Domain Curricula for Code-Switched MT at MixMT 2022

Lekan Raheem & Maab Elrashid
African Institute for Mathematical Sciences (AIMS)
{rwaliyu,mnimir}@aimsammi.org

Abstract

In multilingual colloquial settings, it is a habitual occurrence to compose expressions of text or speech containing tokens or phrases of different languages, a phenomenon popularly known as code-switching or code-mixing (CMX). We present our approach and results for the Code-mixed Machine Translation (MixMT) shared task at WMT 2022: the task consists of two subtasks, monolingual to code-mixed machine translation (Subtask-1) and code-mixed to monolingual machine translation (Subtask-2). Most non-synthetic code-mixed data are from social media but gathering a significant amount of this kind of data would be laborious and this form of data has more writing variation than other domains, so for both subtasks, we experimented with data schedules for out-of-domain data. We jointly learn multiple domains of text by pretraining and fine-tuning, combined with a sentence alignment objective. We found that switching between domains caused improved performance in the domains seen earliest during training, but depleted the performance on the remaining domains. A continuous training run with strategically dispensed data of different domains showed a significantly improved performance over fine-tuning.

1 Introduction

Code-mixing (CMX) denotes the alternation of two languages within a single utterance (Poplack, 1980; Sitaram et al., 2019). Code-mixing occurs mostly in unofficial groups in multilingual environments. More than 77% of Asians are multilingual (Ramakrishnan and Ahmad, 2014), and other statistics estimate that 64.5% of Europeans speak more than two languages, with more than 80% of adults in the region being bilingual (Eurostat, 2019). Code-mixing happens far more often in conversations than in writing, and mostly in unofficial settings, hence it rarely occurs in documented settings. This makes substantial data gathering for computational approaches to translations of code-mixed language difficult. Parallel corpora for code-switched data is very scarce (Menacer et al., 2019), this is because code-mixing mostly occurs in unofficial conversations like social media interactions.

Contemporary Neural Machine Translation (NMT) mostly makes use of parametric sequence-to-sequence models (Bahdanau et al., 2014; Vaswani et al., 2017), where an encoder receives a source sentence and outputs a set of hidden states, the decoder then scrutinizes these hidden states at each step, and outputs a sequence of softmax distribution over the target vocabulary space. Considering that we would need vast quantities of data to train an adequate NMT for this task, we leverage large-scale synthetic and available small data and notably rank data on domain relevance, by fine-tuning with it, initiating training with the relevant domain and strategically placing it at the premier batches of the training data.

Essentially, the characteristics of the data an NMT model is trained on are paramount to its translation quality, in particular in terms of size and domain. It is quintessential to train NMT models based on the domain relevance of corpora. Since most code-mixing occurs in unofficial communication, it is costly to find a lot of labeled data for every domain we are interested in. Hence we attempt to find less expensive exigencies to supplement training data, pretrain on largely available data of different domains, strategically construct synthetic data, and apportion data to make up for missing domains.

In these WMT Subtasks – monolingual to code-mixed machine translation (Subtask-1) and code-mixed to monolingual machine translation (Subtask-2), we also fine-tune on different domains, align representations of data and find the best combination of approaches to solving the subtasks. The main intuition behind our proposed solution is that NMT models exhibit a significant translation correlation when trained on data from the same or similar domains. With different data domain re-

quirements, it performs better when trained with data of the most relevant domain as preliminary batches compared to finetuning. As most natural code-mixed data source is social media and it is difficult to gather a good amount to train a model, it is incumbent to find a strategy that makes the model prioritize this form of data above others. Accordingly, we attempt to find less expensive techniques to supplement training data, pretrain on largely available data of different domains, strategically construct synthetic data, and apportion data to make up for missing domains. Our result showed improved performance on innate code-mixed data (and non-synthetic WMT test set samples) when this was prioritized and performed strongly in a test with a mix of several other data sources. We observed a better performance with domain-specific evaluation upon finetuning but this intensely plummeted performance on other 'pretraining domains', and more balanced performance on passing the interesting domain in the preliminary batches in a single 'all domain' training.

