CS779 Competition: Machine Translation System for India

Rythm Kumar
Roll No. 220925
rythmk22@iitk.ac.in
Indian Institute of Technology Kanpur (IIT Kanpur)

Abstract

This project focuses on neural machine translation for English–Hindi and English–Bengali as part of the Codalab competition. We experimented with GRU, attention-based GRU, bidirectional GRU, and Transformer models. The Transformer achieved the best results, outperforming all recurrent models on development and test sets. Our final submission reached Rank 10 on the leaderboard with BLEU 0.186, ROUGE 0.458, and chrF++ 0.449 on the test set. The study highlights the importance of attention, sequence length selection, and data preprocessing in building effective translation systems.

1 Competition Result

Codalab Username: r_220925
Final Leaderboard Rank on the Test Set: 10
chrF++ Score corresponding to the Final Rank: 0.449
ROUGE Score corresponding to the Final Rank: 0.458
BLEU Score corresponding to the Final Rank: 0.186
Total Number of Submissions in the Training Phase: 23
Total Number of Submissions in the Testing Phase: 5

Table showing the Number of Submissions made each Week during the Training Phase:

Week 1	Week 2	Week 3	Week 4	Week 5
2	4	4	4	5

2 Problem Description

The task in this project is to build translation systems for two Indian languages. The dataset contains two types of parallel sentences: English–Hindi and English–Bengali. Each example has an English sentence and its correct translation in one of the two target languages.

To solve the task, I trained two separate Transformer models. The first model is trained only on the English–Hindi sentence pairs, and the second model is trained only on the English–Bengali sentence pairs.

In the final test set, all sentences are in English, and each sentence has a tag that tells which target language it belongs to. During prediction, the English–Hindi model is used for the sentences assigned to the Hindi track, and the English–Bengali model is used for the sentences assigned to the Bengali track.

This setup works well because English, Hindi, and Bengali differ in grammar, sentence structure, and writing script. Training separate models allows each model to focus on the patterns of its specific language pair and generate better translations.

3 Data Analysis

The dataset is provided separately for two translation directions. The English–Hindi split contains 80,797 training pairs and 23,085 test sentences, while the English–Bengali split contains 68,849 training pairs and 19,672 test sentences. The corpus covers a wide range of topics.

Sentence Lengths

In the English–Bengali data, English sentences again average about 16 tokens (95% below 31), while Bengali translations are shorter, averaging 14–15 tokens (95% below 28). In the English–Hindi data, English sentences average about 16 tokens (95% below 33; 99% below 45), whereas Hindi translations average 18–19 tokens (95% below 37; 99% below 50).

Noise Characteristics

Minor noise is present: occasional mixing of English with Hindi or Bengali, stray punctuation, inconsistent spacing, and a very small number of empty or fragmented sentences. These issues affect only a small portion of the data.

Insights

Hindi translations tend to be longer and more descriptive, whereas Bengali translations are more compact. Longer English sentences generally lead to even greater expansion in both target languages. These differences support using separate models for the two language pairs.

