MUCS@MixMT: indicTrans-based Machine Translation for Hinglish Text

Asha Hegde[a], **Hosahalli Lakshmaiah Shashirekha**[b]
Department of Computer Science, Mangalore University, Mangalore, India
{[a]hegdekasha,[b]hlsrekha}@gmail.com

Abstract

Code-mixing is the phenomena of mixing various linguistic units such as paragraphs, sentences, phrases, words, etc., of two or more languages in any text. It is predominantly used to post the comments by social media users who know more than one language. Processing code-mixed text is challenging because of its complex characteristics and lack of tools that support such data. Developing efficient Machine Translation (MT) systems for code-mixed text is challenging due to lack of code-mixed data. Further, existing MT systems developed to translate monolingual data are not portable to translate code-mixed text mainly due to the informal nature of code-mixed data. To address the MT challenges of code-mixed text, this paper describes the proposed MT models submitted by our team MUCS, to the Code-mixed Machine Translation (MixMT) shared task in the Workshop on Machine Translation (WMT) organized in connection with Empirical models in Natural Language Processing (EMNLP) 2022. This shared task has two subtasks: i) subtask 1 - to translate English sentences and their corresponding Hindi translations written in Devanagari script into Hinglish (English-Hindi code-mixed text written in Latin script) text and ii) subtask 2 - to translate Hinglish text into English text. The proposed models that translate English text to Hinglish text and vice versa, comprise of i) transliterating Hinglish text from Latin to Devanagari script and vice versa, ii) pseudo translation generation using existing models, and iii) efficient target generation by combining the pseudo translations along with the training data provided by the shared task organizers. The proposed models obtained 5th and 3rd rank with Recall-Oriented Under-study for Gisting Evaluation (ROUGE) scores of 0.35806 and 0.55453 for subtask 1 and subtask 2 respectively.

1 Introduction

In linguistic terms, code-mixing is the practice of switching between two or more languages within or across sentences/words in any text (Joshi, 1982). Due to the widespread use of social media platforms like Twitter, Facebook, Reddit, etc., users are generating more and more code-mixed content. In Indian scenario, social media users usually blend English with their mother tongue or local language, for instance, English and Hindi, mainly for the technological limitations of computer keyboard or smartphone keypads to enter text in local languages. Further, as most of the text processing tasks are developed for handling monolingual and formal text, informal and/or code-mixed text such as Hinglish is less explored. As the code-mixed text like Hinglish is increasing day by day, many applications such as MT, sentiment analysis, emotion analysis, etc., are also increasing. This has created a great demand for the tools and resources to process code-mixed data. Sample Hinglish text along with their Hindi and English translations are given in Table 1.

In recent years, pre-trained transformer-based language models have become state-of-the-art models for most of the downstream tasks including MT, text classification, text generation, and natural language understanding. To train such models, underlying data is drawn from a sizable monolingual corpus that is available in Wikipedia, book corpora, etc. Several models like Multilingual Bidirectional and Auto-Regressive Transformer (mBART) (Liu et al., 2020) and Multilingual Text to Text Transformer (mT5) (Xue et al., 2021) are readily available for many languages. However, due to the scarcity of code-mixed corpus, developing the pretrained language models for code-mixed text is very challenging.

MT being one of the important applications of code-mixed texts mainly focuses on translating monolingual text leaving aside the code-mixed data. Further, for under-resourced languages with rich morphological features like Hindi (Sangwan and Bhatia, 2021), developing MT models become more challenging in the code-mixed sce-

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 1131–1135

Hinglish	Hindi	English
tumhen २०११ ka ted prize mil gaya hai	तुम्हें २०११ का टेड प्राइज़ मिल गया है	you won the TED Prize 2011
aur jab unhen yad dilaya jata hai , to ve yad nahin karte	और जब उन्हें याद दिलाया जाता है, तो वे याद नहीं करते	and, when reminded, do not remember
aap prat 10 baje vahan ho tej	आप प्रात 10 बजे वहाँ हो तेज	You be there at 10 A. M. sharp
aaj tonight ek year poora ho jaega	आज रात एक साल पूरा हो जाएगा	Tonight, it will be a year