2 Related Work

It is laborious to obtain 'one-fits-all' training data for NMT. Most publicly available parallel corpora like Tanzil, OPUS, UNPC are sourced from documented communication, and these are often domain-specific. In NMT, data selection e.g. Axelrod et al. (2011) has remained as an underlying and important research concern. Choosing training examples that are relevant to the target domain, or by choosing high-quality examples for data cleaning (also known as denoising), has been essential in domain adaptation. Building a large-scale multi-domain NMT model that excels on several domains simultaneously becomes both technically difficult and practically back-breaking. Addressing research problems such as catastrophic forgetting (Goodfellow et al., 2013), data balancing (Wang et al., 2020), Adapters (Houlsby et al., 2019) have shown improvement. Unfortunately, several domains are difficult to handle with the single-domain data selection techniques currently in use. For instance, improving translation quality of one domain will often hurt that of another (Britz et al., 2017; van der Wees et al., 2017).

Song et al. (2019) replaced phrases with pre-specified translation to perform "soft" constraint decoding. Xu and Yvon (2021) generated code-mixed data from regular parallel texts and showed

this training strategy yields MT systems that surpass multilingual systems for code-mixed texts.

Considering that code-mixed text belongs in less documented domains than most, there may be a need for domain adaptation used on sufficiently available data domains. Our work is inspired by the following approaches: Wang et al. (2019) executed simultaneous data selection across several domains by gradually focusing on multi-domain relevant and noise-reduced data batches while carefully introducing instance-level domain-relevance features and automatically constructing a training curriculum. Park et al. (2022) demonstrated that instance-level features are better able to distinguish between different domains compared to corpus-level attributes. Dou et al. (2019) proposed modeling the difference between domains instead of smoothing over domains for machine translation.

Anwar et al. (2022) showed that an encoder alignment objective is beneficial for code-mixed translation, in addition to Arivazhagan et al. (2019) that proposed auxiliary losses on the NMT encoder that imposed representational invariance across languages for multilingual machine translation.

English	Code-Mixed (CMX)
@dh*v*l2410*6 sure brother :)	@dh*v*l2410*6 sure bhai :)
"I just need reviews like these, this motivates me a lot"	"Bas aise hi reviews ki zaroorat hai, kaafi protsahan milta hai in baaton se. "
When the sorrow got missing in this room, the blood also became thin, #GuessTheSong	Jab gam ye rum mein kho gaya, toh khoon bhi patla hogaya #GuessTheSong

Table 1: Examples from the WMT Shared Task Dataset.

3 Data

In table 1 we show some samples from the WMT shared task, sourced from the non-synthetic validation data. The data provided for Subtask-1 (Srivastava and Singh, 2021) contains synthetic and human-generated data and Subtask-2 Parallel Hinglish Social Media Code-Mixed Corpus (Srivastava and Singh, 2020) for both tasks are mostly unofficial, mostly short conversational sentences, with some letters asterisked for privacy/derogatory reasons.

Since we need to augment provided data for a reasonable quantity to train a NMT model, we generated synthetic code-mixed data from the IITB

English	Code-Mixed (CMX)
Overhead charge is a percentage of the direct costs of providing the services under the contract.	Overhead charge, anubandh ke anusaar pradatt sevaon kee pratyaksh laagat ka ek pratishat hota hai.
A strategy of ignoring potential problems on the basis that they may be exceedingly rare.	us aadhaar par sambhaavit problems ko anadekha karane kee ek yukti, jahaan ki ve ati dushpraapy ho sakate hain.
A standard of measurement, or a unit that can be studied separately / independently.	koee maapadand athava koee a unit that svatantr roop se/alag se adhyayan kiya ja sakata ho.

Table 2: Examples from IITB Corpus.

corpus (Kunchukuttan et al., 2017) which is from 17 sources of different domain mostly HindEnCorp (Bojar et al., 2014), Gyaan-Nidhi Corpus (Garg et al., 2018), Indian Government corpora - CFILT, Mahashabdkosh, Tanzil, and GNOME (Kunchukuttan et al., 2017) (details in section 3.1). Synthetic code-switched sentences generated from the IITB corpus belong to a different domain than the WMT evaluation data, as we illustrate with the English translation samples in table 2.

