Competition: Hindi to English Machine Translation System

Moksha Vora
20111035
{mkvora20}@iitk.ac.in
Indian Institute of Technology Kanpur (IIT Kanpur)

Abstract

The competition is for developing a Hindi to English Machine Translation System. The main model architecture pursued in the competition is Seq2Seq architecture. The various models used for the first three phases are the LSTM Seq2Seq model, LSTM Seq2Seq model with Bahdanau attention, and Bidirectional GRU encoder - GRU decoder with Bahdanau attention, respectively. The model used for the final phase is the same as the model used in phase 3 with different parameters. The rank achieved on the leaderboard is 8. The BLEU score and METEOR score obtained on the final phase model is 0.08982273607926991 and 0.348525713326374, respectively.

1 Competition Result

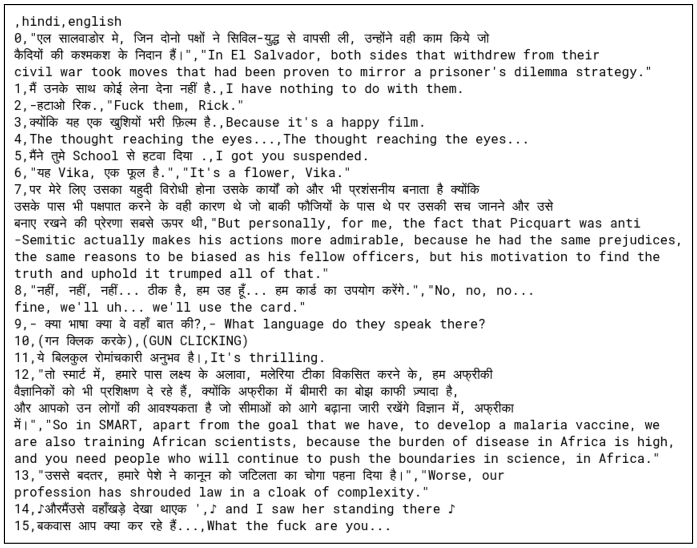
Codalab Username: M_20111035
Final leaderboard rank on the test set: 8
METEOR Score wrt to the best rank: 0.348525713326374
BLEU Score wrt to the best rank: 0.08982273607926991
Link to the CoLab/Kaggle notebook:
https://colab.research.google.com/drive/1CtGvL3-pTvORHlx23ZDxTVAge5tmh-6j?usp=sharing

2 Problem Description

The problem sentence of the competition is to create a Hindi-English Neural Machine Translating System. The effective model architecture for the NMT problem is the Encoder-Decoder model. Here, the encoder takes the source language sentence ('Hindi') and learns representation for it; then, it is fed to the decoder to produce the target language sentence ('English'). With the given train data set, we are required to train either the Seq2Seq model, Transformer model, or a hybrid model. The problem setting is such that it allows us to explore different encoder-decoder architectures along with different decoding strategies like greedy, beam search, etc. The competition lasts for 3 phases and a final phase. For the first three phases, a dev set is provided, which helps get an idea about our model's performance. The final phase presents a test set; the performance of our model on the test set ranks our position on the leaderboard. The evaluation metrics used for performance checking are Macro Average Smoothed BLEU score, Corpus level BLEU score with smoothing, and METEOR score.

