Adapting Multilingual Models for Code-Mixed Translation

Aditya Vavre[1], **Abhirut Gupta**[2], and **Sunita Sarawagi**[1]
[1]IIT Bombay [2]Google Research
{adityavavre,sunita}@cse.iitb.ac.in, abhirut@google.com

Abstract

The scarcity of gold standard code-mixed to pure language parallel data makes it difficult to train translation models reliably. Prior work has addressed the paucity of parallel data with data augmentation techniques. Such methods rely heavily on external resources making systems difficult to train and scale effectively for multiple languages. We present a simple yet highly effective two-stage back-translation based training scheme for adapting multilingual models to the task of code-mixed translation which eliminates dependence on external resources. We show a substantial improvement in translation quality (measured through BLEU), beating existing prior work by up to +3.8 BLEU on code-mixed Hi→En, Mr→En, and Bn→En tasks. On the LinCE Machine Translation leader board, we achieve the highest score for code-mixed Es→En, beating existing best baseline by +6.5 BLEU, and our own stronger baseline by +1.1 BLEU.

1 Introduction

As code-mixing (Diab et al., 2014; Winata et al., 2019; Khanuja et al., 2020; Aguilar et al., 2020) becomes widespread in an increasingly digitized bilingual community, it becomes important to extend translation systems to handle code-mixed input. A major challenge for training code-mixed translation models is the lack of parallel data. Recent work on generating synthetic parallel data using available non-code-mixed parallel data depend on language specific tools for transliteration, word-alignment, and language identification (Gupta et al., 2021). This makes the approach difficult to scale to new languages and increases software complexity. Back-translation (BT) is another effective and popular strategy to handle non-availability of parallel data (Sennrich et al., 2016; Edunov et al., 2018). However, for the code-mixed to English translation task, simple BT is not an option since we cannot assume the presence of an English to code-mixed translation model.

Meanwhile the mainstream translation community is converging on frameworks based on multilingual models for translation between multiple language pairs (Johnson et al., 2017; Aharoni et al., 2019; Arivazhagan et al., 2019; Zhang et al., 2020; Fan et al., 2021). Going forward, code-mixed translation needs to be integrated within these frameworks to impact practical systems.

We propose a novel two stage back-translation methodology called Back-to-Back Translation (B2BT) targeted for adapting multilingual models to code-mixed translation. Our approach is simple and integrates easily with existing multilingual translation models without any need for special models or language specific tools. We compare B2BT with six other baselines on both standalone and mBART-based models across four benchmarks and show significant gains. For example, on code-mixed Hindi to English translation B2BT improves state-of-art accuracy by +3.8 and by +6.3 over default back-translation. We analyze the reasons for the gains via both human evaluation and impact on downstream models. We release a new dataset and will publicly release our code.

2 Our Approach

Our objective is to train a model that can translate a sentence from the code-mixed language $\mathcal{C}$, which contains words from English and an additional language $\mathcal{S}$, to monolingual English $\mathcal{E}$. Following (Myers-Scotton, 1997) we refer to $\mathcal{S}$ as the *matrix language* as it lends its grammar in a code-mixed utterance, and English as the *embedded language* since it lends only its words. We are given parallel $\mathcal{S}$ to English corpus $(S, E) \subset (\mathcal{S}, \mathcal{E})$ and a non-parallel code-mixed corpus $C \subset \mathcal{C}$. Since code-mixing appears more in domains like social media, which differ from formal domains like news in which parallel data (S, E) is available, we addi-

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 7162 - 7170
December 7-11, 2022 ©2022 Association for Computational Linguistics

