language processing research and current models still cannot effectively handle CS. We believe that the reasons behind this are (1) the lack of efforts in leveraging existing large scale multilingual resources or pretrained models and (2) dearth of annotated resources in switching scenarios. In this paper, we present solutions to address these two problems specifically.

The advent of pretraining techniques marshalled the celebrated successes of several language understanding and generation tasks in English (Dong et al., 2019) and multilingual tasks (Chaudhary et al., 2020). However, the same level of commendatory results are not translated to CS scenarios; as studied by Aguilar et al. (2020); Khanuja et al. (2020) presenting a preliminary evaluation of multi-lingual pretrained models for CS scenarios. It is still largely unclear if the inadequacies are resulting due to dearth of data or ineptitude of quick adoption of multilingual models. We study precisely this problem of identifying the artifacts that hinder the competent performance of pretrained models on CS with a case study on sequence labeling tasks including Part-Of-Speech (POS) tagging and Named Entity Recognition (NER).

Our contributions from this work are as follows: (1) We first conduct a comprehensive benchmarking of different pretrained models for two sequence labeling tasks across 4 different language pairs. Specifically we evaluate datasets in Hinglish, Tenglish, Benglish and Spanglish CS for the tasks NER and POS. (2) To broaden understanding towards the usefulness of different fine-tuning strategies, we investigate multitasking, character modeling uncovering the problematic switch point cases in §4. (3) We propose a novel *switch-point bias based self training approach* built upon on observations from the benchmarks and demonstrate improved results on both tasks.

2 Related Work

CS benchmarks: From one of the recent surveys (Sitaram et al., 2019), linguistic CS has been studied in the context of many NLP tasks including language identification (Solorio et al., 2014) (Bali et al., 2014), POS tagging (Soto and Hirschberg, 2018) (Molina et al., 2019) (Das, 2016), NER

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 4389–4397
November 7–11, 2021.

(Aguilar et al., 2019), parsing (Partanen et al., 2018), sentiment analysis(Vilares et al., 2015), and question answering (Chandu et al., 2019) (Raghavi et al., 2015). Many CS datasets have been made available through the shared-task series FIRE (Choudhury et al., 2014); (Roy et al., 2013) and CALCS (Aguilar et al., 2018), which have focused mostly on core NLP tasks. Additionally, other researchers have provided datasets such as humor detection (Khandelwal et al., 2018), sub-word CS detection (Mager et al., 2019) among others. More recently new CS benchmarks (Aguilar et al., 2020) (Khanuja et al., 2020) have been developed to compare models across language pairs, domains and general language processing in CS.

Pretrained Models for CS: Before the advent of pretrained multingual models, pretrained monolingual models were combined in different ways to derive word embeddings (AlGhamdi and Diab, 2019; Pratapa et al., 2018), POS tagging (Bhattu et al., 2020), sentiment analysis (Singh and Lefever, 2020) etc., Similarly, pretrained multilingual models have been explored on various CS tasks like language identification, POS tagging, NER, question answering and Natural language inference (Khanuja et al., 2020). However, (Winata et al., 2021) show that these pretrained models do not assure high quality representations on CS. We examine prospective reasons for this and present a data augmentation technique to mitigate this.

Motivation for our work - Gaps in CS adaptation: Building off the prior work, we will briefly discuss primarily three techniques that demonstrated usefulness in adapting models to CS. First, non-standardization of cross-scripting (i.e, transliteration of words to another language) is identified as one of the major reasons behind the noisiness of CS datasets (Chandu et al., 2019). Prior literature on noisy texts proved the superiority of character level modeling to combat this problem (Cherry et al., 2018); (Adouane et al., 2018). Secondly, the domains of most of these noisy datasets are still vastly scattered. In order to improve generalization in CS patterns, prior studies have shown the potency of multitasking with an auxiliary task of language tag prediction (Winata et al., 2018). Thirdly, the dearth of annotated CS data has been a dramatic problem across tasks. (Bhattu et al., 2020) compare pretrained models with fined-tuned models augmented with unlabeled Twitter text to exemplify the improved performance with the latter model. Despite these takeaways, the usefulness of the three points above is not thoroughly investigated in the context of pretrained models for CS. To this end, we adapt these techniques in conjunction with the pretraining strategies and propose a novel bias-based data iterative augmentation technique to get more bang for the buck in terms of the performance to augmented dataset size ratio.

