The Effectiveness of Intermediate-Task Training for Code-Switched Natural Language Understanding

Archiki Prasad[*†1] **Mohammad Ali Rehan**[*2] **Shreya Pathak**[*2] **Preethi Jyothi**[2]
[1]UNC Chapel Hill [2]IIT Bombay
archiki@cs.unc.edu
{alirehan, shreyapathak, pjyothi}@cse.iitb.ac.in

Abstract

While recent benchmarks have spurred a lot of new work on improving the generalization of pretrained multilingual language models on multilingual tasks, techniques to improve code-switched natural language understanding tasks have been far less explored. In this work, we propose the use of *bilingual intermediate pretraining* as a reliable technique to derive large and consistent performance gains using code-switched text on three different NLP tasks: Natural Language Inference (NLI), Question Answering (QA) and Sentiment Analysis (SA). We show consistent performance gains on four different code-switched language-pairs (Hindi-English, Spanish-English, Tamil-English and Malayalam-English) for SA and on Hindi-English for NLI and QA. We also present a code-switched masked language modeling (MLM) pretraining technique that consistently benefits SA compared to standard MLM pretraining using real code-switched text.

1 Introduction

Code-switching is a widely-occurring linguistic phenomenon in which multiple languages are used within the span of a single utterance or conversation. While large pretrained multilingual models like mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020) have been successfully used for low-resource languages and effective zero-shot cross-lingual transfer (Pires et al., 2019; Conneau et al., 2020; Wu and Dredze, 2019), techniques to help these models generalize to code-switched text have not been sufficiently explored.

Intermediate-task training (Phang et al., 2018, 2020) was recently proposed as an effective training strategy for transfer learning. This scheme involves fine-tuning a pretrained model on data from one or more *intermediate tasks*, followed by fine-tuning on the target task. The intermediate task could differ from the target task and it could also be in a different language. This technique was shown to help with both task-based and language-based transfer; it benefited target tasks in English (Vu et al., 2020) and helped improve zero-shot cross-lingual transfer (Phang et al., 2020).

In this work, we introduce *bilingual intermediate-task training* as a reliable training strategy to improve performance on three code-switched natural language understanding tasks: Natural Language Inference (NLI), factoid-based Question Answering (QA) and Sentiment Analysis (SA). Bilingual training for a language pair X-EN involves pretraining with an English intermediate task along with its translations in X. The NLI, QA and SA tasks require deeper linguistic reasoning (as opposed to sequence labeling tasks like part-of-speech tagging) and exhibit high potential for improvement via transfer learning. (The fact that NLI, QA and SA have more room for improvement compared to POS and NER tagging is evident from the leaderboard statistics in (Khanuja et al., 2020b).) We present SA results for four different language pairs: Hindi-English (HI-EN), Spanish-English (ES-EN), Tamil-English (TA-EN) and Malayalam-English (ML-EN), and NLI/QA results for HI-EN.[1]

Our main findings can be summarized as follows:

- Bilingual intermediate-task training consistently yields significant performance improvements on NLI, QA and SA using two different pretrained multilingual models, mBERT and XLM-R. We also show the impact of translation and transliteration quality on this training scheme.
- Pretraining using a masked language modeling (MLM) objective on real code-switched text can be used, in conjunction with bilingual

[*] Equal contribution
[†] Work done at IIT Bombay

[1]These tasks present an additional challenge with the Indian languages written using transliterated/Romanized text.

Proceedings of the 1st Workshop on Multilingual Representation Learning, pages 176–190
November 11, 2021.

training, for additional performance improvements on code-switched target tasks. We also present a code-switched MLM variant that yields larger improvements on SA compared to standard MLM.

2 Methodology

Intermediate-Task Training. This scheme starts with a publicly-available multilingual model that has been pretrained on large volumes of multilingual text using MLM-based training objectives. This model is subsequently fine-tuned using data from one or more intermediate tasks before finally fine-tuning on code-switched data from the target tasks.

Single Intermediate-Task Training makes use of existing monolingual NLI, SA and QA datasets as intermediate-tasks before fine-tuning on the respective code-switched target tasks. For a language pair X-EN, where X $\in$ {ES, HI, TA, ML}, we explored the use of three different intermediate tasks:

1. Task-specific data in English (EN SING-TASK): In this setting, we carry out intermediate training using a (relatively) larger English corpus of the same task as our final downstream task.

2. Task-specific data in X (X SING-TASK): Here, we carry out intermediate training using a corpus of the same task in the matrix language (i.e., not English) present in our code switched corpus. This corpus can be constructed by translating a monolingual English corpus into the target language, and then further transliterating it to be consistent with the Romanized forms present in the target tasks[2].

3. Task-specific data in both English and X that we refer to as *bilingual intermediate-task training* (X-EN SING-TASK): This intermediate-task pretraining method involves creating training batches with an equal number of examples from both languages. We conjecture that interleaving training instances from both languages within a batch encourages the model to simultaneously perform well on both languages, and could subsequently translate to improved performance on code-switched text in these specific language pairs. This claim is borne out in our experimental results detailed in Section 4. (We also show the importance of mixing instances from both languages rather than adopting a sequential training strategy on instances from both languages in Section 4.2.)

Multi Intermediate-Task Training involves two intermediate-tasks (T_1 and T_2) simultaneously. This training is done using two different task heads (one per task) with the pretrained models. Each batch is randomly populated with instances from tasks T_1 or T_2. We follow Raffel et al. (2020) to sample batches from task T_1 with probability $P_{T_1} = \frac{\min(e_{T_1}, K)}{\min(e_{T_1}, K) + \min(e_{T_2}, K)}$ where e_{T_1} and e_{T_2} are the number of training examples in task T_1 and T_2, respectively; P_{T_2} is similarly computed. The constant $K = 2^{16}$ is used to prevent over-sampling. We experiment with NLI and QA as the two intermediate-tasks T_1 or T_2 and refer to this system as HI-EN/NLI-QA MULTI-TASK. We use the merged EN and HI datasets from HI-EN SING-TASK for each task. We also explored MLM training on real code-switched text as one of the tasks, in addition to the merged X-EN task-specific intermediate-tasks (referred to as X-EN/MLM MULTI-TASK).

Code-Switched MLM. A common approach to training models for code-switched tasks is to perform additional MLM on real (or synthetic) code-switched text. However, randomly masking from the pool of all tokens in a sentence may not be the most effective use of real code-switched text and differentiating it from monolingual text, especially if one has access to word-level language tags. Given word-level language labels for each token in the code-switched sentences, we aim to emphasize switching via the MLM training objective by masking tokens from words that lie on the switching boundaries. We refer to this training strategy as *code-switched MLM*. For example, consider the following sentence where tokens that can be masked are enclosed within boxes for both the standard MLM and code-switched MLM strategies, respectively:

$\boxed{Yeh}_{HI}$ $\boxed{files}_{EN}$ $\boxed{ko}_{HI}$ $\boxed{desk}_{EN}$ $\boxed{pe}_{HI}$ $\boxed{rakh}_{HI}$ $\boxed{do}_{HI}$

$\boxed{Yeh}_{HI}$ $\boxed{files}_{EN}$ $\boxed{ko}_{HI}$ $\boxed{desk}_{EN}$ $\boxed{pe}_{HI}$ $rakh_{HI}$ do_{HI}

(EN Translation: Put these files on the desk.)

