Samsung R&D Institute Philippines @ WMT 2024 Indic MT Task

Matthew Theodore Roque[a] **Carlos Rafael Catalan**[a] **Dan John Velasco**[a]
Manuel Antonio Rufino[a] **Jan Christian Blaise Cruz**[a,b]
[a]Samsung R&D Institute Philippines [b]MBZUAI
{roque.mt,c.catalan,dj.velasco,ma.rufino}@samsung.com
jan.cruz@mbzuai.ac.ae

Abstract

This paper presents the methodology developed by the Samsung R&D Institute Philippines (SRPH) Language Intelligence Team (LIT) for the WMT 2024 Shared Task on Low-Resource Indic Language Translation. We trained standard sequence-to-sequence Transformer models from scratch for both English-to-Indic and Indic-to-English translation directions. Additionally, we explored data augmentation through backtranslation and the application of noisy channel reranking to improve translation quality. A multilingual model trained across all language pairs was also investigated. Our results demonstrate the effectiveness of the multilingual model, with significant performance improvements observed in most language pairs, highlighting the potential of shared language representations in low-resource translation scenarios.

1 Introduction

This paper details our primary submission for the WMT 2024 Shared Task on Low-Resource Indic Language Translation. Our submission covers the following language pairs: English ↔ Assamese (en-as), English ↔ Mizo (en-mz), English ↔ Khasi (en-kh), and English ↔ Manipuri (en-mn). Our approach builds upon the methodology used in Samsung R&D Philippines' WMT23 entry (Cruz, 2023). We employed a standard sequence-to-sequence Transformer architecture (Vaswani et al., 2023), combined with data augmentation through backtranslation (Sennrich et al., 2016), noisy channel reranking (Yee et al., 2019), and additionally experiment with a multilingual model trained on all language pairs.

[b]Work done while at Samsung R&D Institute Philippines

2 Methodology

2.1 Environment

For preprocessing, training, and generation, we utilized PyTorch 2.0 and fairseq 0.12.2. All training was conducted on NVIDIA P100 GPUs.

2.2 Data Analysis

We used the Indic dataset provided from WMT 2023 for all language pairs. First, we conducted an exploratory data analysis for all the languages to see if there were noteworthy patterns that could guide us in our translation in the Indic and English languages. We used various methods in this data analysis such as finding N-most common words, generating N-grams, and histograms of lengths of sentences.

An interesting pattern emerged when generating the histograms of sentence lengths as seen in Figure 1. For the English-Mizo pair, the distributions almost completely overlap. However, for the English-Assamese, English-Khasi, and English-Manipuri pairs, the Indic languages generally exhibit slightly longer sequences. We hypothesize that these longer sequences may cause translation errors in the Indic to English language directions. The models might be driven to provide translations that are driven more by length alignment, and so may attempt to fill in additional tokens to produce longer sequences even if it may not necessarily be semantically accurate.

2.3 Data Preprocessing

We exclusively used the task dataset for all language pairs. For the parallel data, we first removed exact duplicates, then detokenized the text to correct spacing around punctuation. The statistics of parallel data are summarized in Table 1. Following this, we trained a BPE tokenizer (Sennrich et al., 2015), applied BPE tokenization, and binarized the data for use with fairseq. Each language pair

Proceedings of the Ninth Conference on Machine Translation (WMT), pages 735–741
November 15–16, 2024.

source↔target	Pairs	Words (source)	Words (target)	Vocab Size
en↔as	50,000	969,626	825,063	31,448
en↔kh	21,000	729,930	875,545	9,312
en↔mz	50,000	981,468	1,062,414	30,432
en↔mn	21,687	390,730	330,319	30,736

Table 1: Statistics of parallel training data. Note that "Words" refers to word count estimated using the wc command on the plaintext files.

source→target	Unfiltered Pairs	Filtered Pairs	Words (source)	Words (target)
en→as	2,624,715	279,956	3,200,053	3,444,809
en→mz	1,900,848	1,637,838	21,534359	26,367139
en→kh	160,128	19,358	363,441	490,257
en→mn	298,608	10,837	97,418	145,928

Table 2: Statistics of generated backtranslated parallel data. Note that "Words" refers to word count estimated using the wc command on the plaintext files.

has a shared vocabulary between English and the respective Indic language. The preprocessed parallel data was used to train our translation models. The same preprocessing steps were applied to the monolingual data for training the language models. As no monolingual data was provided for English, we used the combined English sides of the parallel data to train the English language model.