Corpus	Notation	Task	# Sentences
Twitter (Singh et al., 2018a)	**EnHi-Tw-P**	POS	1489
UD (Bhat et al., 2018)	**EnHi-UD-P**	POS	1311
ICON (Jamatia et al., 2016)	**EnHi-I-P**	POS	2630
ICON (Jamatia et al., 2016)	**EnBn-I-P**	POS	625
ICON (Jamatia et al., 2016)	**EnTe-I-P**	POS	1979
Miami (AlGhamdi et al., 2019)	**EnEs-M-P**	POS	27893
Twitter (Singh et al., 2018b)	**EnHi-Tw-N**	NER	1243
CALCS (Aguilar et al., 2019)	**EnEs-Tw-N**	NER	50757

Table 1: Details of CS datasets & training sizes

3 Benchmarking Multilingual Pretrained Models

3.1 Datasets and Models

We selected datasets from LinCE(Aguilar et al., 2020) and GlueCOS (Khanuja et al., 2020) benchmarks for all our experiments. The details of these datasets are presented in Table 1. We present a comprehensive evaluation of different BERT-based mono-lingual and multi-lingual pretrained models when adapted to the chosen CS datasets/tasks. We performed sequence tagging on different transformer models: (a) We use the uncased base implementation of **BERT** and **mBERT** (Devlin et al., 2018) (b) **Distill mBERT** (Sanh et al., 2019), (c) **XLM-RoBERTa** (Conneau et al., 2019) trained using knowledge distillation and (d) **Char-BERT**(Boukkouri et al., 2020) that employs Character CNN to capture unknown and misspelled words. Motivated by prior works on multi-task learning (Chandu et al., 2018; Li et al., 2020), we also experiment with language-aware modeling. In these experiments, we added a language token either as the *input encoding* or *output prediction*.

3.2 Analysis of Benchmarking

The results for the aforementioned experiments are presented in Table 2. The baseline in this table indicates the current state-of-the-art models on respective datasets as cited in the table. Here are our main observations from these results.

• *Multi-task Learning did not help much:* Despite the effectiveness of multi-tasking in non-pretrained

Model	Part-Of-Speech tagging						Named Entity Recognition	
	EnHi-Tw-P	EnHi-UD-P	EnEs-M-P	EnHi-I-P	EnBn-I-P	EnTe-I-P	EnHi-Tw-N	EnEs-Tw-N
Baseline	91.03 (A)	90.53 (B)	95.39(B)	85.26 (C)	77.15 (C)	74.88 (C)	78.21 (B)	69.17 (B)
eng-BERT	84.01	82.12	91.77	80.55	75.78	76.11	65.93	55.12
M-BERT	89.27	87.67	93.12	86.38	80.74	79.01	74.2	60.12
M-BERT (lang-input)	89.74	87.96	93.65	86.99	81.67	78.55	75.38	61.46
M-BERT (lang-output)	88.89	86.47	92.89	85.65	81.17	76.13	74.01	60.20
Distill M-BERT	90.28	88.19	93.65	86.92	82.07	79.85	67.26	62.67
XLM-ROBERTa	90.74	89.88	95.34	86.24	80.58	75.83	73.34	66.12
char-BERT	90.89	90.23	96.88	87.11	82.21	80.33	77.24	65.72
char-BERT (lang-input)	91.02	90.93	97.01	87.24	82.87	82.52	77.43	66.34
char-BERT (lang-output)	90.25	89.29	96.25	86.39	82.47	80.98	77.12	66.01

Table 2: Performance of different multilingual models for various POS tagging datasets (Accuracy), NER (F1) in single , multi-task setting (language at input/output). Results are reported for datasets- (A) (Aguilar and Solorio, 2020), (B) (Khanuja et al., 2020), (C) (Bhattu et al., 2020)

(models trained from scratch) CS modeling, vast improvements are not observed upon finetuning pretrained with multitasking objective.