[2]Code-switched data for some Indic languages in our target corpora were only available in the Romanized form. Therefore, we only work with Romanized text in all our experiments including intermediate tasks.

In the first sentence, tokens from all the words can be masked, as in standard MLM pretraining. In the second sentence that uses code-switched MLM pretraining, only tokens from words at the boundary of a language switch can be masked. To implement this, we need access to annotated language tags for each sentence or a highly accurate language identity detection system. (Neither of these were available for Tamil or Malayalam datasets; hence our results for code-switched MLM are restricted to Hindi and Spanish.) An analysis of the MLM data showed that 45% of all tokens belonged to words on a switching boundary, therefore, the MLM masking probability of these tokens was increased from 0.15 to 0.3 to roughly balance the number of tokens that are masked on average.

3 Experimental Setup

3.1 Code-switched Target Datasets

The HI-EN NLI dataset is from a recent code-switched benchmark GLUECOS (Khanuja et al., 2020a) comprising 1.8K/447 training/test examples, respectively. The HI-EN factoid-based QA dataset (Chandu et al., 2018a) is also from GLUECOS, consisting of 259/54 training/test question-answer pairs (along with corresponding context), respectively. While code-switched NLI and QA tasks were only available in HI-EN, we show SA results for four language pairs. The ES-EN SA dataset (Vilares et al., 2016) in GLUECOS consists of 2.1K/211/211 examples in train/dev/test sets, respectively. The HI-EN SA dataset (Patwa et al., 2020) comprises 15K/1.5K/3K code-switched tweets in train/dev/test sets, respectively. The train/dev/test sets in the TA-EN SA dataset (Chakravarthi et al., 2020b) and ML-EN SA dataset (Chakravarthi et al., 2020a) comprise 9.6K/1K/2.7K and 3.9K/436/1.1K code-switched YouTube comments, respectively. As the evaluation metric, we use accuracies for NLI and SA over two (entailment/contradiction) and three labels (positive/negative/neutral), respectively, and F1 scores for the QA task.

3.2 Intermediate Task Datasets

As intermediate tasks for NLI and QA, we used EN and HI versions of the MultiNLI dataset (Williams et al., 2018) with 250/10K examples in the train/dev sets and the SQuAD dataset (Rajpurkar et al., 2016) consisting of 82K/5K question-answer pairs in its train/dev sets, respectively. The HI translations for SQuAD (in Devanagari) are available in the XTREME (Hu et al., 2020) benchmark. We used *indic-trans* (Bhat et al., 2014) to transliterate the HI translations, since NLI and QA in GLUECOS use Romanized HI text. For sentiment analysis in ES-EN and HI-EN, we used the TweetEval (Barbieri et al., 2020) dataset (63K sentences in total) and its translations in ES and HI generated via MarianMT[3] (Junczys-Dowmunt et al., 2018) and IndicTrans MT (Ramesh et al., 2021), respectively, for intermediate-task training. For TA-EN and ML-EN, we used the positive, negative and neutral labelled sentences from the SST dataset (Socher et al., 2013) (100K instances) as the intermediate task. The TA and ML translations were also generated using the IndicTrans MT system. The translations were further transliterated using Bhat et al. (2014) for HI and the Bing Translator API[4] for TA and ML.

3.3 Masked Language Modelling Datasets

We use a corpus of 64K real code-switched sentences by pooling together data from prior work (Singh et al., 2018; Swami et al., 2018; Chandu et al., 2018b); we will call this corpus GEN-CS. We supplant this text corpus with an additional 28K code-switched sentences mined from movie scripts (referred to as MOVIE-CS in Tarunesh et al. (2021b)), which is more similar in domain to GLUECOS NLI. We further used code-switched text from Patwa et al. (2020), Bhat et al. (2017), and Patro et al. (2017) resulting in a total of 185K HI-EN sentences. For ES-EN, 66K real code-switched sentences were accumulated from prior work (Patwa et al., 2020; Solorio et al., 2014; AlGhamdi et al., 2016; Aguilar et al., 2018; Vilares et al., 2016). For TA-EN and ML-EN (Chakravarthi et al., 2020b, 2021; Banerjee et al., 2018; Mandl et al., 2020; Chakravarthi et al., 2020a), we used roughly 130K and 40K real code-switched sentences, respectively.

3.4 XTREME Translation-Transliteration

As mentioned previously, for intermediate-task training, we use the MultiNLI and SQuAD v1.1 data from the translate-train sets of the XTREME benchmark[5,6]. The Romanized version of

[3]Implementation used: http://bit.ly/MarianMT

[4]http://bit.ly/azureTranslate

[5]MultiNLI available at: https://storage.cloud.google.com/xtreme_translations/XNLI/translate-train/en-hi-translated.tsv

[6]SQuAD available at: https://storage.cloud.google.com/xtreme_translations/SQuAD/

	Method	ES-EN (X: ES)			HI-EN (X: HI)			TA-EN (X: TA)			ML-EN (X: ML)		
		F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.
mBERT	Baseline	60.95	61.93	60.43	68.17	68.75	68	76.07	75.33	77.66	75.46	75.72	75.64
	+EN SING-TASK	65.11	66.00	65.00	69.14	69.72	68.96	76.41	75.69	78.11	76.49	76.78	76.44
	+X SINGLE-TASK	64.69	65.71	64.57	68.75	69.37	68.60	75.78	74.89	77.80	75.92	75.96	76.15
	+X-EN SINGLE-TASK	66.64	67.61	66.21	69.20	69.63	69.06	76.75	76.11	78.63	77.00	77.16	77.04
	+MLM	62.02	62.93	61.29	69.89	70.58	69.76	76.73	76.14	78.53	76.13	76.23	76.24
	+CODE-SWITCHED MLM	63.88	64.81	63.13	70.33	71.17	70.10	-	-	-	-	-	-
	+X-EN/MLM MULTI-TASK	67.01	68.11	66.72	69.99	70.29	69.91	77.23	76.6	**79.16**	77.49	77.56	**77.58**
XLM-R	Baseline	66.45	67.45	65.86	69.37	69.38	69.46	75.53	74.56	77.75	74.14	74.35	74.15
	+EN SING-TASK	67.82	68.89	67.41	70.23	70.78	70.09	76.08	75.41	77.65	75.14	75.29	75.42
	+X SINGLE-TASK	66.68	68.40	66.29	69.96	70.38	69.83	76.36	75.52	77.88	76.12	76.10	76.24
	+X-EN SINGLE-TASK	68.97	69.79	68.28	70.23	70.91	70.01	76.49	75.90	77.60	76.68	76.80	76.62
	+MLM	66.37	67.42	65.69	70.92	71.94	70.66	76.95	76.21	78.60	76.28	76.26	76.42
	+CODE-SWITCHED MLM	67.10	68.30	66.55	**71.74**	**72.29**	**71.59**	-	-	-	-	-	-
	+X-EN/MLM MULTI-TASK	**70.33**	**71.41**	**69.57**	71.08	71.43	70.97	**77.50**	**76.84**	78.60	**76.91**	**76.94**	76.98
	Our Best Models (Max)	71.7	72.8	71.3	72.6	73.2	72.4	78	77	79	78	78	78