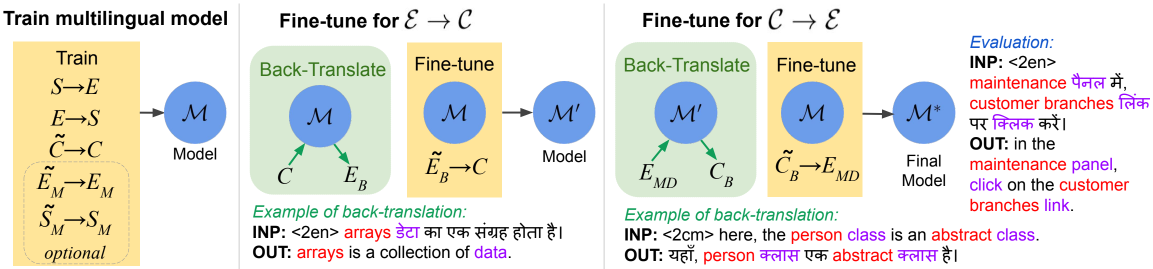

Figure 1: B2BT training pipeline, showing the two-stage back-translation based adaptation of an initial multilingual model. $(\tilde{\cdot})$ indicates source side masking during training.

tionally use a domain-specific monolingual English corpora $E_{MD} \subset \mathcal{E}$. Optionally, we can also exploit monolingual data $S_M \subset \mathcal{S}$ and $E_M \subset \mathcal{E}$. Our method called B2BT of training a $\mathcal{C} \rightarrow \mathcal{E}$ translator without parallel data is summarised in Figure 1 and comprises of an initial training of a multi-lingual model and two stages of back-translation-based fine-tuning that we elaborate on next.

Training Base Multilingual Model The first step is to train a multilingual model ($\mathcal{M}$) on parallel matrix language to English corpus (S, E) in both directions and non-parallel data in English E_M, matrix language S_M, and code-mixed C. Following Johnson et al. (2017) we prefix source sentences with one of <2en>, <2cm>, and <2xx> directing target as English, CM, or $\mathcal{S}$ respectively. For the non-parallel corpora, we train the model to copy the source to the target by masking out 20% tokens in the source as used in (Song et al., 2019b).

The above training exposes $\mathcal{M}$ to all three languages in both encoder and decoder, and a baseline is to just use this bidirectional model for our task. We will show that such a model provides marginal gains over a simple $\mathcal{S} \rightarrow \mathcal{E}$ model. However, we adapt $\mathcal{M}$ further using synthetic parallel data for the $\mathcal{C} \rightarrow \mathcal{E}$ task. Back-translation (BT) of $\mathcal{E}$ to $\mathcal{C}$ using $\mathcal{M}$ to generate synthetic parallel data provides very poor quality as we show in Section 4. This motivates our two stage BT approach. A key insight of B2BT method is that $\mathcal{M}$ trained with parallel $\mathcal{S} \rightarrow \mathcal{E}$ data gives better quality outputs when translating $\mathcal{C}$ to $\mathcal{E}$ than the reverse. The reason is $\mathcal{C}$ shares the grammar structure of $\mathcal{S}$ and $\mathcal{M}$ is trained to handle noise in the input. We describe the two step BT next.

Fine-tune for $\mathcal{E} \rightarrow \mathcal{C}$ Here we prepare $\mathcal{M}$ to back-translate pure English sentences to code-mixed sentences so that the resulting synthetic parallel data can be used to train a better code-mixed to English translation model. We first back-translate the monolingual code-mixed corpus C to English E_B using $\mathcal{M}$. The back-translation is done by prefixing <2en> to the code-mixed input and sampling English output from $\mathcal{M}$. This provides us with a synthetic English to code-mixed parallel corpus (E_B, C). We fine-tune $\mathcal{M}$ on (E_B, C) to produce a model $\mathcal{M}'$ where source sentences are prefixed with <2cm>. Since the target distribution C is preserved during training, we can now generate high quality in-domain code-mixed sentences using $\mathcal{M}'$.

Fine-tune for $\mathcal{C} \rightarrow \mathcal{E}$ In the final step we realise our objective of $\mathcal{C} \rightarrow \mathcal{E}$ translation. We start by back-translating the in-domain monolingual English corpus E_{MD} to code-mixed C_B using $\mathcal{M}'$. This is done by prefixing English sentences with the <2cm> tag, and sampling code-mixed outputs from $\mathcal{M}'$. We now have a synthetic code-mixed to English parallel corpus (C_B, E_{MD}). We fine-tune $\mathcal{M}$ to obtain our final model $\mathcal{M}^*$ on this synthetic parallel corpus where all the source sentences in C_B are prefixed with the <2en> token.

