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 (code-mixing of Hindi and English). Through both subtasks, we aim to address the challenges in building a real-world multilingual translation system involving code-mixed language as the source/target.

Similar to the recent events on code-mixed languages (Chen et al., 2022; Srivastava and Singh, 2021b; Patwa et al., 2020), the MixMT shared task has received participation and engagement with teams from across the globe. In total, we received registration from 38 teams. Throughout the competition, seven teams actively participated and submitted their system for the development phase, test phase, and human evaluation phase.

2 MixMT: Code-mixed Machine Translation

2.1 The two subtasks

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 Hindi language sentences are written in the Devanagari script whereas the target Hinglish language text is written 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 Roman script.

2.2 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.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 806–811
Abu Dhabi, December 7–8, 2022. ©2022 Association for Computational Linguistics

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 Dakshina dataset (Roark et al., 2020). They also perform the post-processing on the output generated from the fine-tuned model. For *CM2M* task, they finetune 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 toolkit[1], 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`

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

Table 1: Human-based evaluation policy for *M2CM* and *CM2M* subtasks. The underlined phrase highlights the center of attention for the corresponding rating.

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-translating 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

Team	*M2CM*		*CM2M*	
	R-L	WER	R-L	WER
Baseline	0.280	0.926	0.250	1.021
CNLP	0.238	0.926	0.330	0.88
DC	0.033	1.560	0.061	1.694
Gui	0.616	0.633	0.414	0.808
MUCS	0.358	0.760	0.550	0.647
NICT	0.462	0.792	0.528	0.715
SIT-NMT	0.57	0.547	0.629	0.607
UEDIN	0.579	0.561	0.621	0.624

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.

on both subtasks followed by UEDIN. Gui stood at the third position for *M2CM* task whereas MUCS is ranked third for *CM2M* subtask. Interestingly, MUCS and NICT get a consistent one score showing poor quality output consisting of lexical and grammatical mistakes. It further highlights the inefficacy of evaluation metrics for code-mixed natural language generation tasks as pointed out in several previous works (Garg et al., 2021; Srivastava and Singh, 2022).

Team	*M2CM*	*CM2M*
CNLP	2.1 ± 0.64	1.35 ± 0.74
DC	1.75 ± 0.71	1.55 ± 1.09
Gui	3.75 ± 1.20	1.8 ± 1.1
MUCS	1 ± 0	2.9 ± 1.51
NICT	1 ± 0	2.85 ± 1.30
SIT-NMT	3.85 ± 1.38	4.1 ± 1.07
UEDIN	3.85 + 1.53	3.75 + 1.16

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.

Further, we analyze the submissions based on the dataset and the models used in the experiment. In Section 2.4, we have highlighted the two criteria for the submission to be considered as constrained. In Table 4, we summarize the submissions based on these two criteria.

We observe that almost all the teams have used at least one external dataset for both subtasks with Gui's submission for *CM2M* subtask being the only exception. We attribute this behavior to the fact we designed both subtasks in a low-resource setting. The submissions by four teams (i.e., CNLP, DC, NICT, and UEDIN) are completely unconstrained for both subtasks as they are using an external dataset and training their own system from scratch.

Team	*M2CM*		*CM2M*	
	OD	PAM	OD	PAM
CNLP	✗	✗	✗	✗
DC	✗	✗	✗	✗
Gui	✗	✓	✓	✓
MUCS	✗	✓	✗	✓
NICT	✗	✗	✗	✗
SIT-NMT	✗	✓	✗	✓
UEDIN	✗	✗	✗	✗

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

Utsab Barman, Amitava Das, Joachim Wagner, and Jennifer Foster. 2014. Code mixing: A challenge for language identification in the language of social media. In *Proceedings of the first workshop on computational approaches to code switching*, pages 13–23.

Irshad Ahmad Bhat, Vandan Mujadia, Aniruddha Tammewar, Riyaz Ahmad Bhat, and Manish Shrivastava. 2015. Iiit-h system submission for fire2014 shared task on transliterated search. In *Proceedings of the Forum for Information Retrieval Evaluation*, FIRE '14, pages 48–53, New York, NY, USA. ACM.