Table 1: Our main results for sentiment analysis. Best results for each model are underlined and the overall best results are in bold. All scores are weighted averages and are further averaged over five runs with random seeds. The last row gives the F1, precision and recall value for the method with the maximum F1 across all five seeds.

these datasets are generated using the *indic-trans* tool (Bhat et al., 2014) starting from their Devanagari counterparts. For NLI, we directly transliterated the premise and hypothesis. For QA, the context, question and answer were transliterated and the answer span was corrected. This was done by calculating the start and stop indices of the span, followed by a piece-wise transliteration. We finally checked if the context-span matched the answer text. All instances passed this check. To benefit future work in this direction, we provide these transliterated datasets[7].

3.5 Model Details

mBERT is a transformer model (Vaswani et al., 2017) pretrained using MLM on the Wikipedia corpus of 104 languages. XLM-R uses a similar training objective as mBERT but is trained on orders of magnitude more data from the CommonCrawl corpus spanning 100 languages and yields competitive results on low-resource languages (Conneau et al., 2020). We use the *bert-base-multilingual-cased* and *xlm-roberta-base* models[8] from the Transformers library (Wolf et al., 2019). We refer readers to Appendix A and Appendix B for more implementation details.

translate-train/squad.translate.train.en-hi.json.

[7] https://www.cse.iitb.ac.in/~pjyothi/CS

[8] We also explored a multilingual model IndicBERT (Kakwani et al., 2020) trained exclusively on Indian languages. However, preliminary experiments using this model did not yield satisfactory performance, so we did not pursue it further. In future work, we will aim to use other recently released pretrained models such as MuRIL (Khanuja et al., 2021).

4 Results and Analysis

4.1 Results on Sentiment Analysis

Table 1 shows our main results for SA on ES-EN, HI-EN, TA-EN and ML-EN. We observe that bilingual intermediate-task training, X-EN SING-TASK, outperforms EN SING-TASK and X SING-TASK with both mBERT and XLM-R. The relative improvements of X-EN SING-TASK over the baseline vary across language pairs reaching up to 9.33% for ES. For all language pairs except HI-EN, X-EN/MLM MULTI-TASK is the best-performing system.[9] This demonstrates the benefits of MLM training in conjunction with intermediate-task training. A notable advantage of our bilingual training is that we outperform (or match) previous state-of-the-art with an order of magnitude less data. Our best ES-EN system yields an F1 of 71.7 compared to Prat-apa et al. (2018) with an F1 of 64.6. For HI-EN, our best F1 of 72.6 matches the 2nd-ranked system (Srinivasan, 2020) on SentiMix 2020 (Patwa et al., 2020). For TA-EN and ML-EN, our best systems match the score of the best TweetEval model in Gupta et al. (2021). While prior work required roughly 17M sentences in ES-EN, 2.09M sentences in HI-EN and 60M tweets to train TweetEval for TA and MA, we use 192K, 180K, 330K and 240K sentences for the four respective language pairs. While MLM training (i.e., +MLM in Table 1) consistently improves over the baseline, we observe that code-switched MLM (i.e., CODE-SWITCHED MLM

[9] We hypothesize the drop in performance for HI-EN could be attributed to domain differences between the SA and MLM corpora.

	Method	GLUECoS NLI *(acc.)*		GLUECoS QA *(F1)*	
		Max	**Mean**	**Max**	**Mean**
mBERT	Baseline	61.07	57.51	66.89	64.25
	+MLM	59.94	58.75	60.8	58.28
	+EN SING-TASK	62.40	60.73	77.62	75.77
	+HI SING-TASK	63.73	62.09	79.63	76.77
	+HI-EN SING-TASK	65.55	64.1	81.61	79.97
	+HI-EN/NLI-QA MULTI-TASK	**66.74**	65.3	83.03	80.25
	+HI-EN/MLM MULTI-TASK	66.66	**65.61**	81.05	79.11
XLM-R	Baseline	-	-	56.86	53.22
	+MLM	-	-	45.9	42.34
	+EN SING-TASK	66.22	63.91	82.04	80.92
	+HI SING-TASK	63.24	61.73	81.48	80.55
	+HI-EN SING-TASK	65.01	64.37	82.41	81.36
	+HI-EN/NLI-QA MULTI-TASK	64.49	64.35	**83.95**	**82.38**
	+HI-EN/MLM MULTI-TASK	66.66	65.01	82.1	80.44
Previous work on GLUECoS					
mBERT (Khanuja et al., 2020b)†		59.28	57.74	63.58	62.23
mod-mBERT (Chakravarthy et al., 2020)		62.41	-	-	-

Table 2: Our main results for NLI and QA from intermediate-task training. All scores are averaged over five runs with random seeds. Max and mean accuracies (for NLI) and F1-scores (for QA) over these runs are listed. Best results for each model are underlined and the overall best results are in bold. †Due to dataset changes, we cannot directly cite the results from the paper and report the numbers from the leaderboard after consulting the authors of GLUECoS.

in Table 1) provides additional performance gains for ES-EN and HI-EN. We do not report code-switched MLM results for TA-EN and ML-EN since we do not have access to language labels or a trained language identification system for either language. The ES-EN MLM dataset contains several sentences with no switching which are discarded for both standard and code-switched MLM. In Table 1, we compare +MLM and CODE-SWITCHED MLM only using sentences that contain code-switching.[10]

With access to translation and transliteration tools for a target language, we show superior results on four different language pairs for the sentiment analysis task. Even in resource-constrained settings like TA-EN and ML-EN, we obtain state-of-the-art performance using our proposed techniques. In Section 4.3, we will examine the influence of translation and transliteration quality on performance.

4.2 Results on NLI and QA

NLI/QA SINGLE TASK Results. Table 2 shows our main results for the NLI and QA tasks in HI-EN. (Code-switched benchmarks in other language pairs are not available for NLI and QA.) Among the SINGLE-TASK systems, HI-EN SING-TASK performs the best (based on mean scores) on both NLI[11] and QA. Another interesting observation is that XLM-R benefits more from EN SING-TASK while mBERT benefits more from HI SING-TASK, compared to the baseline. This could be attributed to XLM-R having encountered Romanized HI text during its pretraining unlike mBERT and the GLUECoS corpus contains only Romanized Hindi.