• *Improvement with Char-BERT:* We observe that Char-BERT gives significant improvement in POS specifically for Indic sets: English-Bengali and English-Telugu. On others, its performance is comparable to current SOTA with mBERT or XLM-RoBERTa. Although the languages in the pretraining of mBERT include the language pairs of concern here, we do not observe benefits from this model as the training data mostly includes data from the script of the source language. For example, training on Devanagari Hindi does not necessarily translate its ability to understand the cross-scripted and usually Romanized CS texts.

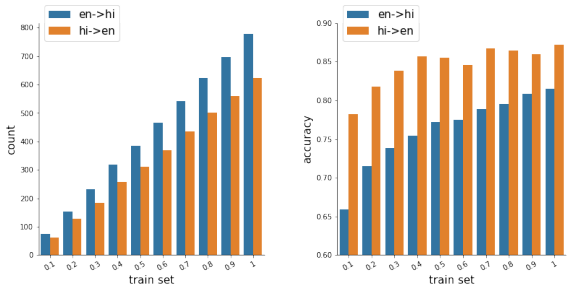
• *Performance at switch-points:* We further investigated the performance at switch-points which distinguishes CS from monolingual texts. We demonstrate this for *EnHi-Tw-P* in Figure 1, where the validation accuracy of switching from English to Hindi (en → hi) is relatively much lower compared to switching from Hindi to English (hi → en). We observe this pattern to be consistent across the datasets in Table 3 and propose a solution to address this in the next section.

4 Switch-Point biased Self Training

As observed in the previous section, performance of the models deteriorates at the switching points (Chatterjere et al., 2020) in CS. This motivates our approach to tackle this problem which can be stated concretely as:

> *The pre-trained model favors embedded-to-matrix over matrix-to-embedded language switching points despite majority of training data in the former pattern.*

We demonstrate this by comparing Figure 1(a) and Figure 1(b) for the case of *EnHi-Tw-P*. They



(a) SP count (b) SP validation accuracy

Figure 1: (a) Count and (b) accuracy over val set for different portions or percentages of training data (*EnHi-Tw-P)* for both switch-points (SP)

present the counts and val accuracy with the increasing percentage of train set on the x-axis to demonstrate the consistency of this pattern. As we can observe, the number of samples with switch points from en → hi is higher than that of hi → en (Fig 1(a)). However, the performance on switch points from en → hi is relatively much lower than the counter part (Fig 1(b)).

Algorithm 1 Switch-Point biased Self Training (selfTr)

Input: Annotator Model A(θ), Labeled Data $\mathcal{D}^l$, Unlabeled Data $\mathcal{D}^u$
Output: Trained End-Task Model $\mathcal{E}(\phi^{'})$
1. Fine-tune $\mathcal{E}(\phi)$ on $\mathcal{D}^l$
2. s ← Identify the low-performing switch-point
3. $\mathcal{D}^u_s$ ← Sub-sample data from $\mathcal{D}^u$ with higher ratio of s
4. $\mathcal{D}^{wl}_s$ ← Annotate $\mathcal{D}^u_s$ with $\mathcal{A}(\theta^{'})$
5. $\mathcal{A}(\theta^{'})$ ← Fine-tune $\mathcal{A}(\theta)$ on $\mathcal{D}^{wl}_s + \mathcal{D}^l$
6. $\mathcal{E}(\phi^{'})$ ← Train $\mathcal{E}(\phi)$ on $\mathcal{D}^{wl}_s + \mathcal{D}^l$
7. Repeat Steps 2 to 6 by updating $\mathcal{A}(\theta^{'})$ and $\mathcal{E}(\phi^{'})$

