Adapting Multilingual Models for Code-Mixed Translation using Back-to-Back Translation

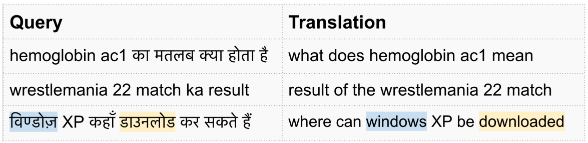
Anonymous ACL Rolling Review submission

Abstract

In this paper, we explore the problem of translating code-mixed sentences to an equivalent monolingual form. The scarcity of gold standard code-mixed to pure language parallel data makes it difficult to train a translation model that can perform this task reliably. Prior work has addressed the paucity of parallel data with data augmentation techniques. Such techniques rely heavily on external resources, which make the systems difficult to train and scale effectively for multiple languages. We present a simple yet highly effective training scheme for adapting multilingual models to the task of code-mixed translation. Our method eliminates the dependence on external resources by creating synthetic data from a novel two-stage back-translation approach that we propose. We show 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

Mixing words or phrases of a dominant language like English with another language is now a widespread phenomenon, causing user-generated content to be increasingly Code-Mixed (CM). Applications like search, recommendation, and advertisement that face the increasing prevalence of such code-mixed user queries can better match to the predominantly English content after translating the query to English. Such a translation step also facilitates greater reuse of existing high quality English NLP tools such as for query segmentation and entity linking. Figure 1 shows examples of code-mixed queries and their translations.



Query	Translation
hemoglobin ac1 का मतलब क्या होता है	what does hemoglobin ac1 mean
wrestlemania 22 match ka result	result of the wrestlemania 22 match
विण्डोज़ XP कहाँ डाउनलोड कर सकते हैं	where can windows XP be downloaded

Figure 1: Code-mixed queries in Hindi and their English translations. Highlighted source words are transliterations of words in the translation, highlighted in the same colour.

A major challenge for training code-mixed to English translation models is the lack of parallel data. Recent work on generating synthetic parallel data using available non-code-mixed parallel data require special-purpose models (Winata et al., 2018; Dhar et al., 2018; Tarunesh et al., 2021) and/or 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. Meanwhile the mainstream translation community is increasingly 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.

A multilingual model $\mathcal{M}$ that has been trained for *bidirectional* translation between a language $\mathcal{S}$ and English is already more capable than a unidirectional model of translating a code-mixed sentence. Additionally, when non-parallel code-mixed sentences are available, we could further inform $\mathcal{M}$ of CM sentences using the masked copy task (Liu et al., 2020). Surprisingly, despite having observed text in both languages and code-mixed text in the encoder, this multilingual model does not offer significant gains over a baseline unidirectional $\mathcal{S}$ to

English translation model.

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 our 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.

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. The simplicity of our training scheme belies its effectiveness. The complex mBERT based method for synthetic data creation in Gupta et al. (2021) improves over a simple monolingual translation model by +2.5 in BLEU for code-mixed Hindi to English translation. Our method improves over the mBERT method by +3.8 in BLEU and +6.3 over the simple baseline.

Our main contributions are as follows:

1. We present a novel training scheme for adapting multilingual models to the task of code-mixed translation. The simplicity of this scheme is in contrast to existing methods which use specialized architectures and external tools. Our approach complements mBART (Liu et al., 2020) which is a popular NMT pre-training strategy, and integrates easily within existing frameworks of multilingual translation.

2. We evaluate our method on four code-mixed datasets (Hindi, Spanish, Bengali, and Marathi), and obtain significant gains in BLEU over existing methods and baselines.

3. We conduct a human evaluation to establish that our method generates higher quality synthetic data for training than the best existing method.

