IITP-MT at CALCS2021: English to Hinglish Neural Machine Translation using Unsupervised Synthetic Code-Mixed Parallel Corpus

Ramakrishna Appicharla,* Kamal Kumar Gupta,* Asif Ekbal, Pushpak Bhattacharyya
Department of Computer Science and Engineering
Indian Institute of Technology Patna
Patna, Bihar, India
{appicharla_2021cs01,kamal.pcs17,asif,pb}@iitp.ac.in

Abstract

This paper describes the system submitted by IITP-MT team to Computational Approaches to Linguistic Code-Switching (CALCS 2021) shared task on MT for English → Hinglish. We submit a neural machine translation (NMT) system which is trained on the synthetic code-mixed (cm) English-Hinglish parallel corpus. We propose an approach to create code-mixed parallel corpus from a clean parallel corpus in an *unsupervised manner*. It is an alignment based approach and we do not use any linguistic resources for explicitly marking any token for code-switching. We also train NMT model on the gold corpus provided by the workshop organizers augmented with the generated synthetic code-mixed parallel corpus. The model trained over the generated synthetic cm data achieves 10.09 BLEU points over the given test set.

1 Introduction

In this paper, we describe our submission to shared task on Machine Translation (MT) for English → Hinglish at CALCS 2021. The objective of this shared task to generate Hinglish (Hindi-English Code-Mixed[1]) data from English. In this task, we submit an NMT system which is trained on the parallel code-mixed English-Hinglish synthetic corpus. We generate synthetic corpus in unsupervised fashion and the methodology followed to generate data is independent of languages involved. Since the target Hindi tokens are written in roman script, during the synthetic corpus creation, we transliterate the Hindi tokens from Devanagari script to Roman script.

Code-Mixing (CM) is a very common phenomenon in various social media contents, product description and reviews, educational domain etc. For better understanding and ease in writing, users write posts, comments on social media in code-mixed fashion. It is not consistent or convenient always to translate all the words, especially the named entities, quality related terms etc.

But translating in code-mixed fashion required code-mixed parallel training data. It is possible to generate code-mixed parallel corpus from a clean parallel corpus. From the term 'clean parallel corpus', we refer to a parallel corpus which consists of the non code-mixed parallel sentences. Generally noun tokens, noun phrases and adjectives are the major candidates to be preserved as it is (without translation) in the code-mixed output. This requires a kind of explicit token marking using parser, tagger (part of speech, named entity etc.) to find the eligible candidate tokens for code-mixed replacement. Since this method is dependent on linguistic resources, it is limited to the high resource languages only.

We introduce an alignment based unsupervised approach for generating code-mixed data from parallel corpus which can be used to train the NMT model for code-mixed text translation.

The paper is organized as follows. In section 2, we briefly mention some notable works on translation and generation of synthetic code-mixed corpus. In section 3, we describe our approach to generate synthetic code-mixed corpus along with the system description. Results are described in section 4. Finally, the work is concluded in section 5.

2 Related Works

Translation of code-mixed data has gained popularity in recent times. Menacer et al. (2019) conducted experiments on translating Arabic-English CM data to pure Arabic and/or to pure English with Statistical Machine Translation (SMT) and Neural Machine Translation (NMT) approaches. Dhar et al. (2018) proposed an MT augmentation pipeline which takes CM sentence and determines the most dominating language and translates the

*Equal contribution
[1]Hindi words are romanized

Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching, pages 31–35

https://doi.org/10.26615/978-954-452-056-4_005

remaining words into that language. The resulting sentence will be in one single language and can be translated to other language with the existing MT systems. Yang et al. (2020) have used code-mixing phenomenon and proposed a pre-training strategy for NMT. Song et al. (2019) augmented the code-mixed data with clean data while training the NMT system and reported that this type of data augmentation improves the translation quality of constrained words such as named entities. Singh and Solorio (2017); Masoud et al. (2019); Mahata et al. (2019) also explored various approaches which utilize linguistic resources (such as language identification etc.) to translate the code-mixed data.

There have been some efforts for creating code-mixed data. Gupta et al. (2020) proposed an Encoder-Decoder based model which takes English sentence along with linguistic features as input and generates synthetic code-mixed sentence. Pratapa et al. (2018) explored 'Equivalence Constraint' theory to generate the synthetic code-mixed data which is used to improve the performance of Recurrent Neural Network (RNN) based language model. While Winata et al. (2019) proposed a method to generate code-mixed data using a pointer-generator network, Garg et al. (2018) explored SeqGAN for code-mixed data generation.

