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

Anonymous ACL-IJCNLP submission

Abstract

Code-switching is a multilingual phenomenon involving the use of multiple languages within a single sentence or conversation. Recently, a new benchmark (GLUECoS) was released for NLP tasks on code-switched languages. In this work, we focus our efforts on arguably the two most challenging natural language understanding tasks of this benchmark: Natural Language Inference (NLI) and Question Answering (QA). We achieve substantial absolute improvements of 7.87% and 20.15% on the mean accuracy and F1 scores over previous state-of-the-art systems for Hindi-English NLI and QA, respectively. We derive these large improvements by fine-tuning pretrained multilingual models on monolingual intermediate tasks and by carefully designing training schemes involving these intermediate tasks. Additionally, we provide an analysis of the impact of translation and transliteration quality on our techniques, along with the effectiveness of joint training with a masked language modeling objective 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), the ability of these models to generalize to code-switched text has not been sufficiently explored. To facilitate systematic investigations into code-switched text, Khanuja et al. (2020b) introduced an evaluation benchmark with a public leaderboard[1], GLUECoS, that covers six different natural language tasks.

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; the scheme 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 show how intermediate-task training could be used to improve performance on two natural language understanding tasks in GLUECoS for code-switched Hindi-English text: Natural Language Inference (NLI) and factoid-based Question Answering (QA). Both these tasks require deeper linguistic reasoning (compared to sequence labeling tasks like part-of-speech tagging or language identification) and exhibit high potential for improvement (Khanuja et al., 2020b). These tasks present an additional challenge since Hindi is present in its Romanized form.[2]. Using intermediate-task training, we significantly outperform previous state-of-the-art approaches by 7.87% and 20.15% on mean accuracy and F1 scores for NLI and QA, respectively. Our main findings are:

- Bilingual intermediate-task training i.e., using both English data and its translation into Hindi (and further transliteration) in the target task, leads to consistent improvements in performance on both NLI and QA. We also show the impact of translation and transliteration quality on bilingual intermediate-task training.
- Pretraining using a masked language modeling (MLM) objective on real code-switched text is domain-sensitive and does not yield performance

[1] https://microsoft.github.io/GLUECoS/

[2] In Khanuja et al. (2020b), only NLI and QA are evaluated on Romanized text

gains when the domains of the target task and the code-switched text differ. HI-EN SINGLE-TASK turns out to be a more reliable training strategy to derive consistent performance improvements.

- On two different pretrained multilingual models, mBERT and XLM-R, we achieve superior performance using our proposed intermediate-task training schemes.

2 Intermediate-Task Training

Intermediate-task training starts with a publicly-available multilingual model that has been pretrained on large volumes of multilingual text using MLM. This model is subsequently fine-tuned using data from one or more intermediate tasks, detailed further below. Finally, the model is fine-tuned on code-switched data from the target tasks.

Single Intermediate-Task Training. We use existing monolingual NLI and QA datasets as intermediate-tasks before fine-tuning on GLUECOS NLI and QA, respectively. We explored the use of three different intermediate tasks in this setting: (i) Task-specific data in English (EN SINGLE-TASK), (ii) Task-specific data in Romanized Hindi (HI SINGLE-TASK), and (iii) Interspersed task-specific data in both English and Romanized Hindi (HI-EN SINGLE-TASK).

Multi Intermediate-Task Training. In this training paradigm, we train the models on two intermediate-tasks (X and Y) simultaneously. This is achieved by using two different task heads, one per task, with the pretrained models. Each batch is loaded with instances from tasks X or Y randomly. We follow Raffel et al. (2020) to sample batches from task X with probability $p_X = \frac{\min(e_X, K)}{\min(e_X, K) + \min(e_Y, K)}$, where e_X and e_Y are the number of training examples in task X and Y, respectively; p_Y 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 X and Y and refer to this system as HI-EN/NLI-QA MULTI-TASK. (We use the merged English and Hindi datasets from HI-EN SINGLE-TASK for each task.) We also explored MLM training on real code-switched text as one of the tasks, in addition to the merged Hindi-English task-specific intermediate-tasks (henceforth referred to as HI-EN/MLM MULTI-TASK). (MLM finetuning was employed as an additional pretraining step in Phang et al. (2020) and Chakravarthy et al. (2020).) The next section provides details of all the datasets we used in our experiments.