3 Related Work

Code-mixing is receiving increasing interest in the NLP community Khanuja et al. (2020); Diab et al. (2014); Aguilar et al. (2018); Solorio et al. (2021); Song et al. (2019a). A primary focus area is training code-switched language models for applications like speech recognition (Winata et al., 2019; Gonen and Goldberg, 2019) under limited code-mixed (CM) data. Pratapa et al. (2018); Chang et al. (2019); Gao et al. (2019); Samanta et al. (2019); Winata et al. (2019) all propose different methods for creating synthetic CM data to augment training data. Tarunesh et al. (2021) generates CM sentences by extending a translation model. The above papers are designed for LM training and do not generate $(\mathcal{C}, \mathcal{E})$ parallel data.

The biggest challenge in translation of code-mixed sentences is the lack of large parallel training data (Mahesh et al., 2005; Menacer et al., 2019; Nakayama et al., 2019; Srivastava and Singh, 2020). Gupta et al. (2021) propose to create synthetic parallel CM data via these two steps: (1) train an mBERT model to identify word set W to switch in a sentence from $\mathcal{S}$ to $\mathcal{E}$, effectively creating a sentence from $\mathcal{C}$ (2) align parallel sentences from (S, E) and replace words in W to their aligned English words. We call this the mBertAln method in this paper. This pipeline for a new language $\mathcal{S}$ requires the following four external tools: (1) mBERT pre-trained on $\mathcal{S}$, (2) a language identifier tool to spot English tokens in a CM sentence, (3) a word alignment model, and (4) a translator $\mathcal{E} \rightarrow \mathcal{S}$ for BT. For low-resource languages such tools may not exist. In contrast B2BT is totally standalone. Even when external tools exist, we show empirically that the synthetic sentences thus generated tend to be of lower quality than ours because of errors in any of the two steps. The CALCS 2021 workshop (Solorio et al., 2021) also released a shared task for CM translation but the submissions so far are straight-forward application of BART multilingual models, with which we also compare our method.

B2BT is reminiscent of dual learning NMT methods (He et al., 2016; Artetxe et al., 2018; Hoang et al., 2018; Cheng et al., 2016) but these methods were designed for two generic languages whereas B2BT for code-mixed translation handles three languages related in specific asymmetric ways. We exploit that asymmetry to design our training schedule. For example, since $\mathcal{C} \rightarrow \mathcal{E}$ translations are more accurate than the reverse we insert the intermediate BT stage.

4 Experiments

We use the notation SoEn→En, to indicate translation from a code-mixed matrix language with code 'So' to English. We evaluate on four code-mixed datasets: Hindi (HiEn→En) from Gupta et al. (2021), Spanish (EsEn→En) on the LinCE leaderboard [1], Bengali (BnEn→En) from Gupta et al. (2021) but augmented with the newly released Samanantar data to create a stronger baseline (evaluation is done on the splits released by the authors), and a new Marathi (MrEn→En) dataset that we introduce [2]. A summary of the training data used, and our model setup is in Appendix A and B.

Lang Pair	Method	ST-Test	ST-OOV	ST-Hard
HiEn →En	Hi→En Model	36.9	33.9	2.1
	Hi→En Model + BT	43.9	41.4	18.6
	mBertAln	46.4	44.6	23.4
	Multilingual	38.0	37.7	17.5
	Multilingual + $\mathcal{E} \rightarrow \mathcal{S}$ BT	44.0	40.9	22.6
	Multilingual + $\mathcal{E} \rightarrow \mathcal{C}$ BT	35.7	35.8	20.6
	B2BT	**50.2**	**49.9**	**30.7**
BnEn →En	Bn→En Model	30.8	31.1	14.1
	Bn→En Model + BT	40.9	41.2	21.2
	mBertAln	41.4	41.9	22.3
	Multilingual	30.9	31.4	13.8
	Multilingual + $\mathcal{E} \rightarrow \mathcal{S}$ BT	41.7	42.0	22.0
	B2BT	**44.2**	**43.4**	**23.4**
MrEn →En	Mr→En Model	26.6	25.7	0.9
	Mr→En Model + BT	39.3	39.2	16.5
	mBertAln	40.6	40.5	17.8
	Multilingual	29.1	29.7	9.0
	Multilingual + $\mathcal{E} \rightarrow \mathcal{S}$ BT	**41.4**	**41.5**	**18.9**
	B2BT	**41.2**	**41.3**	**18.7**