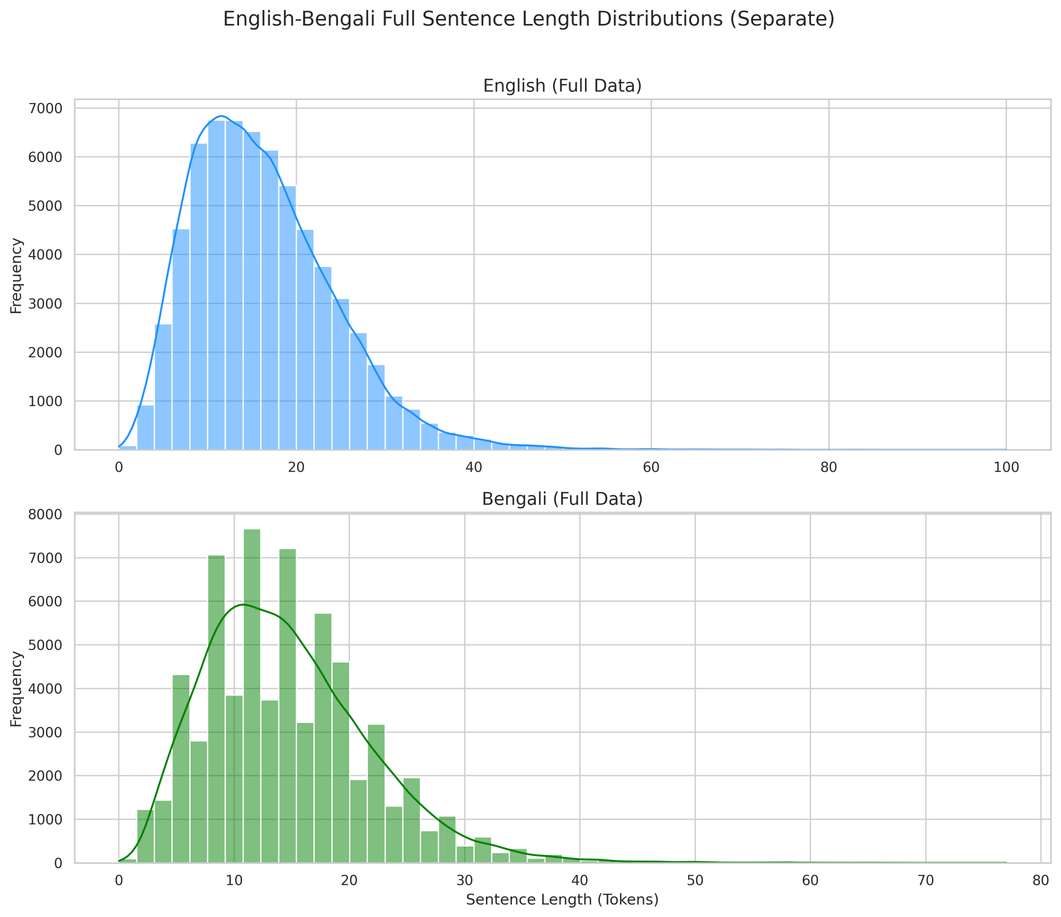

Figure 1: English-Bengali Sentence Length Distributions (Token Count)