Using a merged HI-EN dataset for HI-EN SING-TASK training, with training batches consisting of both HI and EN instances, was critical for improved performance. Table 3 shows the difference in per-

Intermediate-Task Paradigm	Max	Mean	Std.
	GLUECoS NLI *(acc.)*		
EN SING-TASK	62.40	60.73	1.78
HI SING-TASK	63.73	62.09	0.99
HI-EN SING-TASK	65.55	64.1	0.89
Sequential Training: EN → HI	62.02	59.94	1.83
	GLUECoS QA *(F1)*		
EN SING-TASK	77.62	75.77	1.79
HI SING-TASK	79.63	76.77	1.86
HI-EN SING-TASK	81.61	79.97	1.29
Sequential Training: EN → HI	76.23	73.69	1.78

Table 3: Sequential bilingual training with mBERT yields poor performance on both NLI and QA. Scores correspond to five random runs with random seeds.

[10] On using the complete En-Es MLM corpus for +MLM, we obtained an F1 of 62.57 using mBERT and 67.6 using XLM-R on the SA test set of ES-EN.

[11] Like Chakravarthy et al. (2020), we also find that XLM-R baseline/+MLM on GLUECoS NLI does not converge and hence we do not report these scores in Table 2.

Translate − Transliterate	Max	Mean	Std.
	GLUECOS NLI *(acc.)*		
Manual − Google Translate API	62.24	61.6	0.62
Manual − *indic-trans*	62.09	59.71	1.37
Google Translate API (both)	60.18	58.59	1.07
	GLUECOS QA *(F1)*		
Manual − Google Translate API	79.32	77.33	2.22
Manual − *indic-trans*	78.09	76.35	1.36
Google Translate API (both)	78.44	76.72	1.22

Table 4: Effect of translation and transliteration quality on intermediate-task training, using HI-EN SING-TASK for NLI and QA. Scores correspond to five random runs with random seeds.

Translate − Transliterate	Max	Mean	Std.
	GLUECOS NLI *(acc.)*		
Manual − Google Translate API	61.05	59.63	0.96
Manual − *indic-trans*	59.50	59.26	0.23
Google Translate API (both)	60.12	58.54	1.53
	GLUECOS QA *(F1)*		
Manual − Google Translate API	73.50	71.36	1.46
Manual − *indic-trans*	70.19	68.26	1.26
Google Translate API (both)	72.73	69.63	2.2

Table 5: Effect of translation and transliteration quality on intermediate-task training, using HI SING-TASK for NLI and QA. Scores correspond to five random runs with random seeds.

formance between sequentially training on English followed by Hindi versus mixing instances from both languages as in HI-EN SING-TASK. We observe a clear deterioration in performance with sequential training, with the latter performing even worse than its monolingual counterparts (EN SING-TASK and HI SING-TASK). This confirms that bilingual training is essential to improved performance on code-switched tasks.

NLI/QA MULTI TASK Results. Table 2 shows that the MULTI-TASK systems yield additional gains over the SING-TASK systems. Using both NLI and QA as intermediate tasks benefits both NLI and QA for mBERT and QA for XLM-R, and corroborates observations in prior work (Tarunesh et al., 2021a; Phang et al., 2020). Although intermediate-task training is beneficial across tasks, the relative improvements in QA are higher than that for NLI (see Appendix C for some QA examples). We conjecture this is due to varying dataset similarity between intermediate-tasks and target tasks (Vu et al., 2020). In QA, this similarity is higher and in NLI the conversational nature and large premise lengths reduces this similarity. The effect of domain similarity is more pronounced with MLM training resulting in variations between absolute 1.5-2%. More experiments detailing when MLM training benefits the downstream tasks is described in Section 4.4.

4.3 Influence of Translation and Transliteration Quality

Transliteration and translation are the two key preprocessing steps employed for bilingual pretraining. Since we make use of existing translation and transliteration tools that are not error-free, it is useful to understand the impact of such translation and transliteration tools on final downstream task performance.

To assess the impact of both translation and transliteration quality on NLI and QA performance, we use two small datasets XNLI (Conneau et al., 2018) and XQuAD (Artetxe et al., 2020) for which we have manual HI (Devanagari) translations. We combined the test and dev sets of XNLI to get the data for intermediate-task training. We discarded all examples labelled *neutral* and instances where the crowdsourced annotations did not match the designated labels[12]. After this, we were left with roughly 4.2K/0.5K instances in the train/dev sets, respectively (the dev set is used for early stopping during intermediate-task training). For XNLI, the premises and hypotheses were directly translated and for XQuAD we adopted the same translation procedure listed in Hu et al. (2020).

In Table 4, we compare the performance of HI-EN SING-TASK using manual translations with translations from the Google Translate API[13], and also transliterations from this API with those from *indic-trans*. As expected, using manual translations is most beneficial to the downstream task. The use of Google Translate, however, does not significantly hamper performance. Similar to the results in Table 4 for bilingual intermediate-task training, we present a similar analysis in Table 5 when using task-specific data in HI SING-TASK with mBERT and observe the same trends. Keeping the translation method fixed as manual, we tried using *indic-trans* for transliteration instead of the Google API. We see this led to a decrease in performance in all the 4 cases (i.e., across two models and two tasks in Tables 4 and 5), thus indicating transliterations from the Google translate API would be a better choice as compared to *indic-trans*.[14]

[12]This was achieved via the *match* Boolean attribute (Conneau et al., 2018)

[13]https://cloud.google.com/translate

[14]We did not switch to Google Translate for all our main experiments due to the overhead of obtaining Google Translate-

Task	Model	Translit Tool	F1	Prec.	Rec.
TA SINGLE-TASK	mBERT	*indic-trans*	75.42	74.72	76.62
		Bing API	75.78	74.89	77.8
	XLM-R	*indic-trans*	75.51	74.87	76.66
		Bing API	76.36	75.52	77.88
ML SINGLE-TASK	mBERT	*indic-trans*	74.7	74.82	74.71
		Bing API	75.92	75.96	76.15
	XLM-R	*indic-trans*	74.68	74.82	74.66
		Bing API	76.12	76.1	76.24

Table 6: Effect of transliteration quality of intermediate-tasks on SA results. Scores are weighted averages further averaged over 5 random runs.

Original Script	Transliterated using indic-trans	Transliterated via Bing Translator API
ഓർമ്മിക്കപ്പെടും (memorable)	oർmmicppet	ormikapedum
സ്പർശിക്കുന്നു (touching)	spർshikunnu	sparsikunnu
തളർന്നുപോയി. (tired)	talർnnupoyi.	thalarnupoyi
ആകർഷണീയം (appealing)	oakർshaniyam	akarshaniyam

Figure 1: Different transliterations for some descriptive words in MA. *indic-trans* leaves some residual characters in the native script.

Table 6 shows the impact of transliteration on sentiment analysis of TA-EN and ML-EN. Again, we see that using an improved transliteration tool led to improved performance across both Tamil and Malayalam.