4. One of the datasets (Marathi) is introduced by us and will be released in the public domain.

2 Related Work

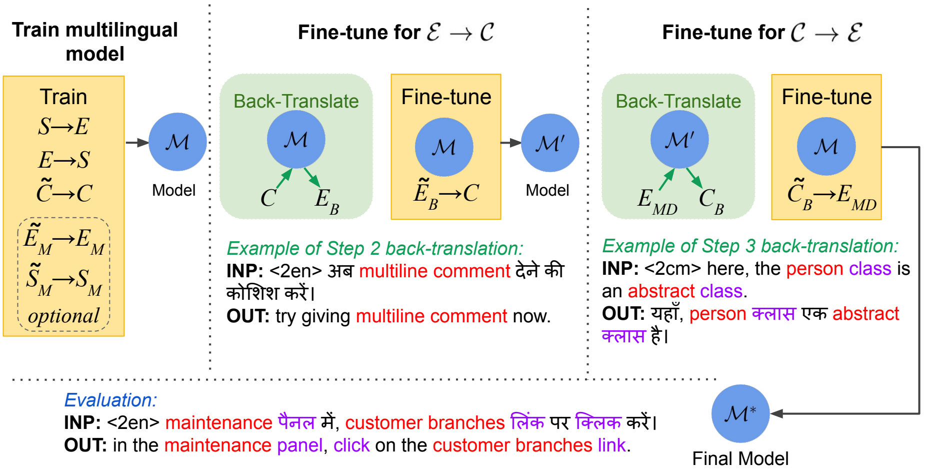
Code-mixing is receiving increasing interest in the Natural Language Processing (NLP) research community, with several efforts underway to improve model performance on code-mixed text for a wide variety of tasks Khanuja et al. (2020); Diab et al. (2014); Aguilar et al. (2018); Solorio et al. (2021); Song et al. (2019a).

Code-Mixed Language Models A primary focus area is training language models for code-switched data in the context of applications like speech recognition (Winata et al., 2019; Gonen and Goldberg, 2019). A major challenge addressed in this setting is lack of code-mixed data for training the language model. 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 code-mixed data which can be used for augmenting training data in language models. Tarunesh et al. (2021) propose a method for generating code-switched text from sentences in the matrix language through extensions of a translation model. None of these work generate code-mixed to English parallel data, which is our focus.

Code-Mixed Translation Translation of code-mixed sentences is a relatively unexplored task. The biggest challenge is the lack of large parallel training data. Srivastava and Singh (2020) release a small parallel dataset of code-mixed social media posts. Gupta et al. (2021) present a method for training translation models with synthetic parallel data created by learning code-switching patterns with an mBERT model and perturbing aligned monolingual parallel data. A major drawback of this approach is the reliance on external models for Language Identification, alignment, transliteration, and back-translation, which make for a complex and brittle training pipeline and increase difficultly in scaling to more languages. The CALCS 2021 workshop (Solorio et al., 2021) also released a shared task for code-mixed translation. So far the only submissions are straight-forward application of BART multilingual models, with which we also compare our method.

3 Our Approach

Our objective is to train a model that can translate code-mixed input, which contains words from English and an additional language $\mathcal{S}$, to monolingual English. 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. Let $\mathcal{S}, \mathcal{C}, \mathcal{E}$ denote the space of matrix language



Example of Step 2 back-translation:
INP: <2en> अब multiline comment देने की कोशिश करें।
OUT: try giving multiline comment now.

Example of Step 3 back-translation:
INP: <2cm> here, the person class is an abstract class.
OUT: यहाँ, person क्लास एक abstract क्लास है।

Evaluation:
INP: <2en> maintenance पैनल में, customer branches लिंक पर क्लिक करें।
OUT: in the maintenance panel, click on the customer branches link.

Figure 2: B2BT training pipeline, showing the two-stage back-translation based adaptation of an initial multilingual model. $(\tilde{\cdot})$ indicates source side masking during training.

sentences, code-mixed sentences, and English sentences respectively. For training our model we assume the presence of a parallel matrix language to English corpus $(S, E) \subset (\mathcal{S}, \mathcal{E})$ and a non-parallel code-mixed corpus $C \subset \mathcal{C}$. Since code-mixing data often appear in application domains like social media, which differ from formal domains like news in which parallel data (S, E) is available, we additionally use a domain-specific monolingual English corpora $E_{MD} \subset \mathcal{E}$. Optionally, we can also exploit monolingual data in each of the source $S_M \subset \mathcal{S}$, and target languages $E_M \subset \mathcal{E}$.

