NICT at MixMT 2022: Synthetic Code-Mixed Pre-training and Multi-way Fine-tuning for Hinglish–English Translation

Raj Dabre
National Institute of Information and Communications Technology (NICT)
Kyoto, Japan
raj.dabre@nict.go.jp

Abstract

In this paper, we describe our submission to the Code-mixed Machine Translation (MixMT) shared task. In MixMT, the objective is to translate Hinglish to English and vice versa. For our submissions, we focused on code-mixed pre-training and multi-way fine-tuning. Our submissions achieved rank 4 in terms of automatic evaluation score. For Hinglish to English translation, our submission achieved rank 4 as well.

1 Introduction

Code-mixed translation is the task of translation involving code-mixed languages. A code-mixed language is one which combines words as well as grammar of two or more languages. Code-mixed translation is difficult because of the lack of training data for the same despite its ubiquitous usage. One widely used code-mixed language is Hinglish which combines Hindi and English. Hinglish sentences are typically constructed either by replacing some Hindi words or phrases with English ones in a Hindi sentence or vice versa. Sometimes, a sentence starts off in one language but ends in another. There are also complex cases where the grammatical structures of both languages are melded into one. Hinglish is typically written in Roman letters, although there are cases when it is written in Devanagari.

In this paper we describe our submissions to the MixMT task which involves Hinglish to English and English/Hindi to Hinglish translation. The main challenge of this task is that the parallel corpus available for training models is rather scarce. The total amount of clean, non-synthetic data available for MixMT is around 18,000 examples for both directions. Therefore, we have no choice but to rely on external sources of data, and use them to pre-train models. In our case, we leverage a large amount of Hindi–English parallel data and synthesize pseudo Hinglish data. To do this, perform word alignment on the Hindi–English data and then replace random English phrases with aligned Hindi phrases. We then use the synthetic Hinglish–English parallel data for pre-training. The pre-trained model is then fine-tuned to train a joint bidirectional Hinglish–English translation model. According to the automatic evaluation metrics, we obtain 4th rank and on human evaluation of Hinglish to English translation, we also obtain 4th rank. Unfortunately, for translation into Hinglish our system ends up copying the English inputs as outputs. Although automatic evaluation scores for this are reasonably high, their human evaluation scores are lowest since the sentences are not Hinglish at all.

2 Related Work

Work on code-mixed machine translation is relatively new, especially for Hinglish. Two important works in this regard are HinGE (Srivastava and Singh, 2021) which proposes a dataset for English/Hindi to Hinglish translation and PHINC (Srivastava and Singh, 2020) which proposes a dataset for Hinglish to English translation. The HinGE dataset contains natural as well as human rated synthetic examples in both Hindi and English as source languages. Having two sources is expected to help in Hinglish generation, as the model will have the advantage of contexts from both sources. In our case, we did not leverage both sources and focused only on English. On the other hand, PHINC is designed for Hinglish to English translation and is much larger than HinGE. Neither of these datasets are perfect and contain some noisy examples, but the lack of other datasets leaves us with no choice.

Due to lack of code-mixed data, it is natural to consider synthetic code-mixed data creation where Gupta et al. (2020) show that leveraging an XLM model (CONNEAU and Lample, 2019) and linguistic features can help generate high quality code-mixed sentences. However, we opted for a quicker way using word alignment and phrase substitution

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 1122 - 1125

approach. Using pre-trained models, can be very helpful in code-mixed translation as they are able to represent them effectively (Santy et al., 2021). Agarwal et al. (2021) have shown that pre-trained models (Liu et al., 2020) are highly effective, but we focused more on using our own models trained on our synthetic data.

Apart from machine translation, code-mix Hinglish has been reasonably explored for natural language understanding tasks, particularly for sentiment analysis. We refer interested readers to the following works: Baroi et al. (2020); Singh and Lefever (2020); Mathur et al. (2018); Bhange and Kasliwal (2020).

3 Methods

We describe the synthetic code mixed pre-training and multi-way fine-tuning approaches we used for our submissions.