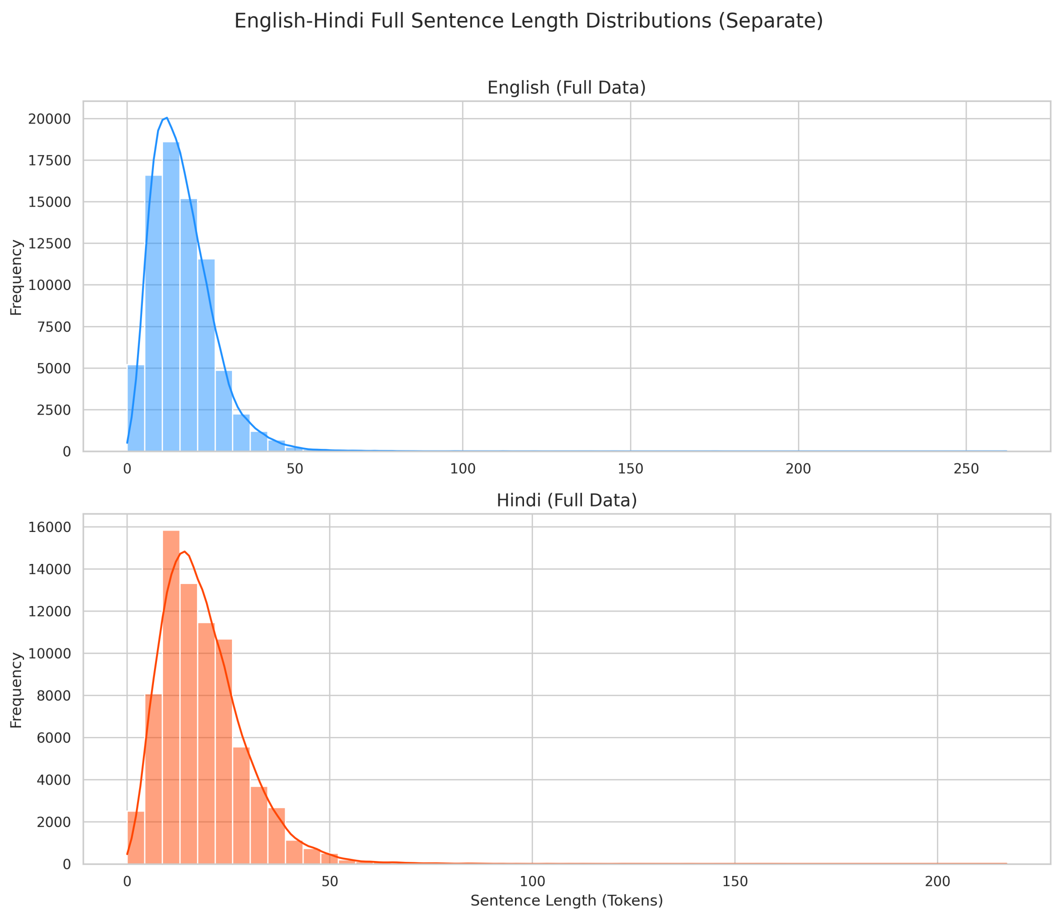

Figure 2: English-Hindi Sentence Length Distributions (Token Count)

4 Model Description

We developed the system in stages. Starting with GRU encoder–decoder models, we added improvements such as teacher forcing, attention, and bidirectional encoding. After understanding how far recurrent models could take us, we transitioned to a Transformer, which became our final and best-performing model.

4.1 Model Evolution Across Phases

The first model was a basic GRU encoder–decoder with hidden size 256. It produced reasonable translations for short sentences but struggled with longer ones because it lacked attention and could not capture long-range dependencies.

Next, we added scheduled teacher forcing to improve learning stability. This helped the model converge earlier, but the decoder still had difficulty generating coherent long sentences. We then introduced Luong attention. With attention, the decoder could focus on relevant encoder states at each decoding step, which improved alignment and overall translation quality. After that, we upgraded the encoder to a bidirectional GRU. This allowed the model to encode both forward and backward context, producing stronger sentence representations. A small projection layer was used to adapt the bidirectional hidden states for the unidirectional decoder.

Finally, we transitioned to a Transformer model. The Transformer uses multi-head attention, positional encoding, and parallel computation to model long-range dependencies more effectively. This architecture clearly outperformed all GRU-based models and became our final submission model for both English–Hindi and English–Bengali translation tasks.

4.2 Inspirations from Prior Work

The GRU-based sequence-to-sequence architecture followed the design of Cho et al. [2]. The attention mechanism drew from Bahdanau et al. [1] and Luong et al. [4]. The final Transformer architecture was inspired by Vaswani et al. [5], which introduced multi-head attention, positional encoding, residual connections, and masking.

4.3 Final Model Used for Submission

The final system was a Transformer encoder–decoder model. The encoder used 512-dimensional token embeddings with sinusoidal positional encoding and consisted of three layers of multi-head self-attention with eight heads, followed by a feed-forward network with inner dimension 2048. Each layer applied residual connections, layer normalization, and dropout (0.12).

The decoder also contained three layers with masked self-attention, cross-attention over encoder outputs, and a 2048-dimensional feed-forward network. Decoder outputs were passed through a linear layer to produce vocabulary logits, and the input and output embeddings of the decoder were tied to improve fluency and reduce the number of parameters.

Separate Transformer models were trained for English–Hindi and English–Bengali translation. During testing, each sentence was routed to the corresponding model based on its language pair.

4.4 Loss Function

The Transformer was trained using cross-entropy loss with label smoothing (0.1). Padding tokens were ignored in the loss calculation. All GRU-based models used negative log-likelihood loss. Across all models, we used the Adam optimizer and applied gradient clipping to prevent exploding gradients.

4.5 Inference Strategy

We used greedy decoding for inference, selecting the highest-probability token at each step. Greedy decoding produced stable results and avoided unnecessary computation. Beam search was also tested, but it increased inference time by approximately 6–8 times without providing meaningful improvements in translation scores. Decoding stopped when the `<eos>` token appeared or when the maximum sequence length was reached.