3 Datasets and Models

Description of Datasets. The Hindi-English NLI dataset in GLUECOS (Khanuja et al., 2020a) consists of conversational premises and hypotheses, with two labels (entailment and contradiction), and contains 1792/447 examples in the train/test sets, respectively. The Hindi-English QA dataset (Chandu et al., 2018a) in GLUECOS is a factoid-based extractive QA dataset with 259 training question-answer pairs (along with corresponding context) and 54 test question-answer pairs.

As intermediate tasks for NLI and QA, we used English and Hindi versions of the MultiNLI dataset (Williams et al., 2018)[3] and the SQuAD dataset (Rajpurkar et al., 2016),[4] respectively. The Hindi 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 Hindi translations into their corresponding Romanized forms, since NLI and QA in GLUECOS use Romanized Hindi text.

For MLM finetuning, we use a corpus of 64K real code-switched sentences by pooling data listed in prior work (Singh et al., 2018; Swami et al., 2018; Chandu et al., 2018b); we will call it GEN-CS. We supplant this text corpus with an additional 28K code-switched sentences mined from movie scripts (referred to as MOVIE-CS), which is more similar in domain to GLUECOS NLI.

Models and Implementation Details. mBERT is a transformer model (Vaswani et al., 2017) that has been pretrained on the Wikipedia corpus of 104 languages using MLM. 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. It gives better results on cross-lingual tasks including tasks on low-resource languages (Conneau et al., 2020). For all our experiments, we use the *bert-base-multilingual-cased* and *xlm-roberta-base* models from the Transformers library (Wolf et al., 2019). We refer readers to Appendix B for more experimental details.

[3] To be consistent with the Hindi-English NLI task, we removed all instances labeled "neutral" from MultiNLI resulting in 250K/10K examples in the train/development sets.

[4] This dataset consists of 82K/5K question-answer pairs with supporting context in its train/development sets.

	Method	GLUECOS NLI *(acc.)*			GLUECOS QA *(F1)*		
		Max	**Mean**	**Std.**	**Max**	**Mean**	**Std.**
mBERT	Baseline	61.07	57.51	2.58	66.89	64.25	2.6
	+EN SINGLE-TASK	62.40	60.73	1.78	77.62	75.77	1.79
	+HI SINGLE-TASK	63.73	62.09	0.99	79.63	76.77	1.86
	+HI-EN SINGLE-TASK	65.55	64.1	0.89	81.61	79.97	1.29
	+HI-EN/NLI-QA MULTI-TASK	**66.74**	65.3	0.92	83.03	80.25	1.76
	+HI-EN/MLM MULTI-TASK	66.66	**65.61**	0.86	81.05	79.11	1.40
XLM-R	Baseline	-	-	-	56.86	53.22	2.35
	+EN SINGLE-TASK	66.22	63.91	1.86	82.04	80.92	1.4
	+HI SINGLE-TASK	63.24	61.73	0.96	81.48	80.55	0.7
	+HI-EN SINGLE-TASK	65.01	64.37	0.57	82.41	81.36	1.32
	+HI-EN/NLI-QA MULTI-TASK	64.49	64.35	0.14	**83.95**	**82.38**	1.18
	+HI-EN/MLM MULTI-TASK	66.66	65.01	1.36	82.1	80.44	1.38
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 1: Our main results from intermediate-task training (max, mean and standard deviations). 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.