We posit that a switch point specific fine-tuning is required to combat this imbalance. Our proposed approach is depicted in Algorithm 1. The baseline for each task is the char-BERT model fine-tuned on the task-specific data, which is referred as end-task

Model	Biased Annotator	Overall		X → en		en → X	
		Acc.	F1	Acc.	F1	Acc.	F1
POS-en-hi	-	89.84	85.21	87.91	86.11	82.62	80.23
POS -en-hi-random	-	89.29	85.11	88.78	87.23	82.40	80.11
POS-en-hi-selfTr	en → hi	**89.91**	85.16	87.27	86.34	**84.38**	**85.01**
POS-en-es	-	96.88	96.25	93.82	90.97	88.59	85.38
POS -en-es-random	-	96.91	96.21	94.10	90.12	88.04	84.32
POS-en-es-selfTr	en → es	**97.05**	**96.41**	**95.51**	**93.42**	**90.6**	**88.12**
NER-en-hi	-	95.45	75.18	96.92	84.84	93.21	77.41
NER-en-hi-random	-	95.42	75.12	97.05	86.12	93.09	76.65
NER-en-hi-selfTr	en → hi	95.41	75.02	95.89	78.78	**95.02**	**80.70**
NER-en-es	-	93.00	65.72	83.28	47.73	93.35	58.19
NER-en-es-random	-	93.10	65.95	84.25	49.22	94.10	59.67
NER-en-es-selfTr	es → en	**93.12**	**66.34**	**86.13**	**56.62**	**94.58**	**62.29**

Table 3: Results of our *switch point biased self training* (selfTr). Here the annotator model is trained on subset of data which is more biased towards lower-performing switch point. The biased annotator model is trained using a subset of the data with the switch point shown in the table. X refers to the language which is mixed with English.

model $\mathcal{E}(\phi)$.

Our first step is to compute switch point ratios. We computed the percentage of switch points from En->X (say a) and from X->En (say b) on the unlabeled data. We then compute s=a/b. If s<1, we bias our annotator model by training with the sentences that has 's greater than 1' i.e biased to En->X data, otherwise, we train it with the sentences that has 's lesser than 1' i.e biased to X->En data. In this way, our annotator model is biased to favor annotations on low-performing switch-point and is further used to annotate the unlabeled dataset.

We then identify the low-performing switch point and derive the Annotator Model A(θ) with the labeled subset of the low-performing switch point (s) from the dataset. This annotator model is now biased to favor annotations on this s to increase its bias for further annotations. We leverage a vast amount of unlabeled dataset $\mathcal{D}^u$. The unlabeled data is gathered from the validation and test subsets of the standard datasets (from Table 1) without considering the true labels. We use the raw samples i.e., sentences and annotate them using the annotator model. Based on the amount of samples available, we iteratively annotate and add samples to our original training dataset with our switch-point bias based self training.

The underlying annotator model can be any of the large scale pretrained models that we experimented with in the previous section. We choose to use char-BERT as our annotator model. This annotator model is used to annotate the subset of the unlabeled data with sequence tags. This weakly annotated noisy data is now augmented to the labeled dataset. Both the annotator model and the end-task model are now finetuned with this augmented dataset. This iterative data augmentation process repeats until the performance stops degrading.

4.1 Results

Adding the annotated data via switch point based self training helps the model better learn at low-performing code switching points. In Table 3, we evaluate this technique on 4 different datasets where we train both our model and annotator by fine-tuning a character-BERT model (as we observed improvements with this model in Section 3.2). Note that X refers to the language which is mixed with English. We can see that among the char-BERT baseline (first row in each segment of the table), the performance is highly biased both in terms of F1 and accuracy towards: (i) switching to English (X → en switch point) in the first 3 segments, and (ii) switching to Spanish (en → X switch point) in the last segment. Accordingly we train annotator models described above and augment the training data. To evaluate the effectiveness of our approach, we also compare these results to the case when annotator model is updated by training with augmented data selected randomly of the same size. It can be seen that our bias based approach performs better than uninformed random data augmentation for training. Our approach demonstrates consistent improvements at the low-performing switch points. The difference in switch-point F1 scores between X → en F1 and en → X F1 compared between the baseline char-BERT and our approach is reduced by a margin of 5%, 3%, 6% and 5% on POS English-Hindi (Singh et al., 2018a), POS English-Spanish (Al-Ghamdi et al., 2019), NER English-Hindi (Singh et al., 2018b) and NER English-Spanish (Aguilar