2.4 Augmenting Data with Backtranslation

Due to time and data constraints, data augmentation via backtranslation was applied only in the English-to-Indic direction. Backtranslated data was generated by translating the monolingual Indic data into English using the trained Indic-to-English models. After generating the backtranslations, we applied ratio-based filters (Cruz, 2023) to remove low-quality parallel data, filtering based on sentence length, token length, character-to-token ratio, pair token ratio, and pair length ratio. For more details, please refer to the original paper. The dataset statistics for the backtranslated data are presented in Table 2.

2.5 Model Training

For each of the four language pairs, we trained four models: two **Translation Models**, one for each translation direction, and two **Language Models**, one for each language. The specifics of these models are described in the following subsections. Three of these four models were combined for noisy channel reranking in one direction, as detailed in Section 2.7. Additionally, we experimented with a **Multilingual Model** using the same architecture as our translation models, but trained across all language pairs.

2.5.1 Translation Models

For the translation models (English→Indic, Indic→English), we trained encoder-decoder Transformer architectures (Vaswani et al., 2023) from scratch using parallel data. Separate models were trained for each language pair and for each translation direction. We used the large variant of the Transformer model with 213M parameters, training for 100,000 steps, with the first 10,000 being warmup steps (Gotmare et al., 2018), with a maximum of 8,000 tokens per step. The learning rates varied across language directions, as follows: en→as (9e-5), en→kh (5e-4), en→mizo (9e-5), en→mn (9e-5), as→en (5e-4), kh→en (5e-4), mizo→en (5e-4), and mn→en (5e-4). All other hyperparameters are detailed in Table 3.

These translation models were not only used as direct translation models but also served as channel translation models for noisy channel reranking, further discussed in Section 2.7.

2.5.2 Language Models

We trained monolingual language models for each language from scratch using the decoder-only component of the original Transformer architecture, as described by (Vaswani et al., 2023). We used the base variant of the Transformer, which contains 65M parameters. For the Indic language models (Assamese, Mizo, Khasi, Manipuri), we trained on the provided monolingual data. For the English language model, we concatenated the English side of the parallel data for training.