Table 1: Sample Hinglish text and their Hindi and English translations

nario. To address these challenges, in this paper, we - team MUCS, describe the models submitted to MixMT-2022[1] shared task organized by WMT-2022 at EMNLP 2022. The shared task consists of two subtasks: i) subtask 1 - to translate English sentences and their corresponding Hindi translations into Hinglish text and ii) subtask 2 - to translate Hinglish text into English text. The proposed methodology consists of i) transliteration of Hinglish text from Latin script to Devanagari script and vice versa, ii) generating pseudo translations for monolingual data using pretrained MT models, and iii) target generation by fine-tuning the pretrained models with a combination of the pseudo parallel data obtained as the output of pseudo translations and the dataset provided by the organizers of the shared task.

The following is a breakdown of the paper's structure: Section 2 contains the related work and the proposed methodology is explained in Section 3. Section 4 gives the details about experiments and results and the paper concludes in Section 5 with future work.

2 Related work

Due to the increasing amount of code-mixed text, MT of code-mixed text is gaining attention of the researchers and the description of few of the models developed to translate Hinglish text into English text and vice versa are given below:

Srivastava and Singh (2020) manually developed a parallel corpus of 13,738 Hinglish sentences and their translations in English with the help of 54 annotators. They proposed a simple tagging approach for tagging each token in a sentence with the language it belongs to and evaluated Bing Translate (BT) and Google Translate (GT) models - the two popular MT services using their parallel corpus. Among the two models, GT model outperformed with a better Bilingual Evaluation Understudy (BLEU) score of 0.153 when compared to that of BT. Dhar et al. (2018) manually developed a Hinglish-English parallel corpus of 6,096 parallel sentences with the help of 4 human translators. Using a language identification technique, they tagged every word in a sentence with the name of the language to which it belongs. They proposed an MT model comprising three steps: i) identifying the matrix language, ii) translation of source text into matrix language, and iii) translation of matrix language into the target language by training BT. Considering steps i) and ii) as preprocessing, they obtained considerable translation with BLEU score of 25.0.

Jawahar et al. (2021) created a parallel corpus of 17.8 million English-Hinglish sentence pairs by leveraging bilingual word embeddings to translate English text into Hinglish text and vice versa. Further, they fine-tuned mBART and mT5 - the pretrained text generators using their newly constructed parallel corpus. The mT5 model obtained a better BLEU score of 13.95 compared to that of mBART. Gautam et al. (2021) proposed an effective fine-tuning of mBART using English-Hinglish dataset[2] to translate English text to Hinglish text and vice versa. They transliterated the Hinglish text in Latin script to Devanagari script to fine-tune the mBART model and obtained BLEU scores of 11.86 and 12.22 for Hinglish to English and English to Hinglish translations respectively.

From the literature, it is clear that very few attempts are made to explore English-Hinglish code-mixed parallel corpus for MT. Hence, there is enough space to explore new techniques in this direction.