Figure 1 illustrates different MA transliterations. From the figure, we notice that *indic-trans* tends to retain some Malayalam characters in its native script (possibly due to incomplete Unicode support) and also does not produce very accurate transliterations. Transliterations from the Bing API are more phonetically accurate. Table 7 shows an example from the HI-EN NLI dataset, that is translated and transliterated using Google Translate and indic-trans. The color-coded transliterations indicate that *indic-trans* often uses existing English words as transliterations. While this is helpful for some specific (uncommon) words, in most cases it leads to ambiguity in sentence meaning (shown in blue). Further, these ambiguous words are far more common in the HI language, and thus have a greater impact on model performance.

In summary, developing more accurate tools for translation and transliteration would be very beneficial for downstream code-switched tasks.

based transliterations for the large intermediate task datasets.

4.4 MLM and Intermediate-Task Training

How does MLM pretraining in conjunction with intermediate-task training impact performance? What is the influence of changing the MLM corpus (and hence its domain) on final task performance? We address these questions in this section by focusing on NLI and QA for HI-EN using mBERT.

Table 8 provides a summary of our experiments on intermediate-task training of mBERT using only English (EN) and both English and Hindi (HI-EN) in conjunction with MLM in the MULTI-TASK setting described in Section 2.

From Table 8, we observe that intermediate training using MLM on code-switched data alone (i.e., the first row for each task) is not as effective as using both MLM and intermediate-task pretraining. NLI benefits from MLM in a multi-task setup in both monolingual and bilingual settings. Further, we note that adding in-domain MOVIE-CS data yields additional improvements for NLI. This shows that sufficient amount of in-domain data is needed for performance gains, and augmenting out-of-domain with in-domain code-switched text can be effective.

In the case of QA, MLM does not improve performance in the monolingual setting, although the mean scores are statistically close. In the bilingual setting, we see a clear improvement using GEN-CS for MLM training. However, using both GEN-CS and MOVIE-CS for MLM results in significant degradation of performance. We believe that this is due to the domain of the passages in GLUECOS QA being similar to the HI-EN blog data present in GEN-CS. However, the MOVIE-CS dataset comes from a significantly different domain and thus hurts performance. This indicates that in addition to the amount of unlabelled real code-switched text, when using MLM training, the domain of the text is very influential in determining the performance on downstream tasks (Gururangan et al., 2020).

For both NLI and QA, we observe the following common trend: Adding code-switched data from the training set of GLUECOS tasks (referred to as GLUECOS NLI CS and GLUECOS QA CS) degrades performance. This could be due to the quality of training data in the GLUECOS tasks. Each dialogue in the NLI data does not have a lot of content and is highly conversational in nature. In addition to this, the dataset is also very noisy. For example, a word *'humko'* is split into its characters *'h u m k o'*. Thus,

Language	Premise/ Hypothesis	Label	Dataset
EN	**PREMISE:** Split Ends a Cosmetology Shop is a nice example of appositional elegance combined with euphemism in the appositive and the low key or off-beat opening. **HYPOTHESIS:** Split Ends is an ice cream shop.	entailment	MultiNLI/ XNLI
HI (Google$^{\diamond}$)	**PREMISE:** split ends ek *kosmetolojee* shop epositiv *aur* kam kunjee ya oph-beet *opaning* mein vyanjana ke saath sanyukt eplaid laality ka ek achchha udaaharan hai. **HYPOTHESIS:** split *ends* ek *aaisakreem* shop *hai*.	entailment	Translation†
HI (Google$^{\diamond}$)	**PREMISE:** split inds ek *kosmetolojee* shop samaanaadhikaran shishtata *aur* kam kunjee ya of-beet *opaning* mein preyokti ke mishran ka ek achchha udaaharan *hai*. **HYPOTHESIS:** split *ends* ek *aaisakreem* kee dukaan *hai*.	entailment	XNLI
HI (indic*)	**PREMISE:** split inds ek *cosmetology* shop samaanaadhikaran shishtataa *or* kam kunjee yaa of-beet opening main preyokti ke mishran kaa ek acha udhaaharan *he*. **HYPOTHESIS:** split *ands* ek *icecream* kii dukaan *he*.	entailment	XNLI

Table 7: NLI examples from some of our datasets. †: obtained by translation of the second row using Google Translate API. $^{\diamond}$: transliterated using Google Translate API, *: transliterated using *indic-trans* (Bhat et al., 2014). In *blue*, we show some of the words with ambiguous transliterations by *indic-trans*. In *purple*, we show some words that are better transliterated by *indic-trans*. Best viewed in color.

Language	MLM Data	Max	Mean	Std.
		GLUECOS NLI *(acc.)*		
-	GEN-CS	59.94	58.75	0.93
EN	-	62.40	60.73	1.78
	GEN-CS	65.07	62.84	1.93
HI-EN	-	65.55	64.1	0.89
	GEN-CS	65.22	64.19	1.22
	GENERAL + MOVIE CS	**66.67**	**65.61**	0.86
	GENERAL + MOVIE CS + GLUECOS NLI CS	66.17	65.21	0.96
		GLUECOS QA *(F1)*		
-	GEN-CS	59.26	57.84	1.29
EN	-	77.62	75.77	1.79
	GEN-CS	76.23	75.49	1.03
HI-EN	-	81.61	79.97	1.29
	GEN-CS	**83.03**	**80.38**	1.68
	GENERAL + MOVIE CS	81.05	79.11	1.40
	GENERAL + MOVIE CS + GLUECOS QA CS	79.63	78.27	1.46

Table 8: Performance on different variations of MLM + intermediate-task training of mBERT. We underline the relatively best model and bold-face the model with the highest performance for each task.

MLM on such data may not be very effective and could hurt performance. For QA, passages in significant portions of the train set are obtained using *DrQA - Document Retriever module*[15] (Chen et al., 2017). These passages are monolingual in nature and thus potentially not useful for MLM training with code-switched text.

5 Related Work

While pretrained multilingual models are being increasingly used for cross-lingual natural language understanding tasks, their effectiveness for code-switched tasks has not been thoroughly explored. Winata et al. (2021) show that embeddings from pretrained multilingual models are not very effective for code-switched tasks and more work is needed to effectively adapt them.

Intermediate task-training has proven to be effective for many NLP target tasks (Pruksachatkun et al., 2020; Vu et al., 2020), as well as cross-lingual zero-shot transfer from English tasks on multilingual models such as XLM-R (Phang et al., 2020) and mBERT (Tarunesh et al., 2021a). Ours is the first work to show improved intermediate task-training strategies for code-switched target tasks.

Pires et al. (2019) and Hsu et al. (2019) showed that mBERT is effective for HI-EN part-of-speech tagging and a reading comprehension task on synthetic code-switched data, respectively. This was extended for a variety of code-switched tasks by Khanuja et al. (2020b), where they showed improvements on several tasks using MLM pretraining on real and synthetic code-switched text. Chakravarthy et al. (2020) further improved the NLI performance of mBERT by including large amounts of in-domain code-switched text during MLM pretraining.