For the pretraining-finetuning setup, we pretrain with synthetic code-switched data generated from IITB corpus and fine-tune on the WMT data provided for each task. For both pretraining and fine-tuning, we coordinate the data similar to (Anwar et al., 2022) – For Subtask-1, Monolingual to code-mixed machine translation subtask, we use the Hindi sentence (Devanagari script) as source sequence and the corresponding code-switched sentence (Roman script) as target, then alternated the English sequence (Roman script) as source sentence and the same corresponding code-switched sentence as the target sequence. The above two source-target parallel data are set after each other. For Subtask-2, Code-mixed to monolingual machine translation subtask, we have a similar arrangement as in Subtask-1, but with the source sequences of Hindi and code-mixing (Hinglish) in Roman script and as the target the corresponding English sequence. We removed sequences shorter than 2 tokens, and those longer than 250 tokens, and a target-to-source token ratio of more than 1.5. After cleaning the pretraining data, for Subtask-1, we have about 2.5M parallel sentences and 2.3M parallel sentences for Subtask-2.

For the finetuning process, we made use of the WMT training data provided for each subtask and organized like the pretraining data as described above. After cleaning, for Subtask-1 (Synthetic + Human-generated), we have a total of over 11K parallel sentences. For subtask-2, over 12K parallel sentences remain after cleaning.

Since the IITB corpus encompasses multiple sources and domains where code-mixing infrequently occurs, we decided to configure our model in a way it first learns from natural code-mixed data provided by WMT. We experiment with a hand-designed curriculum of the Synthetic Code-switched data generated from the IITB corpus and the WMT provided data. We supply the model the non-synthetic WMT data only in the first few batches in the hope that this would faintly familiarize the model with domain-specific features before it learns from the synthetic code-switched data we generated from other domains. We compare the results of this approach to the above described pretraining-finetuning setup. All data is tokenized and normalized using sentencepiece[1].

3.1 Code Switched Data Generation

Given that most publicly available corpora are monolingual, it is requisite to generate sufficient synthetic code-mixed data for training. Moreover, there have been works on generating synthetic code-mixed data linguistically, there are a few rules theories that are essential.

The **Equivalence Constraint Theory** states that intra-sentential code-mixing can only happen where the surface structures of two languages map onto each other, implicitly following both languages' grammatical norms (Poplack, 1980). Fundamentally, we can only attempt code-mixing at points where both languages coincide on the parse tree to equivalent phrase structure.

The **Matrix Language Theory** explains code-mixing by introducing the concept of a "Matrix Language," or base language, into which clusters of the "Embedded Language," or second language, are introduced in such a way that the former sets the grammatical structure of the sentence and the latter "switches-in" at grammatically correct points of the sentence (Myers-Scotton, 2001). The Matrix language has more tokens in the sequence and its rules are designated above the embedded language's.

Considering the linguistic theories above, we

---

[1]https://github.com/google/sentencepiece

generate code-mixed data by locating where both languages coincide based on a word-level alignment extracted and only replace tokens based on the "matrix language theory". Roughly following the recipes by (Song et al., 2019; Rizvi et al., 2021; Xu and Yvon, 2021; Anwar et al., 2022), we generate synthetic code-switched data from the IITB parallel data: We create code-mixed data by first transliterating Hindi (Devanagari script) to Roman script using Ritwik's tool[2], then extract word alignments using the giza++ toolkit (Och and Ney, 2003), and extract minimal alignment units following the approach of (Crego, 2005). We choose Hindi as the "matrix language" by determining this from the provided WMT training data, we extract word alignments and find how many tokens in each sequence belongs to which language using the language detector of Googletrans python library[3] and assign the language with more tokens as the matrix language. Figure 1 shows the Hindi/English matrix language ratio for both subtasks.

Similar to MLM pre-training used by BERT (Devlin et al., 2018), we randomly replace 15% of the tokens in each Hindi sentence with their aligned segments in the embedded language (English). For short sequences with less than 7 tokens we make only one replacement, chosen uniformly at random.

4 Training Objective

Considering the effectiveness of clean finetuning (Wu et al., 2019), and pre-training (Mathis et al., 2019), we attempt a combined pipeline of pretraining+finetuning experiment and also a single training but with tactical positioning of the most important domain. In the finetuning process and training with specially ordered data, as recommended by (Anwar et al., 2022), we add an alignment loss to the encoder to encourage source and target representations to be close in representation space minimizing the max-pooled cosine distance of the encoder representation as shown in equation 1:

$$\Omega = \mathbb{E}_{D_{(en,hi)}}[1 - sim(Enc(x_{src}), Enc(x_{tgt}))] \tag{1}$$

Where Ω is the encoder loss, $D_{(en,hi)}$ is the data consisting of the parallel pairs of code-mixed to

monolingual or monolingual to code mixed depending on which subtask the data belongs to, x_{src} is the source sequence and x_{tgt} is the target sequence, Enc(x) is the max-pooled encoder representation of sentence x similar to (Gouws et al., 2014) and (Coulmance et al., 2015), and sim is the cosine similarity. Unlike (Arivazhagan et al., 2019) where the whole model's parameters are updated as shown in figure 2.