Sentence	enjoying	garam	garam	alu	paratha	and	kullad	ki	chai
LID	en	hi	hi	hi	hi	en	hi	hi	hi
Ground Truth Labels	verb	adj	adj	noun	noun	conj	noun	adp	noun
Random Annotator	verb	adj	adj	noun	noun	conj	adj	adp	noun
Switch-point Biased Annotator	verb	adj	adj	noun	noun	conj	noun	adp	noun

Figure 2: Example of model predictions from Random Annotator model and Switch-point biased Annotator model. *(Meaning of the example sentence: Enjoying hot potato bread and kullad tea.)*

et al., 2019) respectively. In this way, we also improved the overall accuracy and F1 in 3 and 2 datasets respectively, while the scores remained almost the same for 1 and 2 datasets correspondingly.

Figure 2 presents an example sentence from Hindi-English code-switched POS data along with language ids along with ground truth labels and predictions. The random annotator model incorrectly predicts '*kullad*' as *adj* when transitioning from English to Hindi (en $\rightarrow$ X). Our switch-point biased based model correctly labels this word.

Analysis: An inspection of pretrained models revealed different types of errors: (a) Errors on **NUM** when the numerals were in Hindi and (b) Confusion between the classes **PROPN** and **N** (c) Errors due to misspelled words and (d) logical errors due to ambiguous sentences. In general we observed some noise in the dataset labels itself.

We also conducted a categorical error analysis of the performance on one of the language pairs that is Hindi-English data. In this language mixing, for example, we noticed that when switching from X $\rightarrow$ En, the errors are significantly higher for Proper Nouns ($\sim$99%) and Interjections ($\sim$99%) as compared to other POS tags, while the reverse is the case for Determiners ($\sim$98%) and Particles ($\sim$94%). The numbers in the brackets indicate the 'absolute difference' of accuracies between En $\rightarrow$ X and X $\rightarrow$ En for predictions of the corresponding POS tag. This means that Proper Nouns and Interjections are more difficult to tag when switched from Hindi to English, but the same pattern is not observed when switched from English to Hindi.

5 Conclusions

CS, despite being a natural and prevalent form of communication is still vastly understudied in empirical research. This mainly stems from the (1) lack of efforts in re-purposing the celebrated pretrained models to CS scenarios and (2) lack of annotated resources. We tackle precisely these 2 problems with the main focus on evaluating and improving how these models fare at switch points between languages. First, we benchmark a suite of monolingual and multilingual pretrained models on CS and identify that particular switch points fare poorly. We propose a novel switch point bias based self training method to strategically use unlabeled data to enhance performance at switch points. While improving or retaining the overall performance compared to finetuning char-BERT and multitasking, we show that our approach improves the performance of underperforming switch points as well. We believe that this bias based augmentation technique particularly helps in scenarios with less annotated data.

6 Broader Impact

We believe that this work is a step towards effacing the hesitation of utilizing large scale pretrained mono and multilingual models for code-switched scenarios. We were able to successfully demonstrate the utility of a switch point based annotator model to perform biased data augmentation. We do not foresee any immediate ethical concerns branching directly from our work. However, we cautiously advise anyone using or extending our work for their application or research to bear in mind that we inherit any kinds of biases and toxicity and privacy concerns that the pretrained language models bear. Although our end tasks are not directly affected forthwith due to these, we still recommend caution when our self training approach is used for other tasks especially with user interaction such as dialog response generation etc., to ensure the model does not predict toxic content. Overall, we expect the users to benefit from our research to prospectively apply this to scenarios where there is a dearth of annotated resources, thereby economizing on annotations cost and efforts and enabling scaling up to a wealth of crawled data, if available in those language-pairs.