Our starting point is a multilingual model $\mathcal{M}$ that is exposed to sentences from $\mathcal{S}, \mathcal{C}, \mathcal{E}$ in both the encoder and decoder. We achieve this by training for bidirectional translation using the parallel data and masked copying using the non-parallel corpus. Pre-trained models like mBART can also be used as starting point for this step. We elaborate in Section 3.1. We adapt this model for translation from code-mixed $\mathcal{S}$ to English $\mathcal{E}$ using a two stage back-to-back-translation approach which we call B2BT. In the first stage we use BT on $\mathcal{C}$ to teach the model to translate English sentences to code-mixed (Section 3.2). In the second stage we use BT on E_{MD} to achieve the target of code-mixed to English translation (Section 3.3). Figure 2 presents an overview of our training process.

3.1 Training Base Multilingual Model

The multilingual model $(\mathcal{M})$ is trained to translate between the constituent languages in both directions, and denoise code-mixed sentences. The multilingual model uses special prefix tokens to indicate the desired output language to the model. As in Johnson et al. (2017) we prefix source sentences with one of three special tokens (1) <2en> when translating to English (2) <2xx> when translating to the matrix language, where 'xx' denotes the code[1] for the matrix language, and (3) <2cm> when translating to code-mixed.

Training data in this step consists of parallel matrix language to English corpus (S, E) and non-parallel data in English E_M, matrix language S_M, and code-mixed C. For the non-parallel corpora, we train the model to copy the source to the target by masking out tokens in the source as used in (Song et al., 2019b). Our masking strategy comprises of randomly and independently replacing source tokens with a special token <M> with a masking probability of 0.2. The decoder is still expected to produce the complete output and not just the masked spans.

While the model is capable of translating code-mixed sentences to English, our evaluation found no significant gains beyond a simple translation model. This is possibly because the <2en> model has not learned to copy English tokens from the code-mixed input. This necessitates training the model with synthetic parallel data with code-mixed source. If we use the current $\mathcal{M}$ to back-translate English to synthetic CM, we also get poor accuracy as we will show in Section 6.2. This led us to

[1] This refers to 2-letter ISO 639-1 language codes

design a two stage back-translation approach that we describe next.

3.2 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. Note that initial $\mathcal{M}$ is poor at translation $\mathcal{E} \rightarrow \mathcal{C}$ translation since it was only trained to copy over code-mixed sentences.

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 $\mathcal{M}'$.

3.3 Fine-tune for $\mathcal{C} \rightarrow \mathcal{E}$

In the final step of our training process we aim to realise our objective of training to translate code-mixed sentences to English. We start by back-translating the in-domain monolingual English corpus E_{MD} to code-mixed sentences C_B using $\mathcal{M}'$. This is done by prefixing sentences from the English corpus 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.

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. An ablation study in Section 6.5 shows that such masking boosts accuracy of code-mixed translation over and above monolingual masking.

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 1: 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.

The entire training pipeline is summarised in Figure 2.

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), Bengali (BnEn→En) used in Gupta et al. (2021), Spanish (EsEn→En) on the LinCE leaderboard [2], and a new Marathi (MrEn→En) dataset that we introduce. Table 1 presents a summary of the various corpus sizes used for each of the datasets.

4.1 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)

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

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

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.

4.2 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.

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 [5] 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.