Gururangan et al. (2020) empirically demonstrate that pretraining is most beneficial when the domains of the intermediate and target tasks are similar, which we observe as well. Differing from their recommendation of domain adaptive pretraining using MLM on large quantities of real code-switched data, we find intermediate-task training using significantly smaller amounts of labeled data to be more consistently beneficial across tasks and languages. In contrast to very recent work (Gupta et al., 2021) that reports results using a Roberta-based model trained exclusively for sentiment analysis and pretrained on 60M English tweets, we present a bilingual training technique that is consistently effective across tasks and languages while

[15] https://github.com/facebookresearch/DrQA

requiring significantly smaller amounts of data.
Instead of using mBERT and XLM-R that are very broad in their coverage of languages, it would be interesting to examine whether our observed trends hold when using pretrained models specifically trained for the chosen target languages. We could consider using very recent models like IndicBERT (Kakwani et al., 2020) and MuRIL (Khanuja et al., 2021) that are trained exclusively on Indian languages and have been shown to outperform mBERT on cross-lingual tasks (e.g., XTREME) and tasks like IndicGLUE, respectively. We leave this investigation for future work.

6 Conclusion

This is the first work to demonstrate the effectiveness of intermediate-task training for code-switched NLI, QA and SA on different language pairs, and present code-switched MLM that consistently benefits SA more than standard MLM. We also carry out ablations of transliteration systems and compare their performance across the same corpora translated using different techniques. We observe that high-quality translations and transliterations are important to derive performance improvements on downstream tasks.
For future work, we plan to continue exploring pretraining strategies, based on more informed masking objectives and task-adaptive techniques. One key limitation of the newly introduced code-switched MLM approach is the requirement of LID systems for the languages under consideration. Future work can focus on mitigating this requirement.

7 Acknowledgements

The authors would like to thank the anonymous reviewers for their useful and constructive comments that helped improve the draft.

References

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.

Fahad AlGhamdi, Giovanni Molina, Mona Diab, Thamar Solorio, Abdelati Hawwari, Victor Soto, and Julia Hirschberg. 2016. Part of speech tagging for code switched data. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 98–107, Austin, Texas. Association for Computational Linguistics.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

Suman Banerjee, Nikita Moghe, Siddhartha Arora, and Mitesh M. Khapra. 2018. A dataset for building code-mixed goal oriented conversation systems. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3766–3780, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650, Online. Association for Computational Linguistics.

Irshad Bhat, Riyaz A. Bhat, Manish Shrivastava, and Dipti Sharma. 2017. Joining hands: Exploiting monolingual treebanks for parsing of code-mixing data. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 324–330, Valencia, Spain. Association for Computational Linguistics.

Irshad Ahmad Bhat, Vandan Mujadia, Aniruddha Tammewar, Riyaz Ahmad Bhat, and Manish Shrivastava. 2014. IIIT-H System Submission for FIRE2014 Shared Task on Transliterated Search. In *Proceedings of the Forum for Information Retrieval Evaluation*, pages 48–53. ACM.

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, Navya Jose, Anand Kumar M, Thomas Mandl, Prasanna Kumar Kumaresan, Rahul Ponnusamy, Hariharan R L, John P. McCrae, and Elizabeth Sherly. 2021. Findings of the shared task on offensive language identification in Tamil, Malayalam, and Kannada. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 133–145, Kyiv. Association for Computational Linguistics.

Sharanya Chakravarthy, Anjana Umapathy, and Alan W Black. 2020. Detecting entailment in code-mixed Hindi-English conversations. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 165–170, Online. Association for Computational Linguistics.

Khyathi Chandu, Ekaterina Loginova, Vishal Gupta, Josef van Genabith, Günter Neumann, Manoj Chinnakotla, Eric Nyberg, and Alan W. Black. 2018a. Code-mixed question answering challenge: Crowd-sourcing data and techniques. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, pages 29–38, Melbourne, Australia. Association for Computational Linguistics.

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

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. 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.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of*

the 2018 Conference on Empirical Methods in Natural Language Processing, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

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

Akshat Gupta, Sai Krishna Rallabandi, and Alan W Black. 2021. Task-specific pre-training and cross lingual transfer for sentiment analysis in Dravidian code-switched languages. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 73–79, Kyiv. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Tsung-Yuan Hsu, Chi-Liang Liu, and Hung-yi Lee. 2019. Zero-shot reading comprehension by cross-lingual transfer learning with multi-lingual language representation model. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5933–5940, Hong Kong, China. Association for Computational Linguistics.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *Proceedings of the 37th International Conference on Machine Learning*, pages 4411–4421, Virtual. PMLR.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

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 P. Talukdar. 2021. Muril: Multilingual representations for indian languages. *CoRR*, abs/2103.10730.

Simran Khanuja, Sandipan Dandapat, Sunayana Sitaram, and Monojit Choudhury. 2020a. A new dataset for natural language inference from code-mixed conversations. In *Proceedings of the The 4th Workshop on Computational Approaches to Code Switching*, pages 9–16, Marseille, France. European Language Resources Association.

Simran Khanuja, Sandipan Dandapat, Anirudh Srinivasan, Sunayana Sitaram, and Monojit Choudhury. 2020b. 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.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Thomas Mandl, Sandip Modha, Anand Kumar M, and Bharathi Raja Chakravarthi. 2020. Overview of the hasoc track at fire 2020: Hate speech and offensive language identification in tamil, malayalam, hindi, english and german. In *Forum for Information Retrieval Evaluation*, page 29–32.

Jasabanta Patro, Bidisha Samanta, Saurabh Singh, Abhipsa Basu, Prithwish Mukherjee, Monojit Choudhury, and Animesh Mukherjee. 2017. All that is English may be Hindi: Enhancing language identification through automatic ranking of the likeliness of word borrowing in social media. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2264–2274, Copenhagen, Denmark. Association for Computational Linguistics.

Parth Patwa, Gustavo Aguilar, Sudipta Kar, Suraj Pandey, Srinivas PYKL, Björn Gambäck, Tanmoy Chakraborty, Thamar Solorio, and Amitava Das. 2020. SemEval-2020 task 9: Overview of sentiment analysis of code-mixed tweets. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 774–790, Barcelona (online). International Committee for Computational Linguistics.

Jason Phang, Iacer Calixto, Phu Mon Htut, Yada Pruksachatkun, Haokun Liu, Clara Vania, Katharina Kann, and Samuel R. Bowman. 2020. English intermediate-task training improves zero-shot cross-lingual transfer too. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 557–575, Suzhou, China. Association for Computational Linguistics.

Jason Phang, Thibault Févry, and Samuel R Bowman. 2018. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *arXiv preprint arXiv:1811.01088*.

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.

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, Brussels, Belgium. Association for Computational Linguistics.