Table 1: Comparing BLEU scores for B2BT trained from scratch against other baselines including mBertAln of Gupta et al. (2021). *ST-OOV* and *ST-Hard* are subsets of the test set (*ST-Test*) containing sentences with at least two OOV words, and 2,000 sentences the base model performed poorest on respectively.

Baselines We compare our method, B2BT against the mBertAln model (Gupta et al., 2021) and these baselines: (1) the base bi-lingual $\mathcal{S} \rightarrow \mathcal{E}$ model, (2) base model fine-tuned with $\mathcal{E} \rightarrow \mathcal{S}$ BT on domain data E_{MD} , (3) base multilingual model $\mathcal{M}$ obtained after first stage of B2BT, (4) $\mathcal{M}$ fine-tuned with $\mathcal{E} \rightarrow \mathcal{S}$ BT on domain data E_{MD} , (5) $\mathcal{M}$ fine-tuned with $\mathcal{E} \rightarrow \mathcal{C}$ BT on E_{MD} .

Results Table 1 compares B2BT approach against these baselines on HiEn→En, BnEn→En, and MrEn→En. Observe how B2BT significantly outperforms mBertAln and multilingual model adapted with existing single step back-translation across all language pairs. We also see substantial improvements on the two adversarial subsets *ST-OOV* and *ST-Hard*. This establishes the importance of our two-stage back-translation approach. Note in particular that when we fine-tuned with E_{MD} back-translated to code-mixed with $\mathcal{M}$, we observe a huge drop in accuracy! This is because the base multilingual model ($\mathcal{M}$) trained to denoise CM data and translate $\mathcal{S} \rightarrow \mathcal{E}$ is much worse for $\mathcal{E} \rightarrow \mathcal{C}$ translations than $\mathcal{C} \rightarrow \mathcal{E}$. This underlines

[1]https://ritual.uh.edu/lince/leaderboard

[2]Our data is available at https://github.com/adityavavre/spoken-tutorial-codemixed

Lang Pair	Method	BLEU
HiEn →En	mBART Multilingual	35.1
	mBART Multilingual + $\mathcal{E} \to \mathcal{S}$ BT	43.4
	mBART Multilingual B2BT	**48.0**
EsEn →En	mBART (leaderboard)	43.9
	mBART Multilingual	49.3
	mBART Multilingual + $\mathcal{E} \to \mathcal{S}$ BT	**50.0**
	mBART Multilingual B2BT	**50.4**

Table 2: Results comparing B2BT fine-tuned on an mBART checkpoint against baselines and best existing models on the LinCE leaderboard.

Fine-tuning Dataset for Final Model	ST-Test
B2BT ($\mathcal{M}^*$)	50.2
$\mathcal{M}$ + synthetic data from Gupta et al. (2021)	45.3

Table 3: Comparing BLEU on HiEn→En when using synthetic code-mixed data generated from $\mathcal{M}'$ in B2BT vs synthetic data from mBertAln

the importance of the intermediate model ($\mathcal{M}'$) that is fine-tuned to produce good code-mixed data from English.

Our approach can also complement existing multilingual pre-trained models such as mBART. Table 2 presents results with base multilingual model $\mathcal{M}$ trained by fine-tuning an mBART checkpoint. Here again we observe gains beyond simple BT-based fine-tuning of the multilingual model.