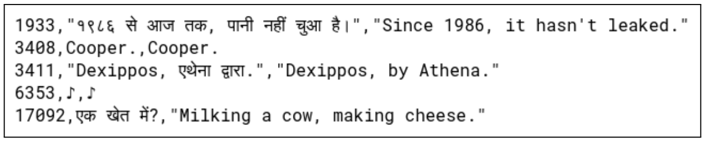
3 Data Analysis

The training dataset contains 102,322 Hindi sentences with their corresponding English translation. The test dataset contains 24,102 Hindi sentences. The dataset has been curated from the following publicly available sources: https://opus.nlpl.eu/ and http://www.opensubtitles.org/. As the dataset is built using publicly available content and websites, there tends to be some noise in the dataset, such as musical symbols, numbers, currency symbols, language mixing (Hinglish), and social media text. This can lead to a reduction in the performance of the model. The sample of the training dataset is shown in Figure 1.



```
,hindi,english
0,"एल सालवाडोर मे, जिन दोनो पक्षों ने सिविल-युद्ध से वापसी ली, उन्होंने वही काम किये जो
कैदियों की कश्मकश के निदान हैं।","In El Salvador, both sides that withdrew from their
civil war took moves that had been proven to mirror a prisoner's dilemma strategy."
1,मैं उनके साथ कोई लेना देना नहीं है.,I have nothing to do with them.
2,-हटाओ रिक.,"Fuck them, Rick."
3,क्योंकि यह एक खुशियों भरी फ़िल्म है.,Because it's a happy film.
4,The thought reaching the eyes...,The thought reaching the eyes...
5,मैंने तुमे School से हटवा दिया .,I got you suspended.
6,"यह Vika, एक फूल है.","It's a flower, Vika."
7,पर मेरे लिए उसका यहुदी विरोधी होना उसके कार्यों को और भी प्रशंसनीय बनाता है क्योंकि
उसके पास भी पक्षपात करने के वही कारण थे जो बाकी फौजियों के पास थे पर उसकी सच जानने और उसे
बनाए रखने की प्रेरणा सबसे ऊपर थी,"But personally, for me, the fact that Picquart was anti
-Semitic actually makes his actions more admirable, because he had the same prejudices,
the same reasons to be biased as his fellow officers, but his motivation to find the
truth and uphold it trumped all of that."
8,"नहीं, नहीं, नहीं... ठीक है, हम उह हूँ... हम कार्ड का उपयोग करेंगे.","No, no, no...
fine, we'll uh... we'll use the card."
9,- क्या भाषा क्या वे वहाँ बात की?,- What language do they speak there?
10,(गन क्लिक करके),(GUN CLICKING)
11,ये बिलकुल रोमांचकारी अनुभव है।,It's thrilling.
12,"तो स्मार्ट में, हमारे पास लक्ष्य के अलावा, मलेरिया टीका विकसित करने के, हम अफ्रीकी
वैज्ञानिकों को भी प्रशिक्षण दे रहे हैं, क्योंकि अफ्रीका में बीमारी का बोझ काफी ज़्यादा है,
और आपको उन लोगों की आवश्यकता है जो सीमाओं को आगे बढ़ाना जारी रखेंगे विज्ञान में, अफ्रीका
में।","So in SMART, apart from the goal that we have, to develop a malaria vaccine, we
are also training African scientists, because the burden of disease in Africa is high,
and you need people who will continue to push the boundaries in science, in Africa."
13,"उससे बदतर, हमारे पेशे ने कानून को जटिलता का चोगा पहना दिया है।","Worse, our
profession has shrouded law in a cloak of complexity."
14,♪औरमैंउसे वहाँखड़े देखा थाएक ',♪ and I saw her standing there ♪
15,बकवास आप क्या कर रहे हैं...,What the fuck are you...
```

Figure 1: Sample of Training dataset



```
1933,"१९८६ से आज तक, पानी नहीं चुआ है।","Since 1986, it hasn't leaked."
3408,Cooper.,Cooper.
3411,"Dexippos, एथेना द्वारा.","Dexippos, by Athena."
6353,♪,♪
17092,एक खेत में?,"Milking a cow, making cheese."
```

Figure 2: Some noisy and erroneous translations in training dataset

In the training dataset, there were many sentences whose English translations were incorrect. The sample of some noisy and erroneous translations in the training dataset is shown in Figure 2. There are many pairs where no Hindi sentence is given; both sentences are only English sentences. After removing such sentences and other noise from the dataset, the total Hindi-English pairs used for training the model were 101,231. The comparison between the training set and test set is as follows:

- The average length of the sentences after tokenization in the training set and test set is 10.7636 and 10.7518, respectively.
- The maximum sentence length in the training set and test set is 117 and 91, respectively.
- The length for which the maximum sentences are there in the training set and test set is 5 and 4, respectively.

The number of Hindi and English tokens after cleaning the data on the whole training set is 40884 and 29774, respectively. But, to train the model, the tokens with a frequency greater than equal to 2 are used to remove the rarely used words. Such Hindi and English tokens are 20453 and 18317, respectively. In the first three phases, the dataset is split into training set and validation set in 80% - 20% ratio. The number of Hindi and English tokens with frequency ≥ 2 in the training set after splitting is 18101 and 16413, respectively. The number of Hindi tokens in the test dataset is 19542.

4 Model Description

4.1 Phase 1:

In phase 1, the 2-layer LSTM Seq2Seq model architecture is used to design NMT as shown in Figure 3. An LSTM (Long Short-Term Memory) unit comprises a cell, an input gate, an output gate, and a forget gate. The cell remembers values over arbitrary time intervals, and the three gates regulate the flow of information into and out of the cell [1].

In the encoder, the source tokens are first passed through the embedding layer, followed by the dropout layer, and then to the LSTM cell. As a 2-layer LSTM structure is used, the output of the first LSTM layer is fed to the second LSTM layer, at the end providing the context vector containing the representation of the source tokens. As the teacher forcing method is used to train the model, the target tokens (after embedding and dropout layer) and the context vector are passed to the decoder LSTM cell. The output of these LSTM cells is passed through a linear layer to generate a single token.

