Overview and Results of MixMT Shared-Task at WMT 2022

Vivek Srivastava  
TCS Research  
Pune, Maharashtra, India  
srivastava.vivek2@tcs.com

Mayank Singh  
IIT Gandhinagar  
Gandhinagar, Gujarat, India  
singh.mayank@iitgn.ac.in

Abstract

In this paper, we present an overview of the WMT 2022 shared task on code-mixed machine translation (MixMT). In this shared task, we hosted two code-mixed machine translation subtasks in the following settings: (i) monolingual to code-mixed translation and (ii) code-mixed to monolingual translation. In both the subtasks, we received registration and participation from teams across the globe showing an interest and need to immediately address the challenges with machine translation involving code-mixed and low-resource languages.

1 Introduction

Code-mixing (or code-switching) is an interesting manifestation of multilingualism in communities across the globe. Lately, we observe an uptick in the interest and efforts of the computational linguistic community to solve a multitude of challenges with code-mixed languages. Several interesting resources and computational models have been proposed for problems such as language identification (Barman et al., 2014; Thara and Poornachandran, 2021), text generation (Gupta et al., 2020; Rizvi et al., 2021), and sentiment analysis (Chakravarthi et al., 2020; Patwa et al., 2020).

Machine translation which is an active area of research and development for monolingual languages is at the outset for code-mixed languages (Chen et al., 2022). In this shared task, we aim to explore the machine translation task involving a popular code-mixed language i.e., Hinglish. The source Hindi and English sentences are translations of each other. The source language is in Devanagari script whereas the target Hinglish language text is in the Roman script.

In the MixMT shared task, we hosted two subtasks involving a code-mixed language i.e. Hinglish. A brief description of both subtasks is given below:

1. **Monolingual to code-mixed machine translation (M2CM):** In this subtask, Hindi and English are the two source languages and the target language is Hinglish. The source Hindi and English sentences are translations of each other. The source language is in Devanagari script whereas the target Hinglish language text is in the Roman script.

2. **Code-mixed to monolingual machine translation (CM2M):** In this subtask, Hinglish is the source language and the target language is English. Both the English and Hinglish text are written in the Roman script.

2 Dataset

**Training datasets:** We provide the following training datasets for both subtasks:

1. **M2CM:** For this subtask, HinGE (Srivastava and Singh, 2021a) is the primary training dataset. It contains parallel English-Hindi sentences along with multiple human-generated Hinglish sentences. For each data instance, it also contains two synthetically generated Hinglish sentences. The dataset was also used as part of the HinglishEval shared task (Srivastava and Singh, 2021b). We provide the entire HinglishEval data of ≈ 2k samples (train, validation, and test set together) as part of the training data for the MixMT shared task.
2. **CM2M**: For this subtask, PHINC (Srivastava and Singh, 2020) is the primary training dataset. It contains 13,738 parallel sentences in the Hinglish and English languages.

**Evaluation datasets**: To evaluate both the subtasks, we have created an in-house hidden evaluation dataset. For both subtasks, the validation dataset contains 500 samples and the test dataset has 1500 samples. The evaluation dataset is available here: bit.ly/3UZLdFm.

2.3 **Baseline system and evaluation**

We use Google Translate as a baseline for both subtasks. For **M2CM** subtask, we translate Hindi sentences (in Devanagari script) into English and evaluate them against the reference Hinglish sentences. For **CM2M** subtask, we translate the Hinglish sentences into English by setting the language of the Hinglish text as Hindi.

**Evaluation**: We use two evaluation metrics for both the subtasks: ROUGE-L (F1-score) and Word Error Rate (WER). Also, we perform a human-based qualitative evaluation of both subtasks. Table 1 shows the policy of the human-based evaluation of both subtasks.

2.4 **Constrained system**

We distinguish between the constrained and unconstrained systems based on the following criteria:
1. The system using an external dataset (apart from HinGE and PHINC datasets) will be considered unconstrained.
2. We allow public pre-trained models in a constrained system given that it is accessible to all the teams.

3 **Submissions**

We received the submissions from seven teams (listed alphabetically):

1. **CNLP-NITS-PP** (Laskar et al., 2022): They leverage the external parallel corpus (Samanantar (Ramesh et al., 2022)) to train their translation model which is built using OpenNMT-py framework (Klein et al., 2017) with the default setting. To generate the synthetic dataset, they transliterate and align the words in parallel sentences. Finally, they augment the provided dataset with the synthetic dataset to train their model.

2. **Domain Curricula for Code-switched MT (DC)** (Raheem and Elrashid, 2022): They experiment with different combinations of pre-training fine-tuning setups. They leverage the synthetic code-mixed dataset generated using the IIT-B parallel corpus (Kunchukuttan et al., 2018) and matrix language theory (Myers-Scotton). Further, the mixed data pretraining with synthetic and task-specific data shows the best result on the evaluation dataset. To build the translation model, they use transformers (Vaswani et al., 2017) and fairseq toolkit (Ott et al., 2019).