4 Results and Discussion

Table 1 provides a comprehensive evaluation of our proposed techniques. We show that our techniques significantly outperform the baselines and prior work across tasks (NLI and QA) and multilingual models (mBERT and XLM-R[5]). Since the GLUECOS test sets are small in size, we report mean accuracies and F1 scores over 5 runs with different seeds for NLI and QA, respectively.

Benefits of Intermediate-Task Training. Among the SINGLE-TASK systems in Table 1, we observe that HI-EN SINGLE-TASK performs the best (based on mean scores) on both NLI and QA. Using a merged Hindi-English dataset for HI-EN SINGLE-TASK training was critical. (Sequentially training on English followed by Hindi resulted in poor performance, as shown in Appendix D.) Another interesting observation is that XLM-R benefits more from EN SINGLE-TASK while mBERT benefits more from HI SINGLE-TASK, compared to the baseline. This could be attributed to XLM-R having encountered Romanized Hindi text during its pretraining unlike mBERT.

The MULTI-TASK systems yield additional gains. Using both NLI and QA as intermediate tasks consistently benefits both NLI and QA for mBERT and QA for XLM-R. Tarunesh et al. (2021) observed that NLI and QA mutually benefit from each other in a multi-task framework using mBERT. Phang et al. (2020) showed that QA benefits from NLI with XLM-R in a multi-task framework, but the converse does not hold. Both these observations hold in our experiments as well.

Although intermediate-task training is beneficial across tasks, the relative improvements in QA are higher than that for NLI (see Appendix F for some QA examples). We conjecture this is due to varying dataset similarity between intermediate-tasks and target tasks. In QA, this similarity is higher and in NLI the conversational nature and large premise lengths reduces this similarity (see Appendix A for examples). Our observation that task and domain similarity play a crucial role in transferability is consistent with Vu et al. (2020).

MLM Pretraining. Table 1 shows that using MLM on real code-switched text as an intermediate-task is helpful for NLI with both mBERT and XLM-R. However, we notice a small drop in the QA performance. We dissect this difference in performance further in Table 2 by doing MLM training separately on GEN-CS and GEN-CS +MOVIE-CS. For NLI, adding in-domain MOVIE-CS text yields additional improvements over GEN-CS. For QA, using GEN-CS by itself improves performance over HI-EN SINGLE-TASK. However, using both GEN-CS and MOVIE-CS for MLM results in a significant drop in performance. We attribute this largely to the domain mismatch between the text in GLUECOS QA and the text in MOVIE-CS. For both tasks, we notice that adding

[5]As in Chakravarthy et al. (2020), we also find that XLM-R baseline on GLUECOS NLI does not converge. However, our techniques mitigate this artefact.

Task	MLM Data	Max	Mean	Std.
NLI	-	65.55	64.1	0.89
	GEN-CS	65.22	64.19	1.22
	GEN-CS + MOVIE-CS	**66.67**	**65.61**	0.86
	GEN-CS + MOVIE-CS + GLUECOS NLI CS	66.17	65.21	0.96
QA	-	81.61	79.97	1.29
	GEN-CS	**83.03**	**80.38**	1.68
	GEN-CS + MOVIE-CS	81.05	79.11	1.40
	GEN-CS + MOVIE-CS + GLUECOS QA CS	79.63	78.27	1.46

Table 2: Using different datasets for HI-EN/MLM MULTI-TASK training with mBERT.

code-switched text from GLUECOS degrades performance. This could be attributed to artefacts specific to GLUECOS; more details are in Appendix E.

Translation and Transliteration Quality. Our Hindi data for the intermediate tasks was obtained from EXTREME; they used an in-house machine translation system to translate the English datasets to Hindi (in Devanagari) and we further transliterated it. To assess the impact of both translation and transliteration quality on final performance, we use two small datasets for NLI and QA from XNLI (Conneau et al., 2018) and XQuAD (Artetxe et al., 2020) for which we have manual Hindi (Devanagari) translations. The NLI/QA datasets contain 4.2K/2.3K training instances, respectively. We used the corresponding English text to generate Hindi translations using the Google Translate API.[6] For XNLI, the premises and hypotheses were directly translated and for XQuAD we adopted the same translation procedure listed in Hu et al. (2020). We use two transliteration tools, *indic-trans* and the transliteration output that is a by-product of the Google Translate API. Table 3 shows the performance of HI-EN SINGLE-TASK for different choices of translation and transliteration outputs. (Appendix C shows a similar comparative analysis using HI SINGLE-TASK.)