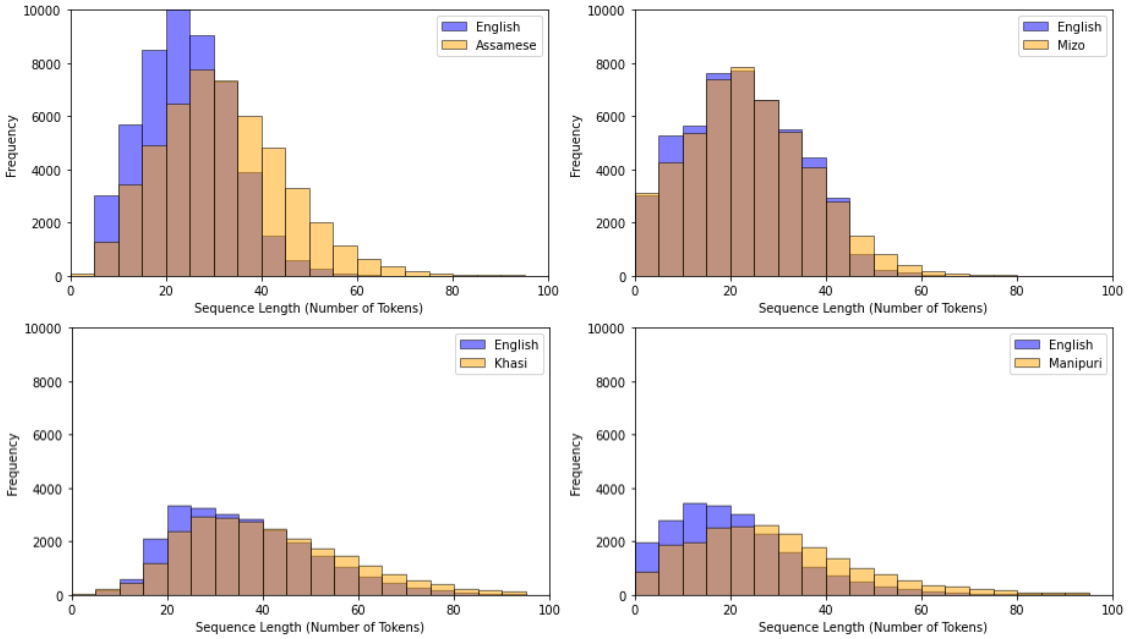

Figure 1: Histogram of Sentence Lengths

All models were trained using the Adam optimizer (Kingma and Ba, 2017) with $\beta_1 = 0.90$ and $\beta_2 = 0.98$. Training was conducted for a maximum of 100,000 steps, with the first 10,000 steps as a warmup (Gotmare et al., 2018). The learning rate started at 1e-7, peaked at 5e-4, and decayed following an inverse square root learning rate schedule. The batch size was set to 32,000 tokens, and a dropout rate of 0.1 was applied. These models were later used in noisy channel reranking, as detailed in Section 2.7.

2.6 Multilingual Model

We trained a large variant of the Transformer model with 213M parameters on all four language pairs, in both the English-to-Indic and Indic-to-English directions, following the approach of last year's entries (Zhang, 2023). Given the low-resource nature of each individual pair, we aimed to enable the language pairs to leverage cross-linguistic knowledge (Aharoni et al., 2019). The training process spanned 50,000 steps, with the first 5,000 steps serving as warmup (Gotmare et al., 2018; Neubig and Hu, 2018). We used 8 P100 GPUs for a maximum of 51,200 tokens per step and a learning rate of 1e-4. The remaining hyperparameters were consistent with those used in the other translation models as shown in Table 3.

Curriculum learning has been shown to improve generalization by introducing tasks progressively, allowing the model to build on prior knowledge (Wang et al., 2019). For our multilingual translation model, we aimed to apply a form of curriculum learning by training on different language pairs one at a time. We prepended source and target language tokens and trained the model sequentially on one language pair at a time. This structured training approach, inspired by Bengio et al. (2009), could help the model learn each language faster and transfer learned knowledge across language pairs. Similar to the benefits seen in multi-task learning by Niehues and Cho (2017), we hypothesized that this sequential training will enhance the model's ability to share representations across languages, ultimately leading to improved performance.

2.7 Noisy-Channel Reranking (NCR)

We experimented with Noisy Channel Reranking (Yee et al., 2019) to reevaluate and improve the translations. For brevity, we refer to this as NCR. This method utilizes three different models: a direct translation model (source→target), a channel model (target→source), and a monolingual language model (target only). These models are combined to rescore each candidate translation token during beam search decoding. The score for a candidate token $\hat{y}_i^{(T)}$ at timestep T is recomputed using the linear combination of the outputs from all three models:

Training Hyperparameters	
Vocab Size	31,960
Tied Weights	Yes
Dropout	0.3
Attention Dropout	0.1
Weight Decay	0.0
Label Smoothing	0.1
Optimizer	Adam
Adam Betas	β_1=0.90, β_2=0.98
Adam ϵ	ϵ=1e-6
LR Schedule	Inverse Sqrt
Batch Size	8,000 tokens

Table 3: Fixed hyperparameters for direct translation models.

$$P(\hat{y}_i^{(T)}|x;\hat{y}^{(T-1)})' = \frac{1}{t}log(P(y|\hat{x}^{(T-1)}) + \frac{1}{s}[\delta_{ch}log(P(x|\hat{y}^{(T-1)}) + \delta_{lm}log(P(\hat{y}^{(T-1)}))] \quad (1)$$

Here, t represents the length of the target sentence y, and s represents the length of the source sentence x, both of which serve as debiasing terms. The weights δ_{ch} and δ_{lm} control the influence of the channel model and the language model, respectively, on the final score.