[1] https://codalab.lisn.upsaclay.fr/competitions/2861#learn_the_details

[2] https://code-switching.github.io/2021

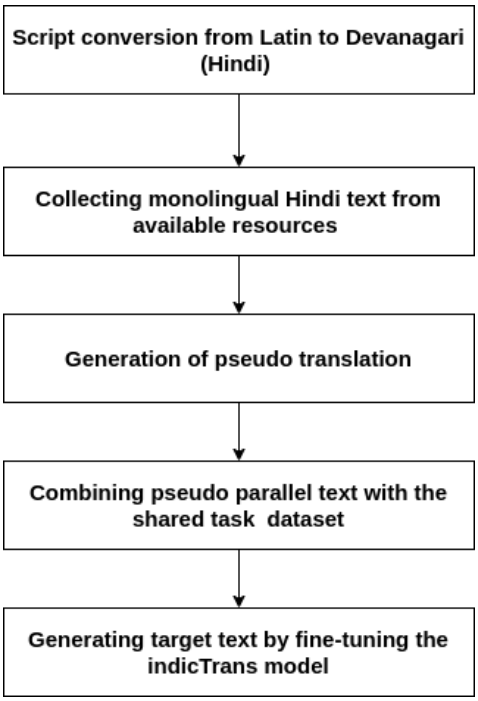

Figure 1: Workflow of the proposed method

3 Proposed methodology

Inspired by Gautam et al. (2021), a pipeline of transliteration and fine-tuning indicTrans[3] (used for translation) pretrained models is proposed to address subtask 1 and subtask 2. EN-Indic and Indic-EN models are used for generating Hinglish text and pseudo translations respectively. The purpose of transliteration is to utilize pretrained models which are trained using the text in their native script. Further, the proposed methodology also consists of the generation of pseudo translations where pseudo translation mimics the translation process.

The framework of the proposed model is given in Figure 1 and the system descriptions of each subtasks are given below:

Subtask 1 - In addition to the dataset provided by the organizers[4] for this shared task, monolingual Hindi text is collected from the available resources[5] and the further steps used to accomplish the subtask 1 are given below:

1. Transliteration is carried out using indic-trans[6] to transliterate Hinglish text in Latin script to Devanagari script

2. EN-Indic and Indic-EN models trained on Samanantar[7] corpus are used for translations (Ramesh et al., 2022)

3. Pseudo translations are generated using the Indic-EN model considering monolingual Hindi text

4. The shared task dataset is combined with the pseudo parallel data and EN-Indic model is then fine-tuned on this data

5. Finally, the target Hinglish text is generated by transliterating Devanagari script to Latin script using indic-trans

Subtask 2 - For subtask 2, the procedure similar to that of subtask 1 is followed considering Hinglish text as the source and English text as the target to generate the required output.

Subtask	Train set	Development set	Test set
subtask 1	2,766	500	1,500
subtask 2	13,738	500	1,500

Table 2: Statistics of the shared task dataset for both the subtasks in terms of the number of sentences

4 Experiments and Results

The statistics of the dataset provided by the organizers for both the subtasks which are used to build the proposed models are given in Table 2. The data provided for subtask 1 is the synthetic data (Srivastava and Singh, 2021) which consists of English sentences and their corresponding Hindi translations as the source and Hinglish as the target. For subtask 2, the dataset consists of Hinglish-English sentence pairs (Srivastava and Singh, 2020) to generate English text.

EN-Indic and Indic-EN models which are trained on Samanantar corpus are fine-tuned with the combination of the shared task dataset and pseudo parallel text. Exhaustive experiments are carried out to get the best results by tuning the hyperparameters, which control the learning process of EN-Indic and Indic-En models. Table 3 gives the hyperparameters and their values used to fine-tune the EN-Indic and Indic-EN models that gave the best results on development set.

The user predictions for the given Test set submitted to the organizers of the shared task are evaluated based on ROUGE score and Word Error Rate (WER). ROUGE score is calculated based

[3] https://indicnlp.ai4bharat.org/indic-trans/
[4] Codalab competitions
[5] https://indicnlp.ai4bharat.org/samanantar/
[6] https://github.com/libindic/indic-trans
[7] https://indicnlp.ai4bharat.org/samanantar/

Hyperparameters	Values
max-token	1,568
learning rate	0.00003
label smoothing	0.1
optimizer	adam
dropout	0.2

Table 3: Hyperparameters and their values used to fine-tune the EN-Indic and Indic-EN models

Subtask		ROUGE	WER
subtask 1	Development set	0.38935	0.72310
	Test set	0.35806	0.76096
subtask 2	Development set	0.54556	0.65938
	Test set	0.55453	0.64737

Table 4: Performance measures of the proposed method for both Development set and Test set

on the overlapping of n-grams between the candidate string and reference string whereas WER score is calculated by dividing the number of errors by the total number of words. Performance measures of the proposed models for both Development set and Test set is given in Table 4. From Table 4, it is clear that the ROUGE score of subtask 2 is better than subtask 1 as the dataset used for subtask 1 is very small compared to that of subtask 2. Further, the performance of Indic-EN model is better than EN-Indic model (Ramesh et al., 2022) and the same is reflected in Table 4. The comparison of the ROUGE score of the proposed models with the models submitted by all the participants of the shared task for subtask 1 and subtask 2 are shown in Figure 2 and 3 respectively. From Figure 2 and 3, it is clear that the proposed method obtained considerable ROUGE scores for both the subtasks.