We observe that manual translation and transliteration using the Google API performs the best. For NLI, manual translation followed by transliteration using *indic-trans* outperforms machine-translation and transliteration by the Google API, while for QA the trend is reversed. This indicates that translation and transliteration quality have varying impacts depending on the task. Interestingly, Tables 1 and 3 show that the performance after training on small amounts of bilingual manually translated data is statistically comparable to the HI SINGLE-TASK systems that use much larger amounts of text.

[6] https://pypi.org/project/googletrans/

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 3: Effect of translation and transliteration quality on intermediate-task training for GLUECOS.

5 Related Work

Pires et al. (2019) and Hsu et al. (2019) showed that mBERT is effective for Hindi-English 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). Their mod-mBERT model obtained after MLM pretraining on real and synthetic code-switched data yielded additional improvements in several tasks. Chakravarthy et al. (2020) further improved the NLI performance of mod-mBERT by including large amounts of in-domain code-switched movie script data in the MLM pretraining. They also explored the use of XLM-R by data-augmentation with SNLI (Bowman et al., 2015) and XNLI, but failed to generate improvements on the test set. Intermediate-task training has been previously shown to be effective for cross-lingual zero-shot transfer from English tasks on multilingual models such as XLM-R (Phang et al., 2020) and mBERT (Tarunesh et al., 2021). More broadly, intermediate training on NLI and QA tasks have proven to be effective for many natural language understanding target tasks (Pruksachatkun et al., 2020; Vu et al., 2020).

6 Conclusion

This is the first work to demonstrate the effectiveness of intermediate-task training for code-switched NLI and QA. We substantially improve over previous state-of-the-art with up to 8% and 20% absolute improvements on NLI and QA, respectively. For future work, we plan to extend our investigation to include more language pairs and more intermediate tasks.

References

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.

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.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. 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: Crowdsourcing 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.

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.

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*.

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.

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.

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.

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. 2021. Meta-learning for effective multi-task and multilingual modelling. *arXiv preprint arXiv:2101.10368*.

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.

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.

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 Additional Dataset and Model Details

A.1 XTREME Translate-Train Datasets

As mentioned previously, we use the MultiNLI and SQuAD v1.1 data from the translate-train sets of the XTREME benchmark[7,8]. The Romanized version of 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, and then, doing 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 the transliterated datasets[9]. In Tables 7 and 8, we present some examples from our datasets.

A.2 Masked Language Modelling

In some of the MLM experiments, we used code-switched data from the GLUECoS datasets. In NLI, we split various dialogues from premises (based on the ## separator) and discarded the ones with less than 5 words (due to insufficient context for MLM). This gives us around 6.5K code-switched sentences. Similarly, we split the sentences from the passage in the QA task. This gives us an additional 4.1K code-switched sentences.

A.3 Number of Model Parameters

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.

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 Hindi within each batch. In the single-task systems, we used `batch_size = 8` and `max_sequence_length = 128` for NLI and `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.

B.2 Fine-tuning on GLUECoS Tasks

The base fine-tuning files have been taken from the GLUECoS repository[10]. 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 ~ 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[11]

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

[8] SQuAD: https://storage.cloud.google.com/xtreme_translations/SQuAD/translate-train/squad.translate.train.en-hi.json.

[9] Available at: http://bit.ly/3tkgyWy

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

[11] 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.

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 4: Effect of translation and transliteration quality during intermediate-task training on GLUECOS results.

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.