2.8 Decoding and Noisy-channel Reranking Hyperparameter Tuning

We determined the optimal length penalty values by sweeping across four values: 0.5, 1.0, 1.5, and 2.0. This was done for each language direction, and the length penalty that resulted in the highest BLEU score on the provided test data was selected. The optimal length penalties for each direction are as follows: en→as (1.5), en→kh (2.0), en→mizo (1.0), en→mn (1.5), as→en (2.0), kh→en (1.5), mizo→en (0.5), and mn→en (2.0). These values were then used to tune the channel and language model weights for NCR.

We applied a similar approach to find the optimal values for the channel weight, δ_{ch}, and the language model weight, δ_{lm}. For the English-to-Indic models, we fixed δ_{ch} at 0.1 and varied δ_{lm} across 0.1, 0.2, 0.3, 0.4, 0.5, and 0.6. For the Indic-to-English models, we reversed the setup by fixing δ_{lm} at 0.1 and varying δ_{ch} across the same values.

These configurations were chosen due to time constraints, which limited our ability to perform more exhaustive evaluations of various combinations of δ_{lm} and δ_{ch}. Additionally, based on our initial evaluation of the translation models, the English-to-Indic models exhibited stronger performance, so we focused on evaluating their impact as channel models in NCR. Conversely, the Indic language models were trained on significantly more data than the English language model, making it essential to assess their influence in the reranking process.

3 Results and Discussion

In this section, we present the results of our experiments and provide an in-depth discussion of our findings. Tables 4 and 5 summarize the BLEU and chrF scores for each model and method on last year's test set, respectively. Table 6 summarizes the hyperparameters used for training the models.

3.1 Baseline

Our baseline consists of standard Transformer models trained from scratch for each language pair, without any backtranslation, length penalty tuning, noisy channel reranking, or multilingual setup. These models were trained using the parallel data provided, with a shared BPE vocabulary between English and each respective Indic language.

For the English-Khasi pair in particular, we set the target vocabulary size to 10,000, while for the other three language pairs, we retained a target of 32,000. Initially, we aimed for a 32,000 vocabulary size across all language pairs, but English-Khasi's vocabulary only reached approximately 20,000. Given that this was our worst-performing pair, we reduced the target size to 10,000, resulting in a BLEU score improvement of about 3 points.

As shown in Table 4, the baseline models performed adequately for most language pairs, with BLEU scores ranging from 4.2 (Khasi→English) to 34.1 (English→Manipuri). Notably, the English-to-Indic models generally outperformed the Indic-to-English models across all language pairs.

3.2 Data Augmentation Using Backtranslation

The number of pairs in the backtranslated data, as shown in Table 2, was greatly reduced after filtering. This reduction most likely stems from the poor performance of the Indic-to-English models used for backtranslation. These models may have produced low-quality translations, leading to a substantial number of backtranslated pairs being dis-

source→target	BLEU Scores				
	Baseline	w/ **BT Data**	**Tuned lenpen**	**NCR**	**Multilingual**
en→as	13.8	3.0	**14.0**	**14.0**	3.5
as→en	9.5	-	9.8	9.8	**10.1**
en→mz	29.7	18.6	29.7	28.5	**30.2**
mz→en	19.9	-	**21.3**	19.2	19.1
en→kh	8.8	6.5	9.6	10.2	**16.1**
kh→en	4.2	-	4.4	4.3	**7.9**
en→mn	34.1	1.1	35.2	34.9	**36.4**
mn→en	16.7	-	17.0	17.0	**21.4**

Table 4: BLEU Scores on WMT2023 Indic MT test data. The use of **BT Data** (training on backtranslated data) showed a decline in performance. The **tuned lenpen** (length penalty) generally improves BLEU score while **NCR** (Noisy Channel Reranking) yielded mixed results. The multilingual setting outperforms all other settings in all language pairs except en→as, where tuned lenpen and NCR showed the same score, and mz→en, where tuned lenpen was best.