5 Experiments

5.1 Data Pre-processing

Both the source and target languages were pre-processed to make the data consistent before training. For English, the text was lowercased, digits and punctuation were removed, and tokenization was performed using spaCy. For Hindi and Bengali, English characters inside sentences were removed and the IndicNLP tokenizer was used. This was necessary because the dataset contained script mixing and noisy symbols that could affect translation quality.

Special tokens such as `<pad>`, `<sos>`, `<eos>` and `<unk>` were added. GRU-based models used the full vocabulary without applying a frequency threshold, while the Transformer used a minimum frequency of 2 to remove very rare tokens, reduce vocabulary size, and improve generalization.

All sentences were padded or truncated to a fixed length. GRU models used a sequence length of 20. For the Transformer, the sequence length was initially 20 and later increased to 45 after analyzing sentence-length distributions. Around 95–99% of sentences in both English–Hindi and English–Bengali pairs were below 45 tokens, so increasing the limit prevented information loss in longer inputs.

5.2 Training Procedure

All models were trained using the Adam optimizer. GRU-based models were trained for eight epochs, with each epoch taking roughly 6–7 minutes. Learning rates between 0.001 and 0.002 were used depending on the model variant. Teacher forcing with a scheduled ratio was used in some GRU models to stabilize the decoder. Gradient clipping with a maximum norm of 1.0 was applied throughout training.

The Luong Attention model and the BiGRU model followed the same training setup, with the BiGRU version additionally using a `ReduceLROnPlateau` learning-rate scheduler to improve stability when validation loss plateaued.

Transformer training was faster. After pruning the vocabulary using `min_freq = 2`, each epoch took about 4 minutes. The initial Transformer runs used 15 epochs, and based on validation improvements, the training was later extended to 45 epochs. Label smoothing with a factor of 0.1 was used to reduce overconfidence and improve generalization.

5.3 Hyperparameters

Different models required different hyperparameters. GRU-based models used a hidden size of 256, dropout between 0.3 and 0.4, batch size 50, and a sequence length of 20. The Luong Attention model included an attention layer for computing alignment scores at each decoding step. The BiGRU + Attention model used a two-layer bidirectional GRU encoder and a projection layer to map the 512-dimensional bidirectional output to the 256-dimensional decoder space.

The Transformer used a model dimension of 512, eight attention heads, three encoder layers, three decoder layers, and a feed-forward dimension of 2048, with dropout set to 0.12. These choices were inspired by the standard Transformer architecture but scaled down to match training-time and hardware constraints while maintaining strong performance.

6 Results

6.1 Results on Development Data

We evaluated all model variants on the development set using BLEU, ROUGE, chrF++, and the combined total score used in the competition. The GRU-based models showed similar performance even after adding teacher forcing, Luong attention, and a bidirectional encoder. In contrast, the Transformer achieved a clear improvement across all metrics.

Model	**BLEU**	**ROUGE**	**chrF++**	**Total Score**
GRU + Attention (no TF)	0.041	0.310	0.189	0.540
GRU + Teacher Forcing	0.045	0.310	0.204	0.559
BiGRU + Attention	0.051	0.286	0.223	0.560
Transformer (15 epochs)	0.159	0.427	0.407	0.993

Table 1: Development set results for different model variants.

The Transformer outperformed all GRU-based models, which motivated its use for the final submission.

6.2 Results on Test Data

Since the Transformer performed best on the development set, we used it exclusively for test predictions. Its training was extended to 45 epochs, which further improved its performance.

6.3 Discussion and Insights

The GRU-based models achieved comparable scores and did not exhibit major improvements despite architectural upgrades. Although attention, teacher forcing, and bidirectional encoding improved

Model	BLEU	ROUGE	chrF++	Total Score	Rank
Transformer (45 epochs)	0.186	0.458	0.449	1.093	10

Table 2: Test set results using the best Transformer model.

training stability, they did not significantly close the performance gap.