References

Wafia Adouane, Simon Dobnik, Jean-Philippe Bernardy, and Nasredine Semmar. 2018. A comparison of character neural language model and bootstrapping for language identification in multilingual noisy texts. In *Proceedings of the Second Workshop on Subword/Character LEvel Models*, pages 22–31.

Gustavo Aguilar, Fahad AlGhamdi, Victor Soto, Mona Diab, Julia Hirschberg, and Thamar Solorio. 2019. Named entity recognition on code-switched data: Overview of the calcs 2018 shared task. *arXiv preprint arXiv:1906.04138*.

Gustavo Aguilar, Fahad AlGhamdi, Victor Soto, Thamar Solorio, Mona Diab, and Julia Hirschberg. 2018. Proceedings of the third workshop on computational approaches to linguistic code-switching. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*.

Gustavo Aguilar, Sudipta Kar, and Thamar Solorio. 2020. LinCE: A centralized benchmark for linguistic code-switching evaluation. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1803–1813, Marseille, France. European Language Resources Association.

Gustavo Aguilar and Thamar Solorio. 2020. From english to code-switching: Transfer learning with strong morphological clues.

Fahad AlGhamdi and Mona Diab. 2019. Leveraging pretrained word embeddings for part-of-speech tagging of code switching data. *arXiv preprint arXiv:1905.13359*.

Fahad AlGhamdi, Giovanni Molina, Mona Diab, Thamar Solorio, Abdelati Hawwari, Victor Soto, and Julia Hirschberg. 2019. Part of speech tagging for code switched data. *arXiv preprint arXiv:1909.13006*.

Kalika Bali, Jatin Sharma, Monojit Choudhury, and Yogarshi Vyas. 2014. "i am borrowing ya mixing?" an analysis of english-hindi code mixing in facebook. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 116–126.

Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Manish Shrivastava, and Dipti Misra Sharma. 2018. Universal dependency parsing for hindi-english code-switching. *arXiv preprint arXiv:1804.05868*.

S Nagesh Bhattu, Satya Krishna Nunna, DVLN Somayajulu, and Binay Pradhan. 2020. Improving code-mixed pos tagging using code-mixed embeddings. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 19(4):1–31.

Hicham El Boukkouri, Olivier Ferret, Thomas Lavergne, Hiroshi Noji, Pierre Zweigenbaum, and Junichi Tsujii. 2020. Characterbert: Reconciling elmo and bert for word-level open-vocabulary representations from characters. *arXiv preprint arXiv:2010.10392*.

Khyathi Chandu, Ekaterina Loginova, Vishal Gupta, Josef van Genabith, Günter Neumann, Manoj Chinnakotla, Eric Nyberg, and Alan W Black. 2019. Code-mixed question answering challenge: Crowdsourcing data and techniques. In *Third Workshop on Computational Approaches to Linguistic Code-Switching*, pages 29–38. Association for Computational Linguistics (ACL).

Khyathi Chandu, Thomas Manzini, Sumeet Singh, and Alan W Black. 2018. Language informed modeling of code-switched text. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, pages 92–97.

Arindam Chatterjere, Vineeth Guptha, Parul Chopra, and Amitava Das. 2020. Minority positive sampling for switching points-an anecdote for the code-mixing language modeling. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 6228–6236.

Aditi Chaudhary, Karthik Raman, Krishna Srinivasan, and Jiecao Chen. 2020. Dict-mlm: Improved multilingual pre-training using bilingual dictionaries. *arXiv preprint arXiv:2010.12566*.

Colin Cherry, George Foster, Ankur Bapna, Orhan Firat, and Wolfgang Macherey. 2018. Revisiting character-based neural machine translation with capacity and compression. *arXiv preprint arXiv:1808.09943*.

Monojit Choudhury, Gokul Chittaranjan, Parth Gupta, and Amitava Das. 2014. Overview of fire 2014 track on transliterated search. *Proceedings of FIRE*, pages 68–89.