source→target	chrF Scores				
	Baseline	w/ **BT Data**	**Tuned lenpen**	**NCR**	**Multilingual**
en→as	**26.1**	14.8	25.2	25.2	6.9
as→en	27	-	26.3	26.8	**28.4**
en→mz	44.4	34.5	44.4	42.8	**45.2**
mz→en	34.4	-	35.4	34.1	**35.6**
en→kh	29.9	27.9	30	30.4	**34.7**
kh→en	23.7	-	23.4	23.3	**27.8**
en→mn	45	11	43.8	43.5	**45.1**
mn→en	35.3	-	34.2	34.7	**44.2**

Table 5: chrF Scores from the WMT2023 Indic MT test data. The use of **BT Data** (training on backtranslated data) showed a decline in performance. The **tuned lenpen** (length penalty) and **NCR** (Noisy Channel Reranking) was tuned for the BLEU scores and yielded mixed results for chrF. The multilingual setting outperformed all other settings in all language pairs except en→as, where the baseline was best.

carded during the filtering process. The pairs that remained after filtering likely were still not of the best quality, which diminished the overall quality of the training. As a result, the models trained on this backtranslated data performed worse, as reflected in their BLEU scores in Table 4 and their chrF scores in Table 5.

3.3 Length Penalty

Our tuning of the length penalty, as shown in Table 6, revealed that most language directions, with the exception of English-to-Mizo and Mizo-to-English, preferred shorter translation sequences. As shown in Figure 1, the distribution of sentence lengths across the language pairs indicates a reasonable amount of overlap, though the Indic languages tend to have slightly longer sequences.

This preference for shorter sequences coincides with a known issue in Neural Machine Translation (NMT) models when handling long input sequences. NMT models typically rely on absolute positional encodings, which use fixed sine and cosine functions to assign vector positions. This approach tends to struggle with longer sequences due to the limitations of these fixed encodings, resulting in less precise representations as sentence length increases (Neishi and Yoshinaga, 2019). This is likely contributing to the models' difficulty in generating coherent longer translations, particularly for underperforming language pairs like English-Assamese and English-Khasi. As sequence length increases, the models are more prone to generating irrelevant or erroneous tokens, leading to a degradation in translation quality.

It is interesting to note that despite the Indic languages generally having longer sequences, a length penalty greater than one was found to be optimal for both directions, even in English-to-Indic translation. This indicates that the models may be biased

source→target	Hyperparameters		
	lenpen	ch_wt	lm_wt
en→as	1.5	0.1	0.2
as→en	2.0	0.6	0.1
en→mz	1.0	0.1	0.1
mz→en	0.5	0.4	0.1
en→kh	2.0	0.1	0.1
kh→en	1.5	0.2	0.1
en→mn	1.5	0.1	0.1
mn→en	2.0	0.3	0.1

Table 6: Final length penalty (lenpen), channel model weight (ch_wt), and language model weight (lm_wt).

towards shorter outputs across most language pairs, potentially as a safeguard against these positional encoding limitations. While this behavior aligns with our expectations for the English-Assamese pair based on its performance, the similar tendencies in the English-Khasi pair were more surprising, given the closer alignment of sentence lengths between these languages.

3.4 Noisy Channel Reranking

The BLEU scores obtained with NCR, as shown in Table 4, yielded mixed results. After tuning the length penalty, we observed that NCR improved performance for only one model out of eight, specifically English-to-Khasi. The chrF scores, as shown in Table 5, also indicate slightly improved performance with NCR solely for the English-to-Khasi pair. For all other language pairs, there was either no change in BLEU and chrF scores or a slight decrease. It is crucial to highlight that these results reflect the best combination of hyperparameters we identified; alternative hyperparameter settings would have resulted in even more pronounced variations in scores.

One notable finding is that the optimal language model weight was consistently around 0.1 across most language pairs. This suggests that the language model contributed minimally to improving translation quality. This issue may stem from either data quality or data quantity limitations. Investigating data quality issues would be valuable, but addressing them poses a significant challenge due to the already low-resource nature of the Indic languages. Further filtering could exacerbate data scarcity, making it difficult to maintain sufficient training data.

