Adapting Multilingual Models for Code-Mixed Translation

Aditya Vavre, Abhirut Gupta, Sunita Sarawagi
1IIT Bombay 2Google 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 $C$, which contains words from English and an additional language $S$, to monolingual English $E$. Following (Myers-Scotton, 1997) we refer to $S$ 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 $S$ to English corpus $(S, E) \subset (S, E)$ and a non-parallel code-mixed corpus $C \subset C$. 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-
We will show that such a model provides marginal insight of B2BT method is that to just use this bidirectional model for our task. However, we adapt M further using synthetic parallel data for the \( C \rightarrow E \) task. Back-translation (BT) of \( E \rightarrow C \) using \( M \) 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 \( M \) trained with parallel \( S \rightarrow E \) data gives better quality outputs when translating \( C \rightarrow E \) than the reverse. The reason is \( C \) shares the grammar structure of \( S \) and \( M \) is trained to handle noise in the input. We describe the two step BT next.

Fine-tune for \( C \rightarrow \mathcal{E} \) Here we prepare \( M \) 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 \( M \). The back-translation is done by prefixing \( <2en> \) to the code-mixed input and sampling English output from \( M \). This provides us with a synthetic English to code-mixed parallel corpus \((E_B, C)\). We fine-tune \( M \) on \((E_B, C)\) to produce a model \( M' \) 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 \( M' \).

Fine-tune for \( \mathcal{E} \rightarrow C \) In the final step we realise our objective of \( \mathcal{E} \rightarrow C \) translation. We start by back-translating the in-domain monolingual English corpus \( E_{MD} \) to code-mixed \( C_B \) using \( M' \). This is done by prefixing English sentences with the \( <2cm> \) tag, and sampling code-mixed outputs from \( M' \). We now have a synthetic code-mixed to English parallel corpus \((C_B, E_{MD})\). We fine-tune \( M \) to obtain our final model \( M'' \) 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 \((C, \mathcal{E})\) 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 S to E, effectively creating a sentence from C (2) align parallel sentences from (S, E) and replace words in W to their aligned English words. We call this the mBERT method in this paper. This pipeline for a new language S requires the following four external tools: (1) mBERT pre-trained on S, (2) a language identifier tool to spot English tokens in a CM sentence, (3) a word alignment model, and (4) a translator E → 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 C → 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.

#### Baselines

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

#### 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 EMD back-translated to code-mixed with M, we observe a huge drop in accuracy! This is because the base multilingual model (M) trained to denoise CM data and translate S → E is much worse for E → C translations than C → E. This underlines

---

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

2Our data is available at https://github.com/adityavavre/spoken-tutorial-codemixed
Table 2: Results comparing B2BT fine-tuned on an mBART checkpoint against baselines and best existing models on the LinCE leaderboard.

<table>
<thead>
<tr>
<th>Lang Pair</th>
<th>Method</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>HiEn → En</td>
<td>mBART Multilingual</td>
<td>35.1</td>
</tr>
<tr>
<td></td>
<td>mBART Multilingual + E → S BT</td>
<td>43.4</td>
</tr>
<tr>
<td></td>
<td>mBART Multilingual B2BT</td>
<td>48.0</td>
</tr>
<tr>
<td>EsEn → En</td>
<td>mBART (leaderboard)</td>
<td>43.9</td>
</tr>
<tr>
<td></td>
<td>mBART Multilingual</td>
<td>49.3</td>
</tr>
<tr>
<td></td>
<td>mBART Multilingual + E → S BT</td>
<td>50.0</td>
</tr>
<tr>
<td></td>
<td>mBART Multilingual B2BT</td>
<td>50.4</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Fine-tuning Dataset for Final Model</th>
<th>ST-Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>B2BT ($\mathcal{M}'$)</td>
<td>50.2</td>
</tr>
<tr>
<td>$\mathcal{M}$ + synthetic data from Gupta et al. (2021)</td>
<td>45.3</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Metric</th>
<th>ST-Test</th>
<th>mBertAln</th>
<th>B2BT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human eval rating</td>
<td>-</td>
<td>3.74</td>
<td>4.27</td>
</tr>
<tr>
<td>Human eval win %</td>
<td>-</td>
<td>17%</td>
<td>39%</td>
</tr>
<tr>
<td>Code-Mixing Index</td>
<td>28.3</td>
<td>20.7</td>
<td>27.2</td>
</tr>
<tr>
<td>Common En tokens</td>
<td>0.16</td>
<td>0.20</td>
<td>0.18</td>
</tr>
<tr>
<td>Code switch probability</td>
<td>0.27</td>
<td>0.24</td>
<td>0.27</td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th>Lang Pair</th>
<th>Fine-tuning Approach</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>HiEn→En</td>
<td>Un-masked</td>
<td>50.1</td>
</tr>
<tr>
<td></td>
<td>Masked</td>
<td>50.2</td>
</tr>
<tr>
<td>BnEn→En</td>
<td>Un-masked</td>
<td>42.8</td>
</tr>
<tr>
<td></td>
<td>Masked</td>
<td>44.2</td>
</tr>
<tr>
<td>MrEn→En</td>
<td>Un-masked</td>
<td>40.6</td>
</tr>
<tr>
<td></td>
<td>Masked</td>
<td>41.2</td>
</tr>
</tbody>
</table>

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


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 se-
quence modeling. In Proceedings of the 2019 Con-
ference of the North American Chapter of the Associa-
tion 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 Pro-
ceedings of the 56th Annual Meeting of the Asso-
ciation for Computational Linguistics (Volume 1:
Long Papers), pages 1543–1553, Melbourne, Aus-
tralia. Association for Computational Linguistics.