5 Experiments and Results

In all of our experiments, we used Transformer-Base (Vaswani et al., 2017) configuration with the Fairseq (Ott et al., 2019) framework. All models were trained on four Tesla T400 GPUs using IITB and WMT-'22 MixMT data for training as described in Section 3, with a shared vocabulary of 77K BPE (Sennrich et al., 2015) sub-words to create a joint vocabulary for both tasks and all models. The model's hyperparameters can be found in Appendix A.

5.1 Results

Based on the human evaluation by the organizers of the subtasks, the translation result of our initial models - v0.2 submitted – which was trained with mixing the IITB with the WMT without prioritizing the target domain – had an overall rating of 1.75 from 10 random translations for each subtask, this ranked inferior to many other submissions.

With the help of native Hindi speakers to investigate our data, we found some of the causes it performed decumbent, which were as a result of some of the different data preprocessing tools we used: For Transliteration, We tried a few devanagari to roman tools but had some shortcomings like:

- Lipika-ime[4]: inappropriate handling of diacritic characters.

- Indic-trans[5]: Removal of vowels (e.g. default -> difolt, highlight -> hilite, method -> methd, etc..), Splitting of words that lead to suboptimal outputs (e.g. "un he" instead of "unhe").

- Sheental[6]: repetition of vowels e.g. jane -> jaane, yaar -> yaara, incorrect replacement of characters e.g. om -> on and occurence of
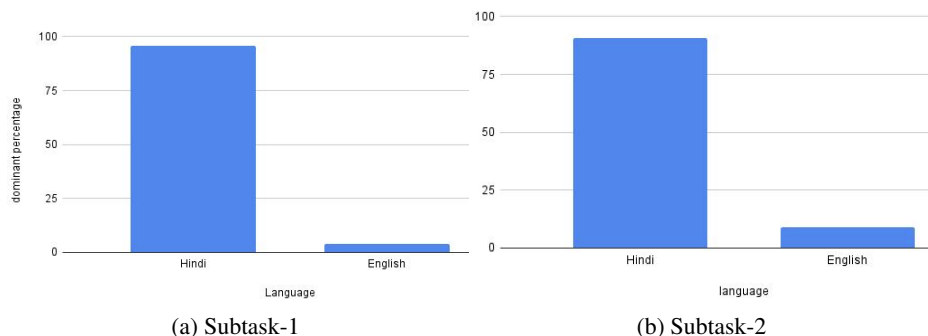
---

[2]https://github.com/ritwikmishra/devanagari-to-roman-script-transliteration

[3]https://github.com/ssut/py-googletrans/blob/master/docs/index.rst

[4]https://github.com/ratreya/lipika-ime

[5]https://github.com/libindic/indic-trans

[6]https://github.com/sheetalgiri/devanagari-to-roman-script

(a) Subtask-1   (b) Subtask-2

Figure 1: Percentage of Hindi vs. English as matrix language from WMT'22 Hinglish validation data for the subtasks.

Model	IITB Eval Set		WMT Eval Set		Mixed Eval Set	
	Subtask-1	Subtask-2	Subtask-1	Subtask-2	Subtask-1	Subtask-2
Pretrained (IITB corpus only)	**0.81**	**0.85**	0.41	0.47	**0.76**	**0.80**
Pretrained (IITB corpus) + Finetuned (WMT provided)	0.49	0.52	0.54	0.58	0.53	0.59
Mixed-data training (target domain first)	0.76	0.79	**0.62**	**0.64**	0.70	0.73

Table 3: Translation accuracy of subtask-1 and subtask-2 of Hindi-English in ROUGE-L (F1-Score) on different **test** data of different domains, based on models trained on different domain training data, data arrangement or training pipeline.
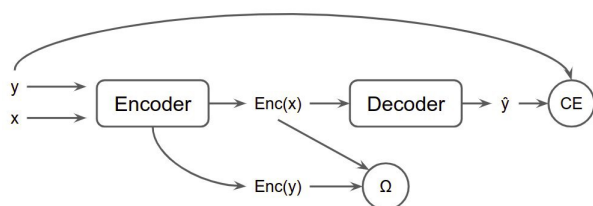

Figure 2: The loss function visualization, CE is the Cross Entropy, Ω is the encoder loss.

needless suffixes e.g. palat -> palata, some diacritic appeared independently.