3. **Gui** (Gahoi et al., 2022): They leverage the multilingual pre-trained models to build their translation system. For **M2CM** task, they fine-tune multilingual-BART (Liu et al., 2020) on the task-specific data with reduced vocabulary. They reduce the vocabulary using the tokens present in the task dataset, IIT-B parallel corpus (Kunchukuttan et al., 2018), and the Dakhshina dataset (Roark et al., 2020). They also perform the post-processing on the output generated from the fine-tuned model. For **CM2M** task, they fine-tune Salesken.AI’s pre-trained model provided on Huggingface Transformers which is a finetuned Helsinki’s OPUS-MT model on AI4Bharat’s Samanantar dataset (Ramesh et al., 2022).

4. **MUCS** (Hegde and Shashirekha, 2022): Their translation model for both the task is built around transliteration (Bhat et al., 2015) and fine-tuning the IndicTrans pre-trained model (Ramesh et al., 2022). They generate synthetic parallel data using the Samanantar corpus (Ramesh et al., 2022). They further fine-tune the IndicTrans model jointly with the synthetic and task-specific datasets.

5. **NICT** (Dabre, 2022): They propose a synthetic code-mixed data based pre-training and a multi-way fine-tuning strategy. To generate the synthetic dataset, they leverage the Samanantar corpus (Ramesh et al., 2022), the transliteration toolkit1, and a min-max based approach for word alignment (Zenkel et al., 2021). They pre-train a multilingual model on the synthetic Hinglish-English and English-synthetic Hinglish dataset. To perform the multi-way fine-tuning, they fine-tune the pre-trained model on Hinglish to English and English to Hinglish jointly using a small subset of the English side

---

1 https://github.com/anoopkunchukuttan/indic_nlp_library
Table 1: Human-based evaluation policy for M2CM and CM2M subtasks. The underlined phrase highlights the center of attention for the corresponding rating.

<table>
<thead>
<tr>
<th>Rating</th>
<th>M2CM</th>
<th>CM2M</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Best Generated Hinglish sentence</td>
<td>Correctly translated sentence conveying the exact same information as the source sentence</td>
</tr>
<tr>
<td>4</td>
<td>A Hinglish sentence with minimal grammatical mistakes but less likely in general parlance</td>
<td>A translated sentence with minimal grammatical mistakes</td>
</tr>
<tr>
<td>3</td>
<td>A Hinglish sentence that contains mainly grammatical mistakes</td>
<td>A translated sentence that contains mainly grammatical mistakes</td>
</tr>
<tr>
<td>2</td>
<td>A Hinglish sentence containing fairly large volumes of lexical and grammatical mistakes</td>
<td>A translated sentence containing fairly large volumes of lexical and grammatical mistakes</td>
</tr>
<tr>
<td>1</td>
<td>Worst Generated Sentence. They are monolingual either in Romanized Hindi or English</td>
<td>A translated sentence with poor semantics and irrelevant to the source sentence</td>
</tr>
</tbody>
</table>

of the synthetic data and the entire parallel corpus (PHINC & HinGE) together. They use a denoising strategy similar to BART (Lewis et al., 2020) by randomly masking English words in the source sentence. They use the YANMTT toolkit (Dabre and Sumita, 2021) to their translation model.

6. SIT-NMT (Khan et al., 2022): They experiment with a variety of multilingual pre-trained models such as multilingual BART (Liu et al., 2020) and multilingual T5 (Xue et al., 2021). They fine-tune these pre-trained models on external datasets. For M2CM task, they use Kaggle Hi-En (Chokhra, 2020) and MUSE Hi-En dictionary (Lample et al., 2018). For CM2M task, they use CMU movie reviews data (Zhou et al., 2018) and CALCS’21 dataset (Chen et al., 2022). They also use selected WMT’14 News Hi-En sentences (Bojar et al., 2014) and the MTNT Fr-En and Ja-En data (Michel and Neubig, 2018). In addition, they also increase the size of the dataset by back-translation samples of the English side of Tatoeba Spanish dataset to the English (Project, 2022) and Sentiment140 dataset (Go et al., 2009) into Hinglish using Google translate. Further, to enhance the model’s performance, they perform the validation tuning on the task-specific validation dataset and use a multi-run ensemble (Koehn, 2020) to combine multiple model’s best checkpoints.

7. UEDIN (Kirefu et al., 2022): Their submission focuses on data generation using back-translation from monolingual resources. For M2CM subtask, they explore the impact of constrained and unconstrained decoding strategies. They use the Samanantar corpus (Ramesh et al., 2022) as an external resource for back-translation. For CM2M subtask, they explore several pretraining techniques, ranging from simple initialization from existing machine translation models to aligned augmentation (Pan et al., 2021) which is a denoising-based pretraining technique.