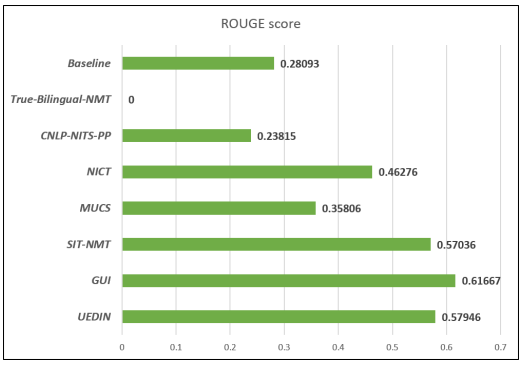

Figure 2: Comparison of ROUGE score of participated teams with the proposed model for subtask 1

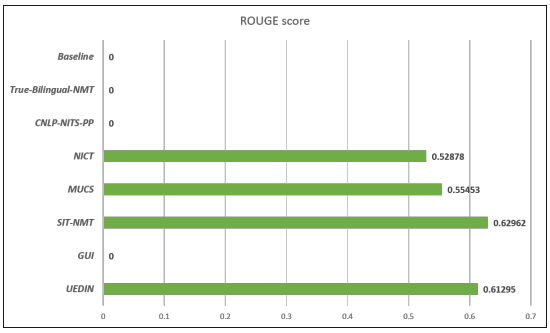

Figure 3: Comparison of ROUGE score of participated teams with the proposed model for subtask 2

5 Conclusion and Future work

This paper describes the models submitted by our team - MUCS to MixMT 2022 shared task to perform MT from English text and their corresponding Hindi translations into Hinglish text and from Hinglish text to English. The proposed models consist of transliteration and pseudo translation generation followed by fine-tuning the pretrained MT models using the combination of pseudo parallel data and the shared task dataset for target generation. These models obtained ROUGE scores of 0.35806 and 0.55453 securing 5th and 3rd rank for subtask 1 and subtask 2 respectively. The efficient transliteration techniques with effective fine-tuning of the pretrained models for code-mixed Hinglish translation will be explored further.

References

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.

Devansh Gautam, Prashant Kodali, Kshitij Gupta, Anmol Goel, Manish Shrivastava, and Ponnurangam Kumaraguru. 2021. CoMeT: Towards Code-Mixed Translation Using Parallel Monolingual Sentences. In *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, pages 47–55.

Ganesh Jawahar, El Moatez Billah Nagoudi, Muhammad Abdul-Mageed, and Laks VS Lakshmanan. 2021. Exploring Text-to-Text Transformers for English to Hinglish Machine Translation with Synthetic Code-mixing. In *arXiv preprint arXiv:2105.08807*.

Aravind K. Joshi. 1982. Processing of Sentences With Intra-Sentential Code-Switching. In *Coling 1982: Proceedings of the Ninth International Conference on 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. In *Transactions of the Association for Computational Linguistics*, pages 726–742.

Gowtham Ramesh, Sumanth Doddapaneni, Aravinth Bheemaraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Divyanshu Kakwani, Navneet Kumar, et al. 2022. Samanantar: The Largest Publicly Available Parallel Corpora Collection for 11 Indic Languages. In *Transactions of the Association for Computational Linguistics*, pages 145–162. MIT Press.

Saurabh R Sangwan and MPS Bhatia. 2021. Denigrate Comment Detection in Low-resource Hindi Language using Attention-based Residual Networks. In *Transactions on Asian and Low-Resource Language Information Processing*, pages 1–14.

Vivek Srivastava and Mayank Singh. 2020. PHINC: A parallel Hinglish Social Media Code-mixed Corpus for Machine Translation. In *arXiv preprint arXiv:2004.09447*.

Vivek Srivastava and Mayank Singh. 2021. Hinge: A Dataset for Generation and Evaluation of Code-mixed Hinglish Text. In *arXiv preprint arXiv:2107.03760*.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498.