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. *arXiv preprint arXiv:1911.02116*.

Amitava Das. 2016. Tool contest on pos tagging for code-mixed indian social media (facebook, twitter, and whatsapp) text.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. *arXiv preprint arXiv:1905.03197*.

Anupam Jamatia, Björn Gambäck, and Amitava Das. 2016. Collecting and annotating indian social media code-mixed corpora. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 406–417. Springer.

Ankush Khandelwal, Sahil Swami, Syed S Akhtar, and Manish Shrivastava. 2018. Humor detection in english-hindi code-mixed social media content: Corpus and baseline system. *arXiv preprint arXiv:1806.05513*.

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.

Xiuhong Li, Zhe Li, Jiabao Sheng, and Wushour Slamu. 2020. Low-resource text classification via cross-lingual language model fine-tuning. In *China National Conference on Chinese Computational Linguistics*, pages 231–246. Springer.

Manuel Mager, Özlem Çetinoğlu, and Katharina Kann. 2019. Subword-level language identification for intra-word code-switching. *arXiv preprint arXiv:1904.01989*.

Giovanni Molina, Fahad AlGhamdi, Mahmoud Ghoneim, Abdelati Hawwari, Nicolas Rey-Villamizar, Mona Diab, and Thamar Solorio. 2019. Overview for the second shared task on language identification in code-switched data. *arXiv preprint arXiv:1909.13016*.

Niko Partanen, KyungTae Lim, Michael Rießler, and Thierry Poibeau. 2018. Dependency parsing of code-switching data with cross-lingual feature representations. In *International Workshop on Computational Linguistics for Uralic Languages*, pages 1–17. ACL.

Adithya Pratapa, Monojit Choudhury, and Sunayana Sitaram. 2018. Word embeddings for code-mixed language processing. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 3067–3072.

Khyathi Chandu Raghavi, Manoj Kumar Chinnakotla, and Manish Shrivastava. 2015. " answer ka type kya he?" learning to classify questions in code-mixed language. In *Proceedings of the 24th International Conference on World Wide Web*, pages 853–858.

Rishiraj Saha Roy, Monojit Choudhury, Prasenjit Majumder, and Komal Agarwal. 2013. Overview of the fire 2013 track on transliterated search. In *Post-Proceedings of the 4th and 5th Workshops of the Forum for Information Retrieval Evaluation*, pages 1–7.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Kushagra Singh, Indira Sen, and Ponnurangam Kumaraguru. 2018a. A twitter corpus for hindi-english code mixed pos tagging. In *Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media*, pages 12–17.

Pranaydeep Singh and Els Lefever. 2020. Sentiment analysis for hinglish code-mixed tweets by means of cross-lingual word embeddings. In *Proceedings of the The 4th Workshop on Computational Approaches to Code Switching*, pages 45–51.

Vinay Singh, Deepanshu Vijay, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018b. Named entity recognition for hindi-english code-mixed social media text. In *Proceedings of the seventh named entities workshop*, pages 27–35.

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

Thamar Solorio, Elizabeth Blair, Suraj Maharjan, Steven Bethard, Mona Diab, Mahmoud Ghoneim, Abdelati Hawwari, Fahad AlGhamdi, Julia Hirschberg, Alison Chang, et al. 2014. Overview for the first shared task on language identification in code-switched data. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 62–72.

Victor Soto and Julia Hirschberg. 2018. Joint part-of-speech and language id tagging for code-switched data. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, pages 1–10.

Genta Indra Winata, Samuel Cahyawijaya, Zihan Liu, Zhaojiang Lin, Andrea Madotto, and Pascale Fung. 2021. Are multilingual models effective in code-switching? *arXiv preprint arXiv:2103.13309*.

Genta Indra Winata, Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018. Code-switching language modeling using syntax-aware multi-task learning. *arXiv preprint arXiv:1805.12070*.