3.1 Synthesizing Code-Mixed Data

We assume the existence of a large amount of Hindi–English parallel corpus, which we use to synthesize Hinglish. Since Hinglish is written in the Roman alphabet, we first Romanize it. We then use an aligner to obtain word alignments between Hindi and English. For each English sentence, we take a random span of tokens, find the corresponding aligned span of tokens in Hindi and replace it with the English tokens span. We note that this assumes that the language structure of Hindi is preserved in this process. To determine the span in the target language, we find the indices of the aligned target words and then choose the smallest as the starting index and the largest as the ending index as the span to be replaced. This is known as the min-max approach, which was used by Zenkel et al. (2021). As a result of this process we obtain a Hinglish–English parallel corpus where Hinglish is synthetic.

3.2 Code-mixed Pre-training

We train a multilingual model (Dabre et al., 2020; Firat et al., 2016; Johnson et al., 2017) model for synthetic Hinglish to English and English to synthetic Hinglish. We append a token indicating the source language at the end of the source sentence and a token indicating the target language at the beginning of the target sentence. This bidirectional model is trained till convergence on the development set provided by the organizers after the dev set evaluation phase. We expect that code-mixed pre-training, even if the Hinglish is synthetic, should help overcome the scarcity of code-mixed parallel corpus.

3.3 Multi-way Fine-tuning

We fine-tune the pre-trained model on Hinglish to English and English to Hinglish jointly. We use a small subset of the English side[1] of our synthetic data and the entire clean parallel corpus (PHINC+HinGE) together. We do this to prevent the model from overfitting on the small training data. The English subset is used as the source as well as the target and hence, in order to prevent the model from learning to copy the English data, we randomly mask spans of English tokens on the source. This is the same as denoising, which is used in BART (Lewis et al., 2020). This concept of using the pre-training data along with the fine-tuning data is also known as mixed fine-tuning (Dabre et al., 2019; Chu et al., 2017). As during pre-training, the development set data is used.

4 Experiments

We describe our experiments in our submissions.

4.1 Datasets and Pre-processing

We use the PHINC and HinGE datasets for our experiments. We do not use the synthetic parts of HinGE. During our preliminary experiments we used the development data provided with HinGE but found it to be unreliable and therefore used the development data provided after the first evaluation phase. We combined the data from both sources and overall we had 18,095 training instances for each direction for a total of 36,190 training instances. Note that HinGE has sources in English as well as Hindi, and this is also available for the development and test sets for translation into Hinglish. However, we do not explore multi-source translation in this paper. For pre-training, we used the Hindi–English part of the Samanantar dataset[2] (Ramesh et al., 2022) which contains 8.56M parallel sentences. We used the Romanization script from the Indic NLP Library[3] to convert

[1] We do not use the Hinglish side since it's synthetic and do not want it to interfere in the learning of actual Hinglish.

[2] https://indicnlp.ai4bharat.org/samanantar/

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

Direction	Rouge-L	WER	Human Rating
Hinglish → English	0.52878 (4)	0.71517 (4)	2.85
English → Hinglish	0.46276 (4)	0.79271 (5)	1.00

Table 1: Official results of evaluation of Hinglish to English and English to Hinglish.

Devanagari to the Roman alphabet for Hindi. No other pre-processing was done.

4.2 Model Training and Decoding

We train transformer models (Vaswani et al., 2017) using the transformer-big settings. We used the YANMTT toolkit[4] (Dabre and Sumita, 2021) for training our models. We trained a joint Hinglish and English tokenizer of 16,000 subwords using all the synthetic and real training data we had. Pre-training was done on 8 NVIDIA V100 GPUs till convergence on the development data. (Mixed) Fine-tuning was done on a single GPU due to the relatively smaller size of the data. Once training has converged, we choose the checkpoints giving the highest development scores for decoding the test sets. We experimented with both BLEU and Rouge-L as metrics to determine convergence, but used BLEU as it is much stricter. We decode using beam search with a beam size of 32 and a length penalty of 1.6 both of which are empirically determined on the development set.

4.3 Results

Table 1 shows the official results obtained using the official evaluation servers. The organizers use Rouge-L and Word Error Rate (WER) as well as Human Ratings by evaluating 50 translations from our submissions. Overall, our automatic evaluation scores achieved a rank of 4 out of 8 participants. Compared to some of the baselines trained using only HinGE and PHINC, our main results using pre-training and fine-tuning are vastly better.