The Transformer achieved the highest scores across all metrics. Its multi-head attention allowed it to capture long-range dependencies, positional encodings helped it model word order, and label smoothing improved generalization. Increasing the number of training epochs further boosted performance. These factors made the Transformer the most effective architecture for this translation task.

7 Error Analysis

7.1 Errors Across Different Models

We evaluated all GRU-based models and the Transformer on both the development and test sets. The GRU variants (baseline GRU, GRU with teacher forcing, GRU with Luong attention, and BiGRU with attention) performed reasonably well on short and simple sentences but struggled with long or complex structures. For example, the sentence *"Many of the country's top places to eat are nestled into remote, rural countryside..."* often resulted in incomplete or shortened translations.

The Transformer consistently produced better translations, especially for long sentences and inputs with multiple clauses. It was also more robust to variability in sentence style and vocabulary.

7.2 Model-Level Error Patterns

Across the GRU models, we observed common issues such as dropping the end of long sentences, repeating words, and struggling with mixed-script inputs. Sentences containing embedded non-Latin script, such as *"Bahumati (Nepali:)"* or *"Bagvati parpradeshe (Nepali:)"*, frequently caused the GRU models to mistranslate surrounding phrases.

The Transformer handled such cases better, although it remained somewhat sensitive to noisy or irregular text.

7.3 Why Models Are Not Perfect

Even the strongest model (Transformer) exhibited several predictable mistakes:

- **Literal translations**: The output sometimes sounded overly direct or dictionary-like, e.g., for *"Crown Cave (Guanyan), A crown-like crag earns the hill its name."*
- **Missing words**: In long sentences, verbs or adjectives were occasionally omitted, such as in *"The park also features the Marine Natural Study Center..."*
- **Noisy input issues**: Irregular sentences like *"Same were given name of 'Yatuman' in Awesta."* produced awkward translations.
- **Script mixing**: English sentences containing Hindi/Bengali/Nepali words disrupted attention and sometimes distorted phrasing.
- **Hallucinations**: Rare cases of repeated or unrelated words, especially in sentences with unusual numeric patterns.

They errors could be reduced through script normalization, denoising the English text, adding multilingual pretrained models (mBART, mT5), using beam search with length penalties, or applying subword regularization. More training data would also help reduce hallucinations and missing-word errors.

7.4 Interesting Insights

English–Bengali translations were generally shorter and more condensed compared to English–Hindi. Mixed-script sentences were a common source of errors across all models. The Transformer handled long-distance dependencies and rare words much better than GRU models, especially in descriptive or informational sentences containing dates, measurements, or parenthetical information.

8 Conclusion

In this project, we explored several neural machine translation models for the English–Hindi and English–Bengali language pairs. We began with GRU-based encoder–decoder architectures and gradually introduced techniques such as teacher forcing, Luong attention, and a bidirectional encoder. These models helped us understand the strengths and limitations of recurrent networks, but their improvements on the development set remained small, especially for longer sentences and mixed-script inputs.

The Transformer model showed a clear advantage over all GRU-based models. Its multi-head attention, positional encoding, and parallel computation allowed it to capture long-range dependencies more effectively and handle diverse sentence structures. After tuning the sequence length based on corpus statistics and training for more epochs, the Transformer achieved the best performance on both the development set and the test set, placing us within the top ranks on the leaderboard.

Overall, we recommend using Transformer-based architectures for similar multilingual translation tasks, particularly when the data contains long sentences or inconsistent writing systems. Future directions include exploring pretrained multilingual models such as mBART [3] or mT5 [6], improving text normalization to handle noise, experimenting with beam search or length penalties, and analyzing attention patterns to better understand model behavior. These steps could further improve translation quality across all metrics.

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2015.

[2] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[3] Yinhan Liu, Jiatao Gu, Naman Goyal, Xiang Li, Sergey Edunov, Marjan Ghazvininejad, and Luke Zettlemoyer. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742, 2020.

[4] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

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

[6] Linting Xue, Noah Constant, Adam Roberts, Mihir Jha, Shashi Narang, and Colin Raffel. 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*, 2021.