- Ritwik's: inappropriately breaking very long sentences into multiple lines, replacing individually occurring tokens like um -> oon and abruptly stopping when ran over large amount of data so we divided the data into chunks each containing not more than 200K sequences, optimized by parallel computing using dask, and replaced the individually occurring tokens changed afterwards.

We also investigated our initial model and discovered a few other issues like:

- Cases of translation of proper nouns in Subtask-2 (e.g. Sapna -> dream) which we deduce as a pointer to insufficient training data.

- Imprecise tokenization and detokenization,

we also switched to use of Google sentence-piece instead of Moses SMT

- Also, the organizers noticed the team's output had an incorrect order. A problem where the post-processing had sorted the hypothesis and fragmented longer sentences also influenced the rating.

Upon inspecting our model outputs we found a few inaccuracies with the tools we used for transliteration and tokenization for the submitted model hypotheses. We fixed these, and present the results in the following section.

5.2 Post-Submission Results

Table 3 shows the experimental results based on different test data of samples each from IITB corpus, WMT, and a Mixed test sample evenly selected from Samanatar (includes IITB corpus, CCMatrix, Hindi-News, Jagran, Livehindustan, Patrika and WMT). We made use of other reputable tools to fix the aforementioned errors, added the domain curriculum technique, and ran the experiment again and present it in table 3.

Table 3 shows that fine-tuning on the WMT domain improves translation accuracy on this domain slightly, but the model suffers 'catastrophic forgetting' on domains it was initially trained on. Pre-training did not lead to a good generalization for

the WMT test samples provided, hence a need for domain adaptation. Placing the relevant domain in the preliminary batches for mixed-data training also improves training on such a domain but hurts other domains slightly.

6 Conclusion

We present a data domain sorting method that improves translation performance based on a target domain for the WMT 2022 code-switching shared tasks. We compared our result to a pretraining and fine-tuning pipeline, and demonstrated that the finetuning method improves on specified domain but upsets on previously learned data domain. An aspect we intend to delve further into is efficient domain adaptation strategies that may help low-resource domains such as code-mixing, and have little or no effect on high-resource domains, we are currently looking into domain adaptation learning curve (Park et al., 2022), extraction of domain-specific parameters (Dou et al., 2019) for better data augmentation strategies, better acquisition of code-mixed data, and the use of Adapters (Houlsby et al., 2019).

Acknowledgement

References

Mohamed Anwar, Lekan Raheem, Maab Elrasheed, Melvin Johnson, and Julia Kreutzer. 2022. True bilingual NMT. In *3rd Workshop on African Natural Language Processing*.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Roee Aharoni, Melvin Johnson, and Wolfgang Macherey. 2019. The missing ingredient in zero-shot neural machine translation. *CoRR*, abs/1903.07091.

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate.

Ondřej Bojar, Vojtěch Diatka, Pavel Rychlý, Pavel Straňák, Vít Suchomel, Aleš Tamchyna, and Daniel Zeman. 2014. HindEnCorp - Hindi-English and Hindi-only corpus for machine translation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3550–3555, Reykjavik, Iceland. European Language Resources Association (ELRA).

Denny Britz, Quoc Le, and Reid Pryzant. 2017. Effective domain mixing for neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 118–126, Copenhagen, Denmark. Association for Computational Linguistics.

Jocelyn Coulmance, Jean-Marc Marty, Guillaume Wenzek, and Amine Benhalloum. 2015. Trans-gram, fast cross-lingual word-embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1109–1113, Lisbon, Portugal. Association for Computational Linguistics.

Josep Crego. 2005. Reordered search and tuple unfolding for ngram-based smt.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Zi-Yi Dou, Xinyi Wang, Junjie Hu, and Graham Neubig. 2019. Domain differential adaptation for neural machine translation. *CoRR*, abs/1910.02555.

Eurostat. 2019. Translate foreign language skills statistics. https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Foreign_language_skills_statistics#Number_of_foreign_languages_known.

Kamal Garg, Ajit Kumar, and Vishal Goyal. 2018. Development of punjabi-english (puneng) parallel corpus for machine translation system. pages 690–693.

Ian J. Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. 2013. An empirical investigation of catastrophic forgetting in gradient-based neural networks.