Why does B2BT outperform mBertAln? We hypothesize that the reason our model performs substantially better is that the synthetic data generated by our model is of higher quality. To test this hypothesis we replace the synthetic code-mixed parallel data of B2BT with synthetic data from mBertAln (Gupta et al., 2021) while keeping the rest of the training of $\mathcal{M}^*$ unchanged. Table 3 presents this result. It is important to note that all the fine-tuning sets have the exact same size and all fine-tuning is performed on the same multilingual base model, $\mathcal{M}$. The only difference is in the method used to create the synthetic side of the fine-tuning dataset. The improvement of almost +4.9 BLEU points on ST-Test over using mBertAln

English Sentence	mBERT Synth Code-Mixed	B2BT Synth Code-Mixed
open layer properties dialog box again.	परत properties dialog फिर से खोलें. layer again open	फिर से layer properties डायलॉग बॉक्स खोलें। again dialog box open
click on open button.	खुले बटन पर ओपन करें. Open button on open	open बटन पर क्लिक करें। button on click

Figure 2: Examples of synthetic sentences from mBertAln vs B2BT. English translations of Devanagari words are provided.

Metric	ST-Test	mBertAln	B2BT
Human eval rating	-	3.74	4.27
Human eval win %	-	17%	39%
Code-Mixing Index	28.3	20.7	27.2
Common En tokens	0.16	0.20	0.18
Code switch probability	0.27	0.24	0.27

Table 4: Comparing the synthetic data generated through mBertAln against B2BT.

data, clearly shows that the synthetic data from our model has better quality.

To directly quantify this fact, we performed human evaluation of data quality. Human raters were asked to rate fluency and intent preservation for source-target pairs (similar to Wu et al. (2016)) on a scale of 0 (irrelevant) to 6 (perfect). Across 500 examples, we observe that synthetic data from B2BT is rated as 4.27 out of 6 on average compared to 3.74 for mBertAln. In 39% of examples B2BT is rated higher than mBertAln, 45% of examples get the same score, and only in 17% examples is mBertAln better (Table 4). In mBertAln the quality of synthetic data could suffer because of poor back-translation, mBERT failing to capture the code-switching pattern, or the alignment model failing to predict the aligned English token. Figure 2 presents examples of synthetic sentences generated by B2BT vs mBertAln. The mBertAln method has word repetition like “open" in row 2, which could be an alignment mistake, and word omissions like “box" in row 1 which could be caused by poor back-translation or alignment.

Finally, we compare code-mixing statistics between the synthetic data generated by B2BT and mBERT in Table 4. The data generated from B2BT is closer to the test data in terms of Code-Mixing Index, fraction of English tokens common in the source and target, and the average probability of switching at a given word.

Varying degree of code-mixing Following Gupta et al. (2021), we also evaluate the effectiveness of our model across different splits of the test set with varying Code-Mixing Index (Gambäck and Das, 2016) (CMI). Figure 3 presents the improvements from our model on the three splits of the test set. We see improvements across all splits, but the largest improvements are on the split with the highest degree of code-mixing. On the high CMI split, we see about +8.7 BLEU point improvement over the mBERT approach, and +14.5 BLEU point improvement over the baseline.

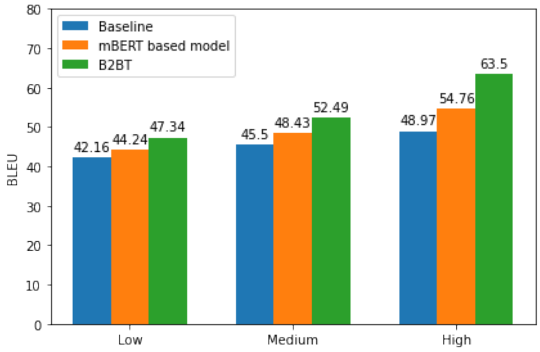

Figure 3: Improvements in BLEU with B2BT against the mBERT based model and the domain-adapted bilingual model baseline across three splits of the test set with varying degree of code-mixing in the source.