5 Results

We compare our method, B2BT against the mBERT model from Gupta et al. (2021) and other baselines. For BnEn→En, we re-train the mBERT based approach with the newly released Samanantar data to create a stronger baseline than what they report. The paper does not present results for

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

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

MrEn→En, so we also train an mBERT model ourselves for this pair. The first pair of baselines we compare against are the base bi-lingual $\mathcal{S} \rightarrow \mathcal{E}$ model and the version fine-tuned with back-translated domain data E_{MD}. Similarly, we also compare against the base multilingual model $\mathcal{M}$, and $\mathcal{M}$ fine-tuned with back-translated domain data. Back-translations for these baselines are obtained from bi-lingual $\mathcal{E} \rightarrow \mathcal{S}$ models.

5.1 Training From Scratch

Table 2 presents results comparing B2BT approach against the baselines and mBERT on HiEn→En, BnEn→En, and MrEn→En from Spoken Tutorial. In these experiments, B2BT is trained on standalone multilingual models. We can see B2BT significantly outperforms the mBERT based approach across all language pairs. We outperform this state of the art mBERT approach by +3.8 BLEU points on HiEn→En, +2.8 BLEU points on BnEn→En, and +0.6 BLEU points on MrEn→En. We also see substantial improvements on the two adversarial subsets of *ST-Test* introduced in Gupta et al. (2021). The *ST-OOV* dataset contains sentences with at least two words which are not present in the parallel training data. The *ST-Hard* dataset contains 2,000 sentences which had the lowest BLEU score when translating with the unidirectional $\mathcal{S} \rightarrow \mathcal{E}$ translation models.

Further, our model also significantly outperforms the multilingual model adapted with the simple back-translated (BT) method. For HiEn→En we get +6.2 BLEU increase, for BnEn→En +2.5 BLEU increase, while for MrEn→En we are at par. This establishes the importance of our two-stage back-translation approach to adapting multilingual models for code-mixed translation.

5.2 Fine-tuning mBART

Our approach can complement existing multilingual pre-trained models such as mBART. In these experiments the base multilingual model $\mathcal{M}$ is trained by fine-tuning an mBART checkpoint. Table 3 presents these results. We see a large improvement of +12.9 BLEU points for HiEn→En. For EsEn→En, we beat the baseline on the leaderboard by +6.5 BLEU points, and also see a signficant improvement of +1.1 BLEU points on the multilingual model. Here again we observe gains beyond simple BT-based fine-tuning of the multilingual model. We get a +4.6 BLEU increase for HiEn.

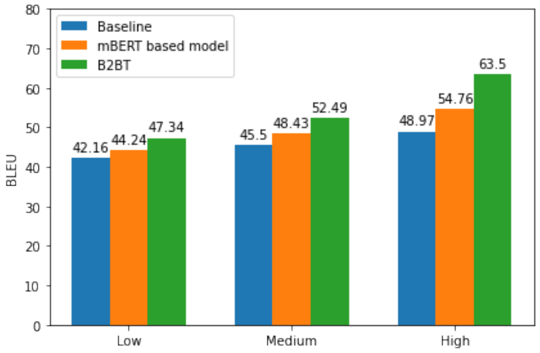


Figure 3: Improvements in BLEU with B2BT against the mBERT based model and the domain-adapted bilingual model baseline across three splits of the test set with varying degree of code-mixing in the source.

6 Analysis

Since references are private for the LinCE Benchmark, we conduct all further analysis of our results on the Spoken Tutorial datasets. We study the HiEn→En task in our analysis as a representative, however for some experiments we also analyze models for other language pairs.

6.1 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.