Stephan Gouws, Yoshua Bengio, and Greg Corrado. 2014. Bilbowa: Fast bilingual distributed representations without word alignments.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Anoop Kunchukuttan, Pratik Mehta, and Pushpak Bhattacharyya. 2017. The IIT bombay english-hindi parallel corpus. *CoRR*, abs/1710.02855.

Alexander Mathis, Mert Yüksekgönül, Byron Rogers, Matthias Bethge, and Mackenzie W. Mathis. 2019. Pretraining boosts out-of-domain robustness for pose estimation. *CoRR*, abs/1909.11229.

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. 2001. The matrix language frame model: Development and responses. *Trends in Linguistics Studies and Monographs*, 126:23–58.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

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. *CoRR*, abs/1904.01038.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation.

Cheonbok Park, Hantae Kim, Ioan Calapodescu, Hyunchang Cho, and Vassilina Nikoulina. 2022. Dalc: Domain adaptation learning curve prediction for neural machine translation.

Shana Poplack. 1980. Sometimes i'll start a sentence in spanish y termino en espaÑol: toward a typology of code-switching 1. *Linguistics*, 18:581–618.

Karthick Ramakrishnan and Farah Z. Ahmad. 2014. Language diversity and english proficiency part of the "state of asian americans and pacific islanders" series.

Mohd Sanad Zaki Rizvi, Anirudh Srinivasan, Tanuja Ganu, Monojit Choudhury, and Sunayana Sitaram. 2021. GCM: A toolkit for generating synthetic code-mixed text. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 205–211, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *CoRR*, abs/1508.07909.

Sunayana Sitaram, Khyathi Raghavi Chandu, Sai Krishna Rallabandi, and Alan W. Black. 2019. A survey of code-switched speech and language processing. *CoRR*, abs/1904.00784.

Marina Sokolova, Nathalie Japkowicz, and Stan Szpakowicz. 2006. Beyond accuracy, f-score and roc: A family of discriminant measures for performance evaluation. volume Vol. 4304, pages 1015–1021.

Kai Song, Yue Zhang, Heng Yu, Weihua Luo, Kun Wang, and Min Zhang. 2019. Code-switching for enhancing NMT with pre-specified translation. *CoRR*, abs/1904.09107.

Vivek Srivastava and Mayank Singh. 2020. Phinc: A parallel hinglish social media code-mixed corpus for machine translation. *arXiv preprint arXiv:2004.09447*.

Vivek Srivastava and Mayank Singh. 2021. Hinge: A dataset for generation and evaluation of code-mixed hinglish text. *arXiv preprint arXiv:2107.03760*.

Marlies van der Wees, Arianna Bisazza, and Christof Monz. 2017. Dynamic data selection for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1410, Copenhagen, Denmark. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Wei Wang, Ye Tian, Jiquan Ngiam, Yinfei Yang, Isaac Caswell, and Zarana Parekh. 2019. Learning a multitask curriculum for neural machine translation. *CoRR*, abs/1908.10940.

Xinyi Wang, Yulia Tsvetkov, and Graham Neubig. 2020. Balancing training for multilingual neural machine translation. *CoRR*, abs/2004.06748.

Felix Wu, Angela Fan, Alexei Baevski, Yann N. Dauphin, and Michael Auli. 2019. Pay less attention with lightweight and dynamic convolutions. *CoRR*, abs/1901.10430.

Jitao Xu and François Yvon. 2021. Can you traducir this? machine translation for code-switched input. *CoRR*, abs/2105.04846.

A Appendix

Table 4 holds all the hyper-parameters we used for training all models. All experiments were set to halt at patience of 15 updates on the BLEU (Papineni et al., 2002) stabilizing, we found it trained longer with BLEU, but evaluated on WMT specified F1-Score (Sokolova et al., 2006) for the subtasks.

Hyper-parameter	Value
Number of Layers	6
Hidden size	512
FFN inner hidden size	2048
Attention heads	8
Attention head size	64
Dropout	0.1
Attention Dropout	0.0
Warmup Steps	4000
Learning Rate	5e-4
Learning Rate Decay	inverse_sqrt
Batch Size	4096 tokens
Label Smoothing	0.1
Weight Decay	0.0001
Adam ϵ	10^{-9}
Adam β_1	0.9
Adam β_2	0.98
Encoder Criterion Weight	10

Table 4: The hyperparameter values setting for training.