Lang Pair	Fine-tuning Approach	BLEU
HiEn→En	Un-masked	50.1
	Masked	50.2
BnEn→En	Un-masked	42.8
	Masked	44.2
MrEn→En	Un-masked	40.6
	Masked	41.2

Table 5: Comparing BLEU on ST-Test between masked vs un-masked fine-tuning to train $\mathcal{M}^*$ in the B2BT approach.

Masking during fine-tuning in B2BT A distinctive property of code-mixed translation is word overlap between the source and target sentences. Such overlap makes the fine-tuned model overly biased towards the easier copy action. We alleviate this bias by introducing random masking of words in the source sentence (with masking probability 0.2). Unlike prior work (Song et al., 2019b) which apply such masking only for pre-training with mononlingual corpora, we propose to mask tokens even when training with parallel data. We evaluate the impact of this source side masking in B2BT's fine-tuning stages. Table 5 compares model performance with and without source side masking when fine-tuning. We observe noticeable gains, with the highest for BnEn at +1.5.

5 Conclusion

We present a simple two-stage back-translation approach (B2BT) for adapting multilingual models for code-switched translation. B2BT shows remarkable improvements on four datasets compared to recent methods, and default back-translation baselines. Our approach fits naturally with existing multilingual translation frameworks, which is crucial in expanding coverage to low resource languages without building per-language pair models. We demonstrate with ablation studies and human evaluations that the synthetic data created through the two step process in B2BT is objectively higher quality than the one used by existing work.

6 Limitations

Our method depends on code-mixed monolingual data which may not be always available. Additionally, for low resource languages, we might not have access to enough non-code-mixed parallel data which also forms a crucial component of our approach.

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.

Gustavo Aguilar, Sudipta Kar, and Thamar Solorio. 2020. LinCE: A Centralized Benchmark for Linguistic Code-switching Evaluation. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 1803–1813, Marseille, France. European Language Resources Association.

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. 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 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. Unsupervised statistical machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3632–3642, Brussels, Belgium. Association for Computational Linguistics.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics.

Ching-Ting Chang, Shun-Po Chuang, and Hung-Yi Lee. 2019. Code-Switching Sentence Generation by Generative Adversarial Networks and its Application to Data Augmentation. In *Proc. Interspeech 2019*, pages 554–558.

Yong Cheng, Wei Xu, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Semi-supervised learning for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1965–1974, Berlin, Germany. Association for Computational Linguistics.

Mona Diab, Julia Hirschberg, Pascale Fung, and Thamar Solorio, editors. 2014. *Proceedings of the First Workshop on Computational Approaches to Code Switching*. Association for Computational Linguistics, Doha, Qatar.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Michael Auli, and Armand Joulin. 2021. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(107):1–48.

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

Yingying Gao, Junlan Feng, Ying Liu, Leijing Hou, Xin Pan, and Yong Ma. 2019. Code-Switching Sentence Generation by Bert and Generative Adversarial Networks. In *Proc. Interspeech 2019*, pages 3525–3529.

Hila Gonen and Yoav Goldberg. 2019. Language modeling for code-switching: Evaluation, integration of monolingual data, and discriminative training. 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 4175–4185, Hong Kong, China. Association for Computational Linguistics.

Abhirut Gupta, Aditya Vavre, and Sunita Sarawagi. 2021. Training data augmentation for code-mixed translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5760–5766, Online. Association for Computational Linguistics.

Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. 2016. Dual learning for machine translation. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24, Melbourne, Australia. Association for Computational Linguistics.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

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

Anoop Kunchukuttan. 2020. The IndicNLP Library. `https://github.com/anoopkunchukuttan/indic_nlp_library/blob/master/docs/indicnlp.pdf`.

Anoop Kunchukuttan, Pratik Mehta, and Pushpak Bhattacharyya. 2018. The IIT Bombay English-Hindi parallel corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

R. Mahesh, K. Sinha, and Anil Thakur. 2005. Machine translation of bi-lingual hindi-english (hinglish) text. In *MTSUMMIT*.