4.4 Analysis

We got a human rating score of 1 for translation into Hinglish and upon investigation we noted that our model simply copies the English sentence to the target. We are not sure why this happens. Regardless, on the development set, copying seems to give high BLEU and Rouge-L scores. However, the output is not Hinglish and is heavily penalized. We also did not conduct back-translation (Sennrich et al., 2016) of English into Hinglish due to this issue. We will probe our models deeper to understand why this happens. Due to lack of access to the official evaluation interface after the submission deadline, we were unable to conduct additional experiments.

[4] https://github.com/prajdabre/yanmtt

5 Conclusion

In this paper, we have described our submission to the MixMT shared task at WMT 2022. We have used a combination of synthetic Hinglish–English parallel data creation, pre-training and fine-tuning to obtain our submissions which ranked 4th. Our analyses reveal that our English to Hinglish translation model actually ended up copying the English sentence as target. We will investigate and fix this in the future.

References

Vibhav Agarwal, Pooja Rao, and Dinesh Babu Jayagopi. 2021. Hinglish to English machine translation using multilingual transformers. In *Proceedings of the Student Research Workshop Associated with RANLP 2021*, pages 16–21, Online. INCOMA Ltd.

Subhra Jyoti Baroi, Nivedita Singh, Ringki Das, and Thoudam Doren Singh. 2020. NITS-Hinglish-SentiMix at SemEval-2020 task 9: Sentiment analysis for code-mixed social media text using an ensemble model. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1298–1303, Barcelona (online). International Committee for Computational Linguistics.

Meghana Bhange and Nirant Kasliwal. 2020. HinglishNLP at SemEval-2020 task 9: Fine-tuned language models for Hinglish sentiment detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 934–939, Barcelona (online). International Committee for Computational Linguistics.

Chenhui Chu, Raj Dabre, and Sadao Kurohashi. 2017. An empirical comparison of domain adaptation methods for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 385–391, Vancouver, Canada. Association for Computational Linguistics.

Alexis CONNEAU and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. 2020. A survey of multilingual neural machine translation. *ACM Comput. Surv.*, 53(5).

Raj Dabre, Atsushi Fujita, and Chenhui Chu. 2019. Exploiting multilingualism through multistage fine-tuning for low-resource neural machine translation. 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 1410–1416, Hong Kong, China. Association for Computational Linguistics.

Raj Dabre and Eiichiro Sumita. 2021. YANMTT: yet another neural machine translation toolkit. *CoRR*, abs/2108.11126.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, San Diego, California. 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.

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.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

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.

Puneet Mathur, Ramit Sawhney, Meghna Ayyar, and Rajiv Shah. 2018. Did you offend me? classification of offensive tweets in Hinglish language. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 138–148, Brussels, Belgium. 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, Srihari Nagaraj, Kumar Deepak, Vivek Raghavan, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh Shantadevi Khapra. 2022. Samanantar: The largest publicly available parallel corpora collection for 11 indic languages. *Transactions of the Association for Computational Linguistics*, 10:145–162.

Sebastin Santy, Anirudh Srinivasan, and Monojit Choudhury. 2021. BERTologiCoMix: How does code-mixing interact with multilingual BERT? In *Proceedings of the Second Workshop on Domain Adaptation for NLP*, pages 111–121, Kyiv, Ukraine. Association for Computational Linguistics.

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.

Pranaydeep Singh and Els Lefever. 2020. Sentiment analysis for Hinglish code-mixed tweets by means of cross-lingual word embeddings. In *Proceedings of the The 4th Workshop on Computational Approaches to Code Switching*, pages 45–51, Marseille, France. European Language Resources Association.

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.

Vivek Srivastava and Mayank Singh. 2021. HinGE: A dataset for generation and evaluation of code-mixed Hinglish text. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 200–208, Punta Cana, Dominican Republic. 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.

Thomas Zenkel, Joern Wuebker, and John DeNero. 2021. Automatic bilingual markup transfer. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3524–3533, Punta Cana, Dominican Republic. Association for Computational Linguistics.