6.2 Ablation: Role of two-stage BT

To examine the relative importance of different components in our training pipeline, we compare our final code-mixed Hi→En model ($\mathcal{M}^*$) against a setup where we do a one-step BT using $\mathcal{M}$. In Row 2, $\mathcal{M}$ is fine-tuned on in-domain English E_{MD} data back-translated to Hi with $\mathcal{M}$. We observe a large gap compared to B2BT. This indicates the importance of training with parallel data containing code-mixed source. In Row 3 we show the model 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 ($\mathcal{M}$) is only trained to do span-level denoising on code-mixed

Lang Pair	Method	ST-Test	ST-OOV	ST-Hard
HiEn→En	Hi→En Model	36.9	33.9	2.1
	Hi→En Model + simple BT E_{MD}	43.9	41.4	18.6
	mBERT (Gupta et al., 2021)	46.4	44.6	23.4
	Multilingual	38.0	37.7	17.5
	Multilingual + simple BT E_{MD}	44.0	40.9	22.6
	B2BT	**50.2**	**49.9**	**30.7**
BnEn→En	Bn→En Model	30.8	31.1	14.1
	Bn→En Model + simple BT E_{MD}	40.9	41.2	21.2
	mBERT (Gupta et al., 2021)	37.4	37.3	17.8
	mBERT (our implementation)	41.4	41.9	22.3
	Multilingual	30.9	31.4	13.8
	Multilingual + simple BT E_{MD}	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 + simpleBT E_{MD}	39.3	39.2	16.5
	mBERT (our implementation)	40.6	40.5	17.8
	Multilingual	29.1	29.7	9.0
	Multilingual + simple BT E_{MD}	**41.4**	**41.5**	**18.9**
	B2BT	**41.2**	**41.3**	**18.7**

Table 2: Comparing BLEU scores on the Spoken Tutorial test set for B2BT trained from scratch against the mBERT model and bilingual and multilingual baselines.

Lang Pair	Method	BLEU
HiEn→En	mBART Multilingual	35.1
	mBART Multilingual + BT	43.4
	mBART Multilingual B2BT	**48.0**
EsEn→En	mBART (leaderboard)	43.9
	mBART Multilingual	49.3
	mBART Multilingual + BT	**50.0**
	mBART Multilingual B2BT	**50.4**

Table 3: Results comparing B2BT fine-tuned on an mBART checkpoint against baselines and best existing models on the LinCE leaderboard.

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

6.3 Comparing with Other Synthetic Code-mixed Data

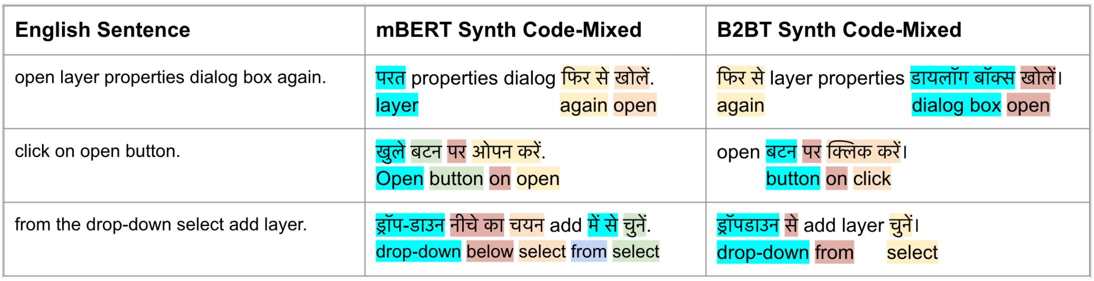
We hypothesize that the reason our model performs substantially better is that the synthetic data generated by our model is of higher quality. To test this hypothesis we replace the synthetic code-mixed parallel data of B2BT with synthetic data from mBERT (Gupta et al., 2021), and VACS (Samanta et al., 2019) while keeping the rest of the training of $\mathcal{M}^*$ unchanged. Table 4 (row 4-5) presents this result. The improvement of almost +4.9 BLEU points on ST-Test over using mBERT data, clearly shows that the synthetic data from our model has better quality. Figure 4 presents examples of synthetic sentences generated by B2BT vs mBERT. The mBERT method has word omissions like “box" in row 1 which could be caused by poor back-translation, or repetition of “open" in row 2, which could be a combination of back-translation and alignment mistakes. Due to reliance on multiple external tools, we found the mBERT synthetic data to be highly noisy.