Mohamed Menacer, David Langlois, Denis Jouvet, Dominique Fohr, Odile Mella, and Kamel Smaïli. 2019. Machine Translation on a parallel Code-Switched Corpus. In *Canadian AI 2019 - 32nd Conference on Canadian Artificial Intelligence*, Lecture Notes in Artificial Intelligence, Ontario, Canada.

Carol Myers-Scotton. 1997. *Duelling languages: Grammatical structure in codeswitching*. Oxford University Press.

Sahoko Nakayama, Takatomo Kano, Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura. 2019. Recognition and translation of code-switching speech utterances. In *2019 22nd Conference of the Orien-*

tal COCOSDA International Committee for the Co-ordination and Standardisation of Speech Databases and Assessment Techniques (O-COCOSDA), pages 1–6.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Adithya Pratapa, Gayatri Bhat, Monojit Choudhury, Sunayana Sitaram, Sandipan Dandapat, and Kalika Bali. 2018. Language modeling for code-mixing: The role of linguistic theory based synthetic data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1543–1553, Melbourne, Australia. 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. *CoRR*, abs/2104.05596.

Bidisha Samanta, Sharmila Reddy, Hussain Jagirdar, Niloy Ganguly, and Soumen Chakrabarti. 2019. A deep generative model for code switched text. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5175–5181. International Joint Conferences on Artificial Intelligence Organization.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

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

Kai Song, Yue Zhang, Heng Yu, Weihua Luo, Kun Wang, and Min Zhang. 2019a. Code-switching for enhancing NMT with pre-specified translation. 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 449–459, Minneapolis, Minnesota. Association for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019b. MASS: Masked sequence to sequence pre-training for language generation. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5926–5936. PMLR.

Vivek Srivastava and Mayank Singh. 2020. PHINC: A parallel Hinglish social media code-mixed corpus for machine translation. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 41–49, Online. Association for Computational Linguistics.

Ishan Tarunesh, Syamantak Kumar, and Preethi Jyothi. 2021. From machine translation to code-switching: Generating high-quality code-switched text. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3154–3169, Online. 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*, volume 30. Curran Associates, Inc.

Genta Indra Winata, Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2019. Code-switched language models using neural based synthetic data from parallel sentences. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 271–280, Hong Kong, China. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639, Online. Association for Computational Linguistics.

Dataset	Source	Size	Avg. tokens/sentence
		HiEn→En	
Test	ST-Test	30K	HiEn-14.46, En-13.09
(S, E)	IITB Parallel	1.5M	Hi-15.47, En-14.47
C	ST CM mono	40K	14.49
E_{MD}	ST En mono	53K	12.59
S_M	News Crawl	2M	18.95
		BnEn→En	
Test	ST-Test	29K	BnEn-11.32, En-13.31
(S, E)	Samanantar	2M	Bn-12.14, En-13.56
C	ST CM mono	31K	11.23
E_{MD}	ST En mono	57K	12.31
S_M	IndicCorp	2M	21.15
		MrEn→En	
Test	ST-Test	28K	MrEn-11.32, En-13.00
(S, E)	Samanantar	2M	Mr-10.86, En-12.43
C	ST CM mono	38K	11.14
E_{MD}	ST En mono	57K	12.58
S_M	IndicCorp	2M	16.22
		EsEn→En	
Test	LinCE	6.5K	EsEn-19.72, En-UNK
(S, E)	WMT 2013	2M	Es-33.32, En-29.74
C	LinCE	15K	19.67
E_{MD}	LinCE	15K	15.36
S_M	News Crawl	2M	28.19
E_M	News Crawl	2M	23.90

Table 6: Brief statistics of the datasets used for each language pair. The English target for EsEn→En is private and results are obtained through submission to the leaderboard.

A Datasets

We describe the evaluation sets and all the different types of training datasets used for our experiments.

Code-Mixed Parallel Test Corpus The Spoken Tutorial test sets are created by scraping and aligning transcripts for video lectures in multiple languages including English from the educational website Spoken Tutorial[3]. The video transcripts for Indian languages (like Hindi, Bengali, and Marathi) are heavily code-mixed, containing a large number of English words.