In phase 1, around 8-9 models with the same architecture but different parameters such as embedding size, hidden states, dropout, learning rate, batch size, step length are trained to find the best possible set of parameters so that this set of parameters can be used in the following phases. The set of best parameters found are shown in Table 2.

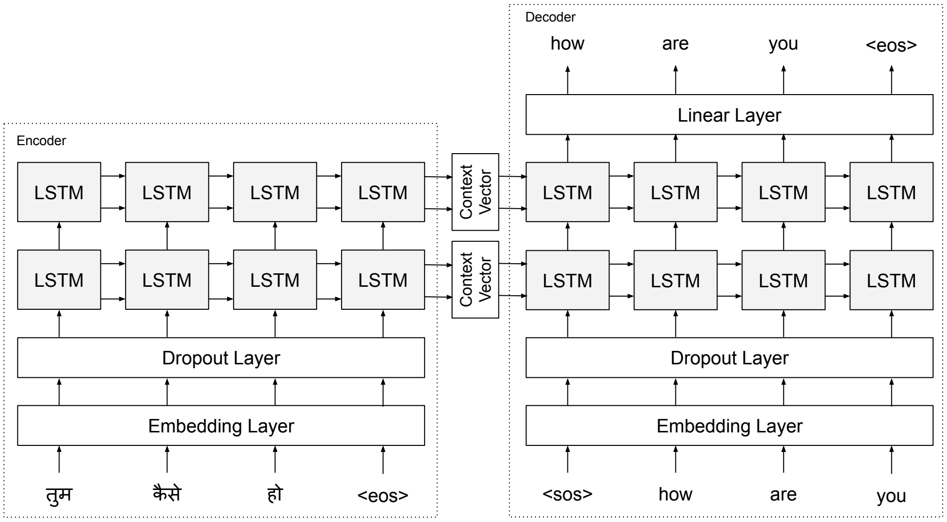


Figure 3: 2-layer LSTM Seq2Seq Architecture

4.2 Phase 2:

In phase 2, to improve the model prediction, the self-attention mechanism called Bahdanau attention or Additive attention is used. The LSTM is better than simple RNN unit at generating the context vector from the source tokens, but sometimes LSTM creates bad summary due to the large length of source string. To improve the context vector generated by LSTM, self-attention mechanism is used. The additive attention takes the previous hidden state and the encoder outputs of each LSTM unit and generates the context vector based on the weighted sum of the encoder outputs. The attention pooling f is instantiated as a weighted sum of the values [2]:

$$f(q, (k_1, v_1), \ldots, (k_m, v_m)) = \sum_{i=1}^{m} \alpha(q, k_i) v_i \in \mathbb{R}^v \tag{1}$$

where the attention weight (scalar) for the query q and key k_i is computed by the softmax operation of an attention scoring function a that maps two vectors to a scalar:

$$\alpha(q, k_i) = \text{softmax}(a(q, k_i)) = \exp(a(q, k_i)) \sum_{j=1}^{m} \exp(a(q, k_j)) \in \mathbb{R}. \tag{2}$$

For our case, k and v are the encoder outputs, and q is the previous hidden state. The phase 2 model architecture 2-layer LSTM Seq2Seq with Bahdanau attention mechanism is shown in Figure 4. The parameters of phase 2 are the same as the parameters used in phase 1.

4.3 Phase 3:

In phase 3, to explore different RNN units, GRU is used instead of LSTM. The GRU (Gated Recurrent Unit) unit comprises a candidate hidden state, a reset gate, and an update gate. Like LSTM, in GRU, the gates control the flow of previous information passing to the next unit. The model architecture in phase 3 is a 2-layer Bidirectional GRU encoder and 2-layer GRU decoder with Bahdanau