Yada Pruksachatkun, Jason Phang, Haokun Liu, Phu Mon Htut, Xiaoyi Zhang, Richard Yuanzhe Pang, Clara Vania, Katharina Kann, and Samuel R. Bowman. 2020. Intermediate-task transfer learning with pretrained language models: When and why does it work? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5231–5247, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Gowtham Ramesh, Sumanth Doddapaneni, Aravinth Bheemaraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Mahalakshmi J, Divyanshu Kakwani, Navneet Kumar, Aswin Pradeep, Kumar Deepak, Vivek Raghavan, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh Shantadevi Khapra. 2021. Samanantar: The largest publicly available parallel corpora collection for 11 indic languages.

Kushagra Singh, Indira Sen, and Ponnurangam Kumaraguru. 2018. 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, Melbourne, Australia. Association for Computational Linguistics.

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, Elizabeth Blair, Suraj Maharjan, Steven Bethard, Mona Diab, Mahmoud Ghoneim, Abdelati Hawwari, Fahad AlGhamdi, Julia Hirschberg, Alison Chang, and Pascale Fung. 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, Doha, Qatar. Association for Computational Linguistics.

Anirudh Srinivasan. 2020. MSR India at SemEval-2020 task 9: Multilingual models can do code-mixing too. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 951–956, Barcelona (online). International Committee for Computational Linguistics.

Sahil Swami, Ankush Khandelwal, Vinay Singh, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. A corpus of english-hindi code-mixed tweets for sarcasm detection. *arXiv preprint arXiv:1805.11869*.

Ishan Tarunesh, Sushil Khyalia, Vishwajeet Kumar, Ganesh Ramakrishnan, and Preethi Jyothi. 2021a. Meta-learning for effective multi-task and multilingual modelling. *arXiv preprint arXiv:2101.10368*.

Ishan Tarunesh, Syamantak Kumar, and Preethi Jyothi. 2021b. From machine translation to code-switching: Generating high-quality code-switched text. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

David Vilares, Miguel A. Alonso, and Carlos Gómez-Rodríguez. 2016. EN-ES-CS: An English-Spanish code-switching Twitter corpus for multilingual sentiment analysis. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4149–4153, Portorož, Slovenia. European Language Resources Association (ELRA).

Tu Vu, Tong Wang, Tsendsuren Munkhdalai, Alessandro Sordoni, Adam Trischler, Andrew Mattarella-Micke, Subhransu Maji, and Mohit Iyyer. 2020. Exploring and predicting transferability across NLP tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7882–7926, Online. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*,

pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Genta Indra Winata, Samuel Cahyawijaya, Zihan Liu, Zhaojiang Lin, Andrea Madotto, and Pascale Fung. 2021. Are multilingual models effective in code-switching? In *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, pages 142–153, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, pages arXiv–1910.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 833–844, Hong Kong, China. Association for Computational Linguistics.

Appendix

A Implementation Details

The mBERT model comprises 179M parameters with the MLM head comprising 712K parameters. The XLM-R model comprises 270M parameters with an MLM head with 842k parameters. For both models, the NLI (sequence classification) and QA heads comprise 1536 parameters each. For SA (sequence classification) the head comprises of 2304 parameters.

B Hyperparameter Tuning

In all experiments, we have used the AdamW algorithm (Loshchilov and Hutter, 2019) and a linear scheduler with warm up for the learning rate. These experiments were run on a single NVIDIA GeForce GTX 1080 Ti GPU. Some crucial fixed hyperparameters are: `learning_rate = 5e-5`, `adam_epsilon = 1e-8`, `max_gradient_norm = 1`, and `gradient_accumulation_steps = 10`.

B.1 Intermediate-Task Training

The training for all the main intermediate-task experiments was carried out for 4 epochs and the model with the highest performance metric on the task dev set was considered (all the metrics stagnated after a certain point in training). For NLI + QA tasks, two separate models were stored depending on the performance metric on the respective dev set. No hyperparameter search was conducted at this stage. During bilingual training, the batches were interspersed—equal number of examples from English and Romanized HI within each batch. In the single-task systems, we used `batch_size = 8` and `max_sequence_length = 128` for NLI, `batch_size = 8` and `max_sequence_length = 256` for SA, `batch_size = 4` and `max_sequence_length = 512` for QA. During multi-task training, the `max_sequence_length` was set to the maximum of the aforementioned numbers and the respective batch-sizes. Any multi-task training technique requires at least 14-15 hours for validation accuracy to stagnate. Single task intermediate training requires 4-5 hours for monolingual versions and 8-9 hours for the bilingual version. SA data being smaller in size requires 8-9 hours for multitask, 4-5 hours for bilingual intermediate task and 1-2 hours for monolingual intermediate task. The `logging_steps` are set to approximately 10% of the total steps in an epoch.

B.2 Fine-tuning on GLUECoS NLI & QA Tasks

The base fine-tuning files have been taken from the GLUECoS repository[16]. Given that there no dev sets in GLUECoS, and that the tasks are low-resource, we use train accuracy in NLI and train loss in QA as an indication to stop fine-tuning. Manual search is performed over a range of epochs to obtain the best test performance. For NLI, we stopped fine-tuning when training accuracy is in the range of 70-80% (which meant fine-tuning for 1-4 epochs depending upon the model and technique used). For QA, we stopped when training loss reached $\sim$ 0.1. Thus, we explored 3-5 epochs for mBERT and 4-8 epochs for XLM-R. We present the statistics over the best results on 5 different seeds. We used `batch_size = 8` and `max_sequence_length = 256` for GLUECoS NLI[17] and `batch_size = 4` and `max_sequence_length = 512` for GLUECoS QA. All our fine-tuning runs on GLUECoS take an average of 1 minute per epoch.

B.3 Fine-tuning on downstream SA tasks

The dev set, being available for all language pairs was used to find the checkpoint with best F1 score, and this model was used for evaluation on the test set. The mean values were presented after carrying out the above procedure for 6 different seeds. The `logging_steps` are set to approximately 10% of the total steps in an epoch. Each epoch takes around 1 minute for TA, MA and ES, 2 minutes for HI (SemEval).

C Example Outputs

In Table 9, we show some instances from the HI-EN QA dataset. The color-coded transliterations indicate that *indic-trans* often uses existing English words as transliterations. While for some specific (uncommon) words that is helpful, in most cases it leads to ambiguity in the sentence meaning (shown in blue). Further, these ambiguous words (in blue) are far more common in the HI language, and thus, have a greater impact on model performance. We also note that transliterations of these common words in the GLUECoS dataset matches closely with the transliterations produced using the

[16] https://github.com/microsoft/GLUECoS

[17] The sequence length was doubled as compared to the intermediate-task training to incorporate the long premise length of GLUECoS NLI. This resulted in higher accuracy.