The Computational Approaches to Linguistic Code-Switching worksop (CALCS), 2021, released a code-mixed translation shared task. The code-mixing machine translation test sets are a part of the LinCE Benchmark (Aguilar et al., 2020). We conduct experiment with the EsEn→En (referred to as the Spanglish-English task on the leaderboard) test set as this exactly matches our setting.

Parallel Corpus (S, E) For HiEn→En experiments, we use the IIT Bombay English-Hindi Parallel Corpus (Kunchukuttan et al., 2018) as the base parallel training data (S, E) for our models. Test and validation splits are from the WMT 2014 English-Hindi shared task (Bojar et al., 2014). We move about 2,000 randomly selected sentences from the training set to augment the small (500 sentences) validation set. For BnEn→En and MrEn→En, we use 2M randomly sampled parallel sentences from Samanantar (Ramesh et al., 2021) as our parallel data (S, E) for training and 2000 randomly sampled pairs each for validation and testing. For EsEn→En, we use 2M randomly sampled sentence pairs from the Common Crawl corpus released by WMT 2013.

Non-Parallel Code-Mixed Corpus (C) We collect all code-mixed sentences from the Spoken Tutorial Project that are not a part of the parallel test data. For the EsEn→En task on the LinCE leaderboard, a set of 15K code-mixed Spanish sentences are provided as a part of the setup.

Monolingual Corpora $(E_{MD}$, E_M, $S_M)$ For the in-domain English corpus (E_{MD}), we collect sentences from Spoken Tutorial transcripts which are not a part of the parallel test data. For the EsEn→En task on the LinCE leaderboard, we use the monolingual English tweets provided for the reverse translation task as the in-domain monolingual corpus.

We use the News Crawl corpus of WMT 2014 as the additional monolingual English data (E_M) for all experiments. For the monolingual matrix language (S_M), we use the News Crawl corpus of WMT 2014 for HiEn→En. For BnEn→En and MrEn→En, we use the IndicCorp Bengali and Marathi monolingual corpus [4] respectively. For EsEn→En, we use the News Crawl corpus from WMT 2013.

B Model Setup

All models are trained with the Fairseq toolkit (Ott et al., 2019). We experiment with two types of multilingual models: (1) standalone models that we train only on the given corpus above, and (2) mBART initialized models. During decoding we use a beam size of 5 in all experiments. The BLEU scores are computed using the mosesdecoder script [5].

[3]https://spoken-tutorial.org/

[4]https://indicnlp.ai4bharat.org/corpora/

[5]https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu-detok.perl

Standalone Multilingual Models For training all non-mBART models, we use the standard transformer architecture from Vaswani et al. (2017) with six encoder and decoder layers. In the data pre-processing step, we first tokenize with Indic-NLP (Kunchukuttan, 2020) tokenizer for Indic language sentences and code-mixed sentences and Moses tokenizer [6] for pure English sentences. Next, we apply BPE with code learned on monolingual English and monolingual non-code-mixed datasets jointly, for 20,000 operations (the resulting dictionary is manually appended with the special tokens <2en>, <2xx>, <2cm> and <M>). We use Adam optimizer with a learning rate of 5e-4 and 4000 warmup steps. We train all models for up to 100 epochs and select the best checkpoint based on loss on the validation split. For the two BT based fine-tuning stages in B2BT we use a constant learning rate of 1e-4 and use a random 2K subset of the BT data as the validation split.

Pre-trained mBART-based Multilingual Models The mBART models are trained by fine-tuning the CC25 mBART checkpoint. The model has 12 encoder and decoder layers, with model dimension of 1024 and 16 attention heads (~610M parameters). We modify the existing sentence piece model by adding the three special tokens <2en>, <2xx> and <2cm>, so they are not tokenized and also add them to the dictionary by replacing three tokens in a language we are not currently experimenting with. The multilingual model is trained for 100K steps, while fine-tuning stages of B2BT are trained for up to 25K steps.

[6] https://github.com/moses-smt/mosesdecoder