4 Results and Analysis

In this section, we present the results from automatic and human-based evaluation of the submissions from the seven teams. As discussed in Section 2.3, we use ROUGE-L F1 score (R-L) and Word Error Rate (WER) for automatic evaluation. R-L score can vary from 0 to 1 whereas WER can take a value greater than or equal to 0. A high R-L score and a low WER score are preferred.

Table 2 shows the results of the automatic evaluation for both subtasks. For M2CM subtask, the Gui team’s submission achieves best R-L score whereas the team UEDIN and SIT-NMT achieve the second and third best R-L scores respectively. SIT-NMT’s submission outperforms the other systems and scores the lowest WER for this subtask. For CM2M subtask, SIT-NMT is the best-performing team followed by UEDIN and MUCS on both metrics.

Table 3 shows the human-based evaluation of the submissions from different teams. The evaluation policy is given in Table 1. Following the evaluation policy, we evaluate the output of 20 samples for each subtask from each team. SIT-NMT ranks first
Table 2: Evaluation results on the test set. We color code the best, second best, and third best team on a given metric for a subtask.

<table>
<thead>
<tr>
<th>Team</th>
<th>M2CM</th>
<th>CM2M</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R-L</td>
<td>WER</td>
</tr>
<tr>
<td>Baseline</td>
<td>0.280</td>
<td>0.926</td>
</tr>
<tr>
<td>CNLP</td>
<td>0.238</td>
<td>0.926</td>
</tr>
<tr>
<td>DC</td>
<td>0.033</td>
<td>1.560</td>
</tr>
<tr>
<td>Gui</td>
<td>0.616</td>
<td>0.633</td>
</tr>
<tr>
<td>MUCS</td>
<td>0.358</td>
<td>0.760</td>
</tr>
<tr>
<td>NICT</td>
<td>0.462</td>
<td>1.560</td>
</tr>
<tr>
<td>SIT-NMT</td>
<td>0.571</td>
<td>0.547</td>
</tr>
<tr>
<td>UEDIN</td>
<td>0.579</td>
<td>0.561</td>
</tr>
</tbody>
</table>

Table 3: Human-based evaluation of submitted systems on the test set. We color code the best, second best, and third best team for a subtask.

<table>
<thead>
<tr>
<th>Team</th>
<th>M2CM</th>
<th>CM2M</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CNLP</td>
<td>2.1 ± 0.64</td>
<td>1.35 ± 0.74</td>
</tr>
<tr>
<td>DC</td>
<td>1.75 ± 0.71</td>
<td>1.55 ± 1.09</td>
</tr>
<tr>
<td>Gui</td>
<td>3.75 ± 1.20</td>
<td>1.8 ± 1.1</td>
</tr>
<tr>
<td>MUCS</td>
<td>1 ± 0</td>
<td>2.9 ± 1.51</td>
</tr>
<tr>
<td>NICT</td>
<td>1 ± 0</td>
<td>2.85 ± 1.30</td>
</tr>
<tr>
<td>SIT-NMT</td>
<td>3.85 ± 1.38</td>
<td>4.1 ± 1.07</td>
</tr>
<tr>
<td>UEDIN</td>
<td>3.85 + 1.53</td>
<td>3.75 + 1.16</td>
</tr>
</tbody>
</table>

Table 4: Analysis of datasets and models used across submissions. Here, OD: organizer’s dataset only and PAM: publicly available models.

5 Discussion

In this paper, we present the findings from the MixMT shared task. We hosted two subtasks involving a code-mixed language i.e. Hinglish. Given the low-resource nature of the code-mixed languages (and the subtasks), the majority of the submissions rely on data augmentation either synthetically or from other external sources. The lack of dedicated pre-trained models for code-mixed languages pushed the teams to explore the available alternatives along with bold attempts to train the models from scratch. We posit several open challenges with code-mixed machine translation such as creating large-scale parallel data, efficient data augmentation strategies, and robust evaluation measures. The insights and findings from this task will be useful to future works on machine translation involving code-mixed and low-resource languages. They will broaden the horizon for works on multilingual machine translation.

References


Ondřej Bojar, Vojtěch Diatka, Pavel Rychlík, Pavel Straňák, Vit Suchomel, Aleš Tamchyna, and Daniel


Parth Chokhra. 2020. *Hindi to hinglish corpus*.


Asha Hegde and Hosahalli Lakshmaiah Shashirekha. 2022. Mucs@mixmt: indic-trans-based machine translation for hinglish text.


Tatoeba Project. 2022. Tab-delimited bilingual sentence pairs.