C Experiments on Translation and Transliteration Quality

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).

Similar to Table 3 for bilingual text, we present a similar analysis of romanized Hindi text in Table 4 for mBERT. All the relative trends remain the same.

In Tables 7 and 8, we show some instances from the datasets. 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 Hindi language, and thus, have a greater impact on model performance. We also note that transliteration of these common words in the GLUECOS dataset matches closely with the transliteration produced using the Google Translate API. Further, there is not a lot of difference between the machine and human translations, which might be due to translation bias.

D Alternate Bilingual Training Paradigm

To further probe the effectiveness of bilingual task-specific intermediate training, we train on the monolingual task data sequentially. That is, we first train on the English corpus and then on the Romanized Hindi corpus from XTREME. We observe that this results in relatively poor performance on both tasks, even worse than the monolingual counterparts. This indicates that just additional data is not sufficient and our scheme of simultaneous bilingual training is important to achieve good performance. Table 5 shows the performance of various techniques and validates our observations.

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

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

Table 5: Sequential bilingual training yields poorest performance on both the tasks using the mBERT model.

E MLM + Intermediate-Task Training

Table 6 provides a comprehensive summary of our experiments exploring intermediate-task training of mBERT in conjunction with MLM in a multi-task

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 6: 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.

Language	Premise/ Hypothesis	Label	Dataset
HI-EN	**PREMISE:** CLERK : Yeh ? ## CLERK : Yeh toh guzar gaya . Haadsa ho gaya ek . ## DEVI : Iski ye file hai humaare paas . Kuch paise baaki hain , ek do books bhi hain uski jo lautani hain . Aap pata de sakte hain ? ## CLERK : Number hai ghar ka . Ye addresss hai unka Allahabad mein . Note karo . Par bolna mat kahin ki maine diya hai . ## **HYPOTHESIS:** CLERK ki kuch files DEVI ke paas hain.	contradictory	GLUECOS NLI
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 transliteration by *indic-trans* and their counterparts. In *purple*, we show some words that are better transliterated by *indic-trans*. Best viewed in color.

framework. (Some of the numbers from Table 2 are reproduced here, for more clarity.) As described in Section 3, our setup is similar to Raffel et al. (2020). The first key take-away from Table 6 is that intermediate training using MLM on code-switched data alone is not effective (first row of each task).

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. We do not present the experiment with MLM on only the MOVIE-CS data that is relatively smaller in size because of its inferior performance. 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 because of the fact that the domain of the passages in GLUECOS QA is similar to the Hindi-English 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 GLUECOS tasks.

For both these tasks, we observe a common trend: Adding code-switched data from the training set of GLUECOS tasks 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 characters as *'h u m k o'*. Thus, MLM on such data may not be very effective and could hurt performance. The reasoning in the case of QA is different: For a significant portion of the train set, the passage is obtained using *DrQA - Document Retriever module*[13] (Chen et al., 2017). These passages are monolingual in nature and thus not useful for MLM training with code-switched text.

F Example Outputs from GLUECOS QA

The examples below are outputs on the GLUECOS QA test set. Our techniques improve the ability

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

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 Championship 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 8: 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.

of the multilingual models to do deeper reasoning. Consider the following question which is correctly answered by the baseline mBERT:

Q: *"Continuum ye feature kaunsi company ne launch kiya hai?"* ("Which company launched the continuum feature?")

A: *Microsoft*

However the question,

Q: *"Microsoft ke kaunse employee nein Continuum ka kamaal dikhaya?"* ("Which employee of Microsoft showed the wonders of Continuum?")

A: *Joe Belfear*

is not correctly answered. Our EN SINGLE-TASK system (trained on English SQuAD) answers both questions correctly.

Our EN SINGLE-TASK system is also able to correctly answer questions that invoke numerical answers, such as:

Q: *"4G SIM paane ke kitne option hai??"* ("How many options are there to get a 4G SIM?").

A: 2