Conversely, the channel model weights were found to be more effective, with optimal values varying by language pair but generally falling in the mid-range. For the best-performing Indic-to-English pairs with NCR, specifically Mizo-to-English and Manipuri-to-English, the channel model weights were 0.4 and 0.3, respectively. These language pairs also had the best direct translation models and channel models, suggesting a stronger alignment between model quality and channel model effectiveness for these particular languages.

3.5 Multilingual Model

The multilingual model trained on all language pairs demonstrated considerable improvements over the baseline models, achieving the best performance in 6 out of the 8 language pairs. We attribute this success to the model's ability to learn from a broader context across all five languages, allowing for the creation of shared language representations. This approach is especially beneficial given the small size of the training datasets, as the multilingual model can leverage cross-linguistic knowledge to enhance translation quality.

However, due to time constraints, we were unable to explore the potential of using the multilingual model as a channel model within NCR. This remains a promising avenue for future research. Further studies could also investigate pretraining on the available monolingual data before fine-tuning for translation tasks. Additionally, fine-tuning the multilingual model for language modeling could further improve its utility in NCR, potentially acting out all three functions in NCR, leveraging shared linguistic knowledge on all languages and tasks, enhancing performance in low-resource language pairs.

4 Conclusion

In this paper, we presented our approach to the WMT 2024 Shared Task on Low-Resource Indic Language Translation. Our experiments demonstrated that the multilingual model trained across all language pairs performed exceptionally well, particularly in comparison to the baseline models, achieving the highest BLEU scores in 6 out of 8 language pairs and the highest chrF scores in 7 out of 8 language pairs. This indicates that leveraging shared language representations, especially when dealing with small datasets, can significantly enhance translation performance by utilizing cross-linguistic knowledge.

Despite some success, our attempts to improve results through data augmentation using backtranslation and noisy channel reranking yielded mixed outcomes. The poor quality of the Indic-to-English backtranslated data led to performance degradation, emphasizing the importance of both data quality and quantity in low-resource scenarios. Additionally, while noisy channel reranking provided benefits in isolated cases, its overall impact was limited, potentially due to suboptimal language model and channel model contributions.

The promising performance of our multilingual model suggests that further research could explore its integration within noisy channel reranking, possibly utilizing it as both a translation and a channel model. Additionally, future work should focus on enhancing the quality of backtranslated data and investigating pre-training strategies on monolingual data to boost the performance of low-resource language pairs.

References

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. *arXiv preprint arXiv:1903.00089*.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.

Jan Christian Blaise Cruz. 2023. Samsung R&D institute Philippines at WMT 2023. In *Proceedings of the Eighth Conference on Machine Translation*, pages 103–109, Singapore. Association for Computational Linguistics.

Akhilesh Gotmare, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. A closer look at deep learning heuristics: Learning rate restarts, warmup and distillation. *arXiv preprint arXiv:1810.13243*.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization.

Masato Neishi and Naoki Yoshinaga. 2019. On the relation between position information and sentence length in neural machine translation. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 328–338, Hong Kong, China. Association for Computational Linguistics.

Graham Neubig and Junjie Hu. 2018. Rapid adaptation of neural machine translation to new languages. *arXiv preprint arXiv:1808.04189*.

Jan Niehues and Eunah Cho. 2017. Exploiting linguistic resources for neural machine translation using multi-task learning. *arXiv preprint arXiv:1708.00993*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need.

Wei Wang, Ye Tian, Jiquan Ngiam, Yinfei Yang, Isaac Caswell, and Zarana Parekh. 2019. Learning a multi-task curriculum for neural machine translation. *ArXiv, abs/1908.10940*.

Kyra Yee, Yann Dauphin, and Michael Auli. 2019. Simple and effective noisy channel modeling for 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 5696–5701, Hong Kong, China. Association for Computational Linguistics.

Wenbo Zhang. 2023. Iol research machine translation systems for wmt23 low-resource indic language translation shared task. In *Proceedings of the Eighth Conference on Machine Translation*, pages 978–982.