3 System Description

In this section, we describe the synthetic parallel corpus creation, dataset and experimental setup of our system.

3.1 Unsupervised Synthetic Code-Mixed Corpus Creation

We utilize the existing parallel corpus to create synthetic code-mixed data. First we learn word-level alignments between source and target sentences of a given parallel corpus of a specific language pair. We use the implementation[2] of *fast_align* algorithm (Dyer et al., 2013) to obtain the alignment matrix. Let $X = \{x_1, x_2, ..., x_m\}$ be the source sentence and $Y = \{y_1, y_2, ..., y_n\}$ be the target sentence. We consider only those alignment pairs $\{x_j, y_k\}$ [for $j = (1, ..., m)$ and $k = (1, ..., n)$] which are having one-to-one mapping, as candidate tokens. By 'One-to-one mapping', we mean that neither $\{x_j\}$ nor $\{y_k\}$ should be aligned to more than one token from their respective counter sides except $\{y_k\}$ and $\{x_j\}$ respectively. The obtained candidate token set is further pruned by removing the pairs where x_j is a stopword. Based on the resulting candidate set, the source token x_j is replaced with aligned target token y_k. The generated code-mixed sentence is in the form: $CM = \{x_1, x_2, ..., y_k, y_l, ..., x_m\}$. Figure 1 shows an example of English-Hindi code-mixed sentence generated through this method.

3.2 Romanization of the Hindi text

The task is to generate Hinglish data in which Hindi words are written in Roman script. But in the generated synthetic code-mixed corpus, Hindi words are written in Devanagari script. In order to convert the Devanagari script to Roman script, we utilize Python based transliteration tool.[3] This convert the Devanagari script to Roman script.

We also create another version of the synthetic code-mixed corpus by replacing the two consecutive vowels with single vowel (Belinkov and Bisk, 2018). We call this version of code-mixed corpus as synthetic code-mixed corpus with user patterns. The main reason to create noisy version of the corpus is to simulate the user writing patterns when writing romanized code-mixed sentences in real-life. An example of such scenario would be, user may write 'Paani' (water) as 'Pani' (water). We tried to capture these scenarios by replacing the consecutive vowels with single vowel. These vowel replacement is done at target side (Hinglish) of the synthetic code-mixed corpus only and source (English) is kept as it is. The gold corpus provided by organizers is not modified in any way and also kept as it is.

3.3 Dataset

We consider English-Hindi IIT Bombay (Kunchukuttan et al., 2018) parallel corpus. We tokenize and true-case English using Moses tokenizer (Koehn et al., 2007) and truecaser [4] scripts and Indic-nlp-library [5] to tokenize Hindi. We remove the sentences having length greater that 150 tokens and created synthetic code-mixed corpus on the resulting corpus as described earlier. The statistics of data used in the experiments are shown in Table 1.

[2]https://github.com/clab/fast_align/

[3]https://github.com/libindic/Transliteration

[4]3https://github.com/mosessmt/mosesdecoder/blob/RELEASE-3.0/scripts/tokenizer/tokenizer.perl

[5]https://github.com/anoopkunchukuttan/indic_nlp_library

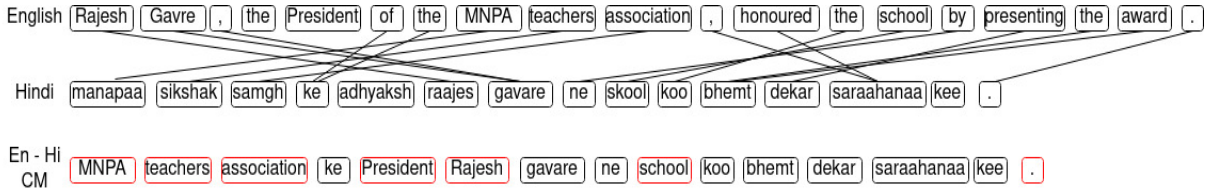

Figure 1: An example of alignment between parallel sentence pair and generated CM sentence. In the CM sentence, the source words that are replaced are shown with red border.

Corpus	Train	Dev
Synthetic CM	1,549,115	-
Synthetic CM + User Patterns	1,549,115	-
Gold	8,060	942
Total	3,106,290	942

Table 1: Data statistics used in the experiment. Synthetic CM: Size of synthetic code-mixed data. Synthetic CM + User Patterns: Size of synthetic code-mixed data with addition of user writing patterns. Gold: Size of gold standard parallel corpus provided by organizers. Train, Dev denotes Training and Development set statistics respectively. In the experiments we use only gold standard corpus as development set.