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.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Vigneshwaran Muralidaran, Shardul Suryawanshi, Navya Jose, Elizabeth Sherly, and John P McCrae. 2020. Overview of the track on sentiment analysis for dravidian languages in code-mixed text. In *Forum for information retrieval evaluation*, pages 21–24.

Shuguang Chen, Gustavo Aguilar, Anirudh Srinivasan, Mona Diab, and Thamar Solorio. 2022. Calcs 2021 shared task: Machine translation for code-switched data. *arXiv preprint arXiv:2202.09625*.

Parth Chokhra. 2020. Hindi to hinglish corpus.

Raj Dabre. 2022. Nict at mixmt 2022: Synthetic code-mixed pre-training and multi-way fine-tuning for hinglish–english translation.

Raj Dabre and Eiichiro Sumita. 2021. Yanmtt: yet another neural machine translation toolkit. *arXiv preprint arXiv:2108.11126*.

Akshat Gahoi, Jayant Duneja, Anshul Padhi, Shivam Mangale, Saransh Rajput, Tanvi Kamble, Dipti Misra Sharma, and Vasudeva Varma. 2022. Gui at mixmt 2022 : English-hinglish: An mt approach for translation of code mixed data. *ArXiv*, abs/2210.12215.

Ayush Garg, Sammed Kagi, Vivek Srivastava, and Mayank Singh. 2021. Mipe: A metric independent pipeline for effective code-mixed nlg evaluation. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 123–132.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009.

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.

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

Abdul Rafae Khan, Hrishikesh Kanade, Girish Amar Budhrani, Preet Jhanglani, and Jia Xu. 2022. Sit at mixmt 2022: Fluent translation built on giant pre-trained models. *arXiv preprint arXiv:2210.11670*.

Faheem Kirefu, Vivek Iyer, Pinzhen Chen, and Laurie Burchell. 2022. The university of edinburgh's submission to the wmt22 code-mixing shared task (mixmt). *ArXiv*, abs/2210.11309.

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. 2020. *Neural machine translation*. Cambridge University Press.

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)*.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations*.

Sahinur Rahman Laskar, Rahul Singh, Shyambabu Pandey, Riyanka Manna, Partha Pakray, and Sivaji Bandyopadhyay. 2022. Cnlp-nits-pp at mixmt 2022: Hinglish–english code-mixed machine translation.

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.

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.

Paul Michel and Graham Neubig. 2018. Mtnt: A testbed for machine translation of noisy text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 543–553.

Carol Myers-Scotton. The matrix language frame model: Development and responses.

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. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53.

Xiao Pan, Mingxuan Wang, Liwei Wu, and Lei Li. 2021. Contrastive learning for many-to-many multilingual neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 244–258.

Parth Patwa, Gustavo Aguilar, Sudipta Kar, Suraj Pandey, Srinivas Pykl, Björn Gambäck, Tanmoy Chakraborty, Thamar Solorio, and Amitava Das. 2020. Semeval-2020 task 9: Overview of sentiment analysis of code-mixed tweets. In *Proceedings of the fourteenth workshop on semantic evaluation*, pages 774–790.

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

Lekan Raheem and Maab Elrashid. 2022. Domain curricula for code-switched mt at mixmt 2022. *arXiv preprint arXiv:2210.17463*.

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.

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.

Brian Roark, Lawrence Wolf-Sonkin, Christo Kirov, Sabrina J Mielke, Cibu Johny, Isin Demirsahin, and Keith Hall. 2020. Processing south asian languages written in the latin script: the dakshina dataset. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2413–2423.

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.

Vivek Srivastava and Mayank Singh. 2021a. 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.

Vivek Srivastava and Mayank Singh. 2021b. Quality evaluation of the low-resource synthetically generated code-mixed hinglish text. *INLG 2021*, page 314.

Vivek Srivastava and Mayank Singh. 2022. Code-mixed nlg: resources, metrics, and challenges. In *5th Joint International Conference on Data Science & Management of Data (9th ACM IKDD CODS and 27th COMAD)*, pages 328–332.

S Thara and Prabaharan Poornachandran. 2021. Transformer based language identification for malayalam-english code-mixed text. *IEEE Access*, 9:118837–118850.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

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.

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.

Kangyan Zhou, Shrimai Prabhumoye, and Alan W Black. 2018. A dataset for document grounded conversations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 708–713.