Language	QA Context	Dataset
HI-EN	Mitashi ne ek Android Tv ko Launch kiya hain. Jise tahat yeh Tv Android Operating System par chalta hain. Iski Keemat Rs. 51,990 rakhi gayi hain. Ab aaya Android TV Mitashi Company ne Android KitKat OS par chalne wale Smart TV ko Launch kar diya hain. Company ne is T.V. ko 51,990 Rupees ke price par launch kiya hain. Agar features ki baat kare to is Android TV ki Screen 50 inch ki hain, Jo 1280 X 1080 p ka resolution deti hain. USB plug play function ke saath yeh T.V. 27 Vidoe formats ko support karta hain. Vidoe input ke liye HDMI Cable, PC, Wi-Fi aur Ethernet Connectivity di gyi hain. Behtar processing ke liye dual core processor ke saath 512 MB ki RAM lagayi gyi hain. Yeh Android TV banane wali company Mitashi isse pahle khilaune banane ka kaam karti thi. Iske alawa is company ne education se jude products banane shuru kiye. 1998 mein stapith huyi is company ne Android T.V. ke saath-saath India ki pahli Android Gaming Device ko bhi launch kiya hain.	GLUECOS QA
EN	Their local rivals, Polonia Warsaw, have significantly fewer supporters, yet they managed to win Ekstraklasa Champi-onship in 2000. They also won the country's championship in 1946, and won the cup twice as well. Polonia's home venue is located at Konwiktorska Street, a ten-minute walk north from the Old Town. Polonia was relegated from the country's top flight in 2013 because of their disastrous financial situation. They are now playing in the 4th league (5th tier in Poland) -the bottom professional league in the National – Polish Football Association structure.	SQuAD/XQuAD
HI (Google$^{\diamond}$)	unake sthaaneey pratidvandviyon, *poloniya voraso* ke paas kaaphee kam samarthak hain, phir bhee ve 2000 *mein* ekastraklaasa *chaimpiyanaship* jeetane *mein* kaamayaab rahe. unhonne 1946 *mein* desh kee *chaimpiyanaship* bhee jeetee, *aur* do baar *kap* bhee jeeta. *poloniya* ka ghareloo sthal konaveektarsaka street par sthit *hai*, jo old taun se uttar *mein* das *minat* kee paidal dooree par *hai*. apanee vinaashakaaree vitteey sthiti ke kaaran *poloniya* ko 2013 *mein* desh kee sheersh udaan se hata diya gaya tha. ab ve neshanal (polish polish esosieshan) sanrachana *mein* 4 ven leeg (polaind *mein* 5 ven star) *mein* khel rahe hain.	Translation†
HI (Google$^{\diamond}$)	unake sthaaneey pratidvandviyon, *poloniya vaaraso*, ke paas kaaphee kam samarthak hain, phir bhee ve 2000 *mein* ekalastralaasa *chaimpiyanaship* jeetane *mein* kaamayaab rahe. unhonne 1946 *mein* raashtriy *chaimpiyanaship* bhee jeetee, *aur* saath hee do baar *kap* jeete. *poloniya* ka ghar konaveektarsaka street par sthit *hai*, jo old taun se uttar *mein* das *minat* kee paidal dooree par *hai*. *poloniya* ko 2013 *mein* unakee kharaab vitteey sthiti kee vajah se desh kee sheersh udaan se hata diya gaya tha. ve ab botam profeshanal leeg ke 4th leeg (polaind *mein* 5 ven star) neshanal polish futabol esosieshan sanrachana *mein* khel rahe hain.	XQuAD
HI (indic*)	unke sthaneey pratidwandviyon, *polonia warsaw*, ke paas kaaphi kam samarthak hai, phir bhi ve 2000 *main* ecrestlasa *championships* jeetne *main* kaamyaab rahe. unhone 1946 *main* rashtri *championships* bhi jiti, *or* saath hi do baar *cap* jite. *polonia* kaa ghar konwictarska street par sthit *he*, jo old toun se uttar *main* das *minute* kii paidal duuri par *he*. *polonia* ko 2013 *main* unki karaab vittiya sthiti kii vajah se desh kii sheersh udaan se hataa diya gaya tha. ve ab bottm profeshnal lig ke 4th lig (poland *main* 5 wein str) neshnal polish footbaal association sanrachana *main* khel rahe hai.	XQuAD

Table 9: QA examples from some of our datasets. †: obtained by translation of the second row using Google Translate API. $^{\diamond}$: transliterated using Google Translate API, *: transliterated using *indic-trans* (Bhat et al., 2014). In *blue*, we show some of the words words with ambiguous transliteration by *indic-trans* and their counterparts. In *purple*, we show some words that are better transliterated by *indic-trans*. Best viewed in color.

Language	Sentence	Label	Dataset
EN	It's definitely Christmas season! My social media news feeds have been all about Hatchimals since midnight! Good luck parents!	positive	TweetEval
HI$^{\diamond}$	yeah nishchit roop se christmas ka mausam hai! mera social media news feed adhi raat se hatchimal ke baare mein hai! mata-pita ko shubhkamnayen!	positive	Translation†
ES	¡Es definitivamente la temporada de Navidad! Mis noticias en las redes sociales han sido todo acerca de Hatchimals desde medianoche! ¡Buena suerte padres!	positive	Translation‡
ML$^{\diamond}$	ith theerchayayum chrismas seesonnan, ente social media news feads ardharathri muthal hachimalsine kurichan!	positive	Translation†
TA$^{\diamond}$	itu nichchayam christumus column! nalliravu muthal enathu samook utaka seithi oottngal anaithum hatchimals patriadhu! petrors nalvazthukal!	positive	Translation†
EN	the story and the friendship proceeds in such a way that you 're watching a soap opera rather than a chronicle of the ups and downs that accompany lifelong friendships .	negative	SST
HI$^{\diamond}$	kahani or dosti is tarah se aage badhati hai ki op jeevan bhar ki dosti ke saath aane vale utaar-chadhav k kram k bajay ek dharavahik dekh rahe hain	negative	Translation†
ES	la historia y la amistad proceden de tal manera que estás viendo una telenovela en lugar de una crónica de los altibajos que acompañan a las amistades de toda la vida.	negative	Translation‡
ML$^{\diamond}$	ajeevanantha sauhradangalil undakunna uyarchayeyum thazhchayeyum kurichulla oru kathayalla, marich oru sopp opera kanunna reethiyilan kathayum souhrdavum munnot pokunnath.	negative	Translation†
TA$^{\diamond}$	kadhaiyum natpum vaazhnaal muzhuvathum natputan inaintha etra erakangalin kalavarisai cottlum neengal oru soap oberov paarkkum vagaiyil selgiradhu.	negative	Translation†

Table 10: Sentiment analysis examples from our datasets. †: obtained by translation of the corresponding EN sentence using IndicTrans MT (Ramesh et al., 2021). ‡: obtained by translation of the corresponding EN sentence using MarianMT. $^{\diamond}$: transliterated using Bing Translator API.

Google Translate API. Further, there is not a lot of difference between the machine and human translations, which might be due to translation bias. Table 10 shows examples from the sentiment analysis datasets in HI-EN, ES-EN, TA-EN and ML-EN.