6.4 Human Evaluation and Code-mixing Stats for Generated Code-mixed Data

We ask human annotators to rate the the synthetic code-mixed text for fluency and intent preservation when presented as translations for the English text they were created from. Raters are asked to evaluate quality of source-target pairs (similar to Wu et al. (2016)) on a scale of 0 to 6. A score of 0 indicates a completely irrelevant translation, and a score of 6 indicates a translation that is fluent and captures intent perfectly. 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 the mBERT model. In 39% of examples our model is

#	Fine-tuning Dataset for Final Model	ST-Test	ST-OOV	ST-Hard
1	B2BT ($\mathcal{M}^*$) - $\mathcal{M}$ fine-tuned with sampled $\mathcal{E} \to \mathcal{C}$ from $\mathcal{M}'$	50.2	49.9	30.7
2	$\mathcal{M}$ fine-tuned with sampled $\mathcal{E} \to \mathcal{S}$ from $\mathcal{M}$	46.1	45.2	26.6
3	$\mathcal{M}$ fine-tuned with sampled $\mathcal{E} \to \mathcal{C}$ from $\mathcal{M}$	35.7	35.8	20.6
4	$\mathcal{M}$ fine-tuned with synthetic data from mBERT (Gupta et al., 2021)	45.3	43.1	24.1
5	$\mathcal{M}$ fine-tuned with synthetic data from VACS (Samanta et al., 2019)	44.0	41.5	23.6

Table 4: Comparing translation accuracy (BLEU) on the HiEn→En task when using synthetic code-mixed data generated from $\mathcal{M}'$ in our approach against (1) synthetic data sampled from $\mathcal{M}$ in our approach (2) synthetic data from other methods like mBERT and VACS.



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
from the drop-down select add layer.	ड्रॉप-डाउन नीचे का चयन add में से चुनें. drop-down below select from select	ड्रॉपडाउन से add layer चुनें। drop-down from select

Figure 4: Examples of Spoken Tutorial synthetic sentences generated from mBERT vs B2BT. English translations of Devanagari words in the code-mixed sentences are also provided with highlights.

Metric	ST-Test	mBERT	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 5: Comparing the synthetic data generated through mBERT against B2BT.

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

rated higher than the mBERT model, 45% of examples get the same score, and only in 17% examples is mBERT better (Table 5).

We compare code-mixing statistics between the synthetic data generated by B2BT and mBERT on ST-Test in Table 5. We find that the data generated from B2BT is closer to the test data distribution 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.

6.5 Effect of Source Side Masking

Finally, we evaluate the impact of source side masking in B2BT's fine-tuning stages. Table 6 compares model performance with and without source side masking when fine-tuning. We observe noticeable gains, with the highest for BnEn at +1.5.

7 Conclusion

We present a simple two-stage back-translation approach (B2BT) for adapting multilingual models for code-switched translation. We demonstrate remarkable improvements on four datasets compared to recent state of the art methods, and default back-translation baselines. Detailed ablation studies and contrast with alternative methods of generating synthetic code-mixed data underline the significance of our two-stage approach. Through human evaluation, we find that B2BT's synthetic data is objectively higher quality than the one used by existing work. Most importantly, we remove the dependence on external resources like models for language identification, alignment, transliteration, and back-translation in creating this synthetic data. The straightforward two step back-translation approach reduces code-complexity which is highly desirable in models to be used in production. Finally, our approach naturally fits with existing multilingual translation frameworks, which are crucial in expanding coverage to multiple languages without building per-language pair models.

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*.

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.

Mrinal Dhar, Vaibhav Kumar, and Manish Shrivastava. 2018. Enabling code-mixed translation: Parallel corpus creation and MT augmentation approach. In *Proceedings of the First Workshop on Linguistic Resources for Natural Language Processing*, pages 131–140, Santa Fe, New Mexico, USA. 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.

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).

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

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

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. 2018. Learn to code-switch: Data augmentation using copy mechanism on language modeling. *CoRR*, abs/1810.10254.

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, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv: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.