Gowtham Ramesh, Sumanth Doddapaneni, Aravinth
Bheemaraj, Mayank Jobanputra, Raghavan AK,
Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Ma-
halakshmi J, Divyanshu Kakwani, Navneet Ku-
mar, Aswin Pradeep, Kumar Deepak, Vivek Raga-
van, Anoop Kunchukuttan, Pratyush Kumar, and
Mitesh Shantadevi Khapra. 2021. Samantar: The
largest publicly available parallel corpora collection

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 An-
nual Meeting of the Association for Computational
Linguistics (Volume 1: Long Papers), pages 86–96,
Berlin, Germany. Association for Computational Lin-
guistics.

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

Kai Song, Yue Zhang, Heng Yu, Weihua Luo, Kun
enhancing NMT with pre-specified translation.
In Proceedings of the 2019 Conference of the North
American Chapter of the Association for Computa-
tional Linguistics: Human Language Technologies,
Volume 1 (Long and Short Papers), pages 449–459,
Minneapolis, Minnesota. Association for Compu-
tational Linguistics.

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

Vivek Srivastava and Mayank Singh. 2020. PHINC:
A parallel Hinglish social media code-mixed cor-
pus 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 Pro-
ceedings of the 59th Annual Meeting of the Associa-
tion for Computational Linguistics and the 11th Inter-
national Joint Conference on Natural Language Pro-
cessing (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, L ukasz
Kaiser, and Illia Polosukhin. 2017. Attention is all
you need. In Advances in Neural Information Pro-

Genta Indra Winata, Andrea Madotto, Chien-Sheng Wu,
models using neural based synthetic data from par-
allel sentences. In Proceedings of the 23rd Confer-
ence on Computational Natural Language Learning
(ConLL), pages 271–280, Hong Kong, China. Asso-
ciation 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 trans-

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sen-
nrich. 2020. Improving massively multilingual neu-
ral machine translation and zero-shot translation. In
Proceedings of the 58th Annual Meeting of the Asso-
ciation for Computational Linguistics, pages 1628–
1639, Online. Association for Computational Linguis-
tics.
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 workshop (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 \((E_{MD}, E, S_{M})\)
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/


---

**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.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Source</th>
<th>Size</th>
<th>Avg. tokens/sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>HiEn→En</td>
<td>Test</td>
<td>30K</td>
<td>HiEn-14.46, En-13.09</td>
</tr>
<tr>
<td></td>
<td>(S, E)</td>
<td></td>
<td>Hi-15.47, En-14.47</td>
</tr>
<tr>
<td>C</td>
<td>ST CM</td>
<td>40K</td>
<td>14.99</td>
</tr>
<tr>
<td></td>
<td>En mono</td>
<td></td>
<td>12.59</td>
</tr>
<tr>
<td>E_{MD}</td>
<td>ST En</td>
<td>53K</td>
<td>12.59</td>
</tr>
<tr>
<td></td>
<td>S_{M}</td>
<td>2M</td>
<td>18.95</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BnEn→En</th>
<th>Test</th>
<th>25K</th>
<th>BnEn-11.32, En-13.31</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(S, E)</td>
<td></td>
<td>Bn-12.14, En-13.56</td>
</tr>
<tr>
<td>C</td>
<td>ST CM</td>
<td>31K</td>
<td>11.23</td>
</tr>
<tr>
<td></td>
<td>En mono</td>
<td></td>
<td>11.34</td>
</tr>
<tr>
<td>E_{MD}</td>
<td>ST En</td>
<td>57K</td>
<td>12.31</td>
</tr>
<tr>
<td></td>
<td>S_{M}</td>
<td>2M</td>
<td>21.15</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MrEn→En</th>
<th>Test</th>
<th>25K</th>
<th>MrEn-11.32, En-13.00</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(S, E)</td>
<td></td>
<td>Mr-10.86, En-12.43</td>
</tr>
<tr>
<td>C</td>
<td>ST CM</td>
<td>38K</td>
<td>11.14</td>
</tr>
<tr>
<td></td>
<td>En mono</td>
<td></td>
<td>12.58</td>
</tr>
<tr>
<td>E_{MD}</td>
<td>ST En</td>
<td>57K</td>
<td>12.58</td>
</tr>
<tr>
<td></td>
<td>S_{M}</td>
<td>2M</td>
<td>16.22</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EsEn→En</th>
<th>Test</th>
<th>6.5K</th>
<th>EsEn-19.72, En-UNK</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(S, E)</td>
<td></td>
<td>Es-33.32, En-29.74</td>
</tr>
<tr>
<td>C</td>
<td>LinCE</td>
<td>15K</td>
<td>19.67</td>
</tr>
<tr>
<td></td>
<td>E_{MD}</td>
<td>LinCE</td>
<td>15.36</td>
</tr>
<tr>
<td></td>
<td>S_{M}</td>
<td>News Crawl</td>
<td>2M</td>
</tr>
<tr>
<td>E_{M}</td>
<td>News Crawl</td>
<td>2M</td>
<td>23.90</td>
</tr>
</tbody>
</table>
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 IndicNLP (Kunchukuttan, 2020) tokenizer for Indic language sentences and code-mixed sentences and Moses tokenizer 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.

---

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