3.4 Experimental Setup

We conduct the experiments on Transformer based Encoder-Decoder NMT architecture (Vaswani et al., 2017). We use 6 layered Encoder-Decoder stacks with 8 attention heads. Embedding size and hidden sizes are set to 512, dropout rate is set to 0.1. Feed-forward layer consists of 2048 cells. Adam optimizer (Kingma and Ba, 2015) is used for training with 8,000 warmup steps with initial learning rate of 2. We use Sentencepiece (Kudo and Richardson, 2018) with joint vocabulary size of 50K. Models are trained with OpenNMT toolkit [6] (Klein et al., 2017) with batch size of 2048 tokens till convergence and checkpoints are created after every 10,000 steps. All the checkpoints that are created during the training are averaged and considered as the best parameters for each model. During inference, beam size is set to 5.

4 Results

We train two models. Baseline model which is trained on the Gold standard corpus. Second model on the synthetic code-mixed data. We upload our model predictions on the test set provided by organizers to shared task leaderboard[7]. The test set contains 960 sentences. Our model achieved BLEU (Papineni et al., 2002) score of 10.09. Table 2 shows the BLEU scores obtained from the trained models on Development and Test sets. Table 3 shows some sample translations.

Model	Dev	Test
Baseline	2.55	2.45
Synthetic CM	11.52	10.09

Table 2: BLEU scores of the Baseline model and Synthetic Code-Mixed model on Development and Test sets.

Source	Who is your favorite member from the first avengers?
Reference	Tumhara favorite member kaun hai first avengers mein se?
Output	first avengers se aapka favorite member kon hai?
Source	I think it was a robotic shark, but am not sure.
Reference	me sochta hoon voh robotic shark thi, but me sure nahi hoon.
Output	mujhe lagata hai ki yah ek robotik shark hai ,lekin sure nahi hai.
Source	Do you like action movies?
Reference	aap ko action movies pasand hein kya?
Output	Kya tumhe action movies pasand hai?

Table 3: Sample translations generated by trained model

5 Conclusion

In this paper, we described our submission to shared task on MT for English → Hinglish at CALCS 2021. We submitted a system which is trained on synthetic code-mixed corpus generated in unsupervised way. We trained an NMT model

[6] https://opennmt.net/

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

on the synthetic code-mixed corpus and gold standard data provided by organizers. On the test set, the model trained over the gold data provided by the workshop achieves 2.45 BLEU points while the model trained over our generated synthetic cm data yields BLEU score of 10.09. We believe that the proposed method to generate synthetic code-mixed data can be very useful for training MT systems in code-mixed settings as the proposed method does not require any linguistic resources to generate code-mixed data.

References

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations*.

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.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

Saurabh Garg, Tanmay Parekh, and Preethi Jyothi. 2018. Code-switched language models using dual RNNs and same-source pretraining. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3078–3083, Brussels, Belgium. Association for Computational Linguistics.

Deepak Gupta, Asif Ekbal, and Pushpak Bhattacharyya. 2020. A semi-supervised approach to generate the code-mixed text using pre-trained encoder and transfer learning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2267–2280, Online. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

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).

Sainik Kumar Mahata, Soumil Mandal, Dipankar Das, and Sivaji Bandyopadhyay. 2019. Code-mixed to monolingual translation framework. In *Proceedings of the 11th Forum for Information Retrieval Evaluation*, pages 30–35.

Maraim Masoud, Daniel Torregrosa, Paul Buitelaar, and Mihael Arčan. 2019. Back-translation approach for code-switching machine translation: A case study. In *27th AIAI Irish Conference on Artificial Intelligence and Cognitive Science*. AICS2019.

Mohamed Amine Menacer, David Langlois, Denis Jouvet, Dominique Fohr, Odile Mella, and Kamel Smaïli. 2019. Machine translation on a parallel code-switched corpus. In *Canadian Conference on Artificial Intelligence*, pages 426–432. Springer.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. 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.

Thoudam Doren Singh and Thamar Solorio. 2017. Towards translating mixed-code comments from social media. In *International Conference on Computational Linguistics and Intelligent Text Processing*, pages 457–468. Springer.

Kai Song, Yue Zhang, Heng Yu, Weihua Luo, Kun Wang, and Min Zhang. 2019. 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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

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.

Zhen Yang, Bojie Hu, Ambyera Han, Shen Huang, and Qi Ju. 2020. CSP:code-switching pre-training for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2624–2636, Online. Association for Computational Linguistics.