A Experimental Setup

In the majority of our approaches, we perform task adaptive fine-tuning on BERT, mBERT , XLM-Roberta and character BERT for few epochs on an Nvidia GeForce GTX 1070 GPU. We primarily used Pytorch and Huggingface library for implementing different models. We experiment with

- batch sizes of 8,16, and 32
- learning rates between 1e-5 and 5e-5.

For some models, we observed variation in performance on test set based on subset of data used for training. To overcome this, we did 5 fold cross validation where there were no pre-defined train, dev and test data splits.

B Comparing pre-trained Models

Distillation seems to help compared to the corresponding full model. When finetuned for CS cases, distilled variants of BERT and mBERT performed significantly better than their pre-trained counterparts. We plan to investigate the reason behind the mixed results in future work.

C Benchmarking Arabic-English

We observed a similar trend in the benchmarking experiments for Arabic-English code-switching case as well. We performed NER using the dataset by (Molina et al., 2019). These results are shown in Table 4. We observe that a finetuned BERT model is already much better than the previous state-of-the-art model on the dataset. The M-BERT model further improves this score. However, distilled M-BERT did not show the same improvements as was shown on some other datasets. The trend with distilled models does not seem to be consistent (as discussed in Section B), and we believe that further investigation is needed to understand the reasons behind this performance. We do not include this in the results for benchmarking in Table 2. This is because we could not comprehensively compare the multitasking model with the rest of the models due to the lack of gold label annotations for this dataset (The remaining datasets compared in Table 2 were annotated with lexical level language ids as well). Finally, while char-BERT showed improvements both over the state of the art model and the finetuned BERT, it did not give the same improvement over the latter. We believe this needs further investigation as well.

Model	**Named Entity Recognition**
	msa-ea-N
Baseline	71.61
eng-BERT	74.13
M-BERT	79.73
Distill M-BERT	77.28
XLM-ROBERTa	77.68
char-BERT	74.46

Table 4: Performance of different multilingual models on MSA-EA (Molina et al., 2019) dataset.

D Self-Training Experiment Details

We show incremental model performance as we augment training data with batches of un-annotated data in Table 5. As we can observe from the table, the performance of the models increase and then decline after a point when further augmented. We believe the reason behind this is that we are overly biasing the model with this switch point beyond a certain level when the performance starts flipping towards decline. The optimal point of this iterative augmentation with self training is achieved before the flip in the overall performance.

Model	Biased Annotator	Sentences Added	Overall		X → en		en → X	
			Acc.	F1	Acc.	F1	Acc.	F1
POS-en-hi	-	-	89.84	85.21	87.91	86.11	82.62	80.23
POS-en-hi-selfTr	en → hi	+400	89.89	85.32	87.727	86.23	83.13	82.11
POS-en-hi-selfTr	en → hi	+400	89.91	85.16	87.27	86.34	84.38	85.01
POS-en-es	-	-	96.88	96.25	93.82	90.97	88.59	85.38
POS-en-es-selfTr	en → es	+150	97.01	96.29	94.15	91.02	89.66	87.01
POS-en-es-selfTr	en → es	+150	97.05	96.41	95.51	93.42	90.6	88.12
NER-en-hi	-	-	95.45	75.18	96.92	84.84	93.21	77.41
NER-en-hi-selfTr	en → hi	+100	95.71	77.44	96.92	83.11	93.66	77.96
NER-en-hi-selfTr	en → hi	+100	95.57	77.01	96.41	83.87	94.57	80.70
NER-en-hi-selfTr	en → hi	+100	95.41	75.02	95.89	78.78	95.02	80.70
NER-en-es	-	-	93.00	65.72	83.28	47.73	93.35	58.19
NER-en-es-selfTr	es → en	+500	93.32	65.84	83.89	50.62	93.98	60.29
NER-en-es-selfTr	es → en	+500	93.43	66.14	84.75	53.54	93.5	60.89
NER-en-es-selfTr	es → en	+500	93.12	66.34	86.13	56.62	94.58	62.29

Table 5: Results from Switch point biased self training. X refers to the language which is mixed with English. Iteratively # number of sentences are added to training set.