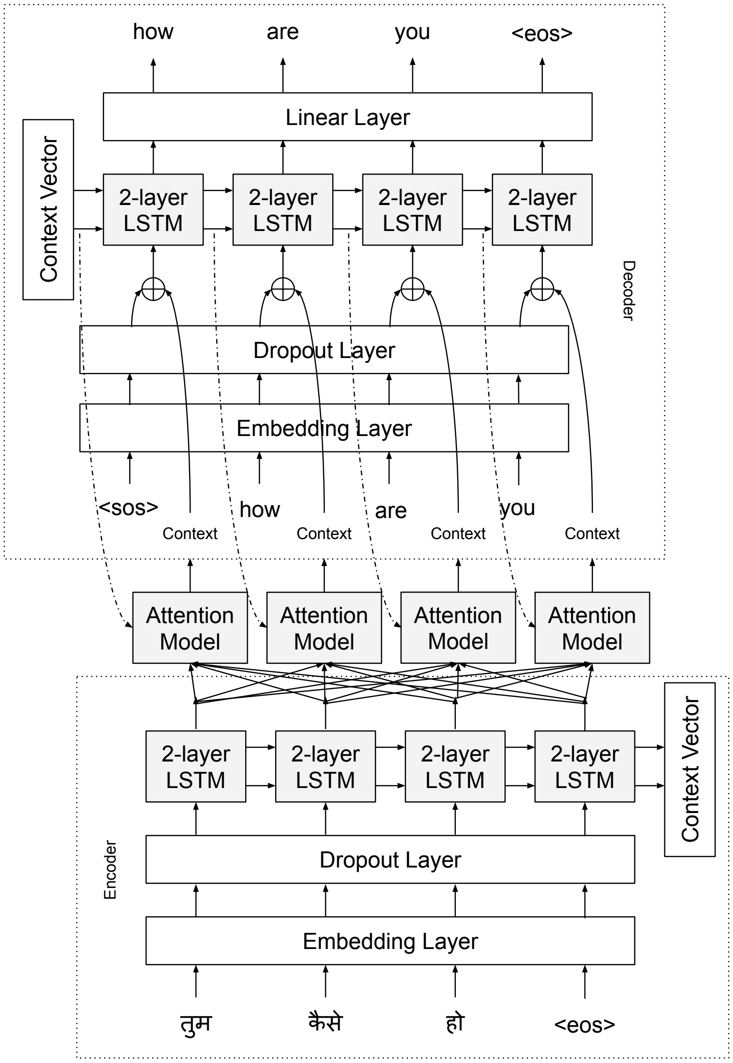


Figure 4: 2-layer LSTM Seq2Seq Architecture with Bahdanau Attention

Attention mechanism [3]. The bidirectional encoder helps better understand the sequence of tokens in both forward and backward directions as the hidden states of GRU units of both directions are concatenated. The combination of both bidirectional encoder with the self-attentive decoder led to a sudden improvement in the model's performance with the same parameters as in phase 2. The model architecture is the same as that in phase 2 with a difference that in place of 2-layer LSTM units, 2-layer bidirectional GRU units (shown in Figure 5) and 2-layer GRU units are replaced in encoder and decoder, respectively.

4.4 Final Phase:

The phase 3 model worked best until now, so in the final phase, the model parameters are played with to understand the importance of different parameters and the effect of every parameter on model performance. In the final phase, the model architecture used is a 2-layer BiGRU encoder and 2-layer GRU decoder with Bahdanau attention mechanism. The key insights obtained from the use of different parameter values are as follows:

- The small embedding size = 50, 64 works better than 125, 256. The reason for this could be the size of the dataset. As the dataset provided is small, if the embedding size is large, it may lead to a sparse embedding matrix.

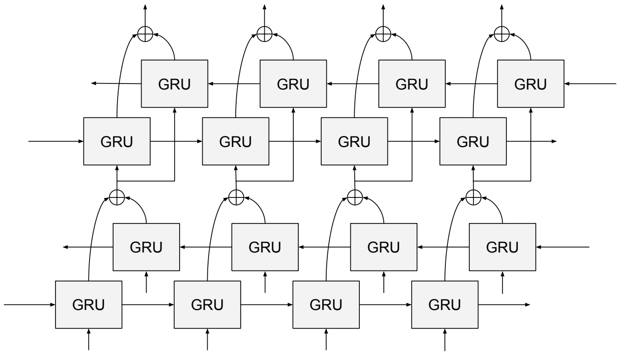


Figure 5: 2-layer BiGRU

- The step length also drastically affects the model's performance. The step length = 20 works best for our case. A larger step length can lead to a poor encoding of the information.
- Based on our dataset size and the length of sentences, the hidden units = 128 works best.
- Batch size = 64 works best; increasing the batch size leads to faster runtime but poor performance due to exploding gradients while decreasing the batch size increases the training time with not much improvement in the performance.

The information of the best model (Model 2) in final phase on test set is shown in Table 4.

4.5 Model Loss Function:

The loss function used in all the phases of the competition is the Masked Softmax Cross-Entropy Loss Function. The reason to use masking is to remove the effect of the padding (<pad> tokens) done at the data pre-processing stage over the loss function. As the decoder has to predict the target token, the softmax function is applied to get the probability distribution for the output tokens. After that, the cross-entropy loss is calculated for model optimization [2].

4.6 Decoding Strategy:

The decoding strategy tried are greedy decoding and beam search decoding with beam size = 8. The **greedy search** is the strategy in which the token with the highest prediction probability is predicted as the output token. While the **beam search** is an improved version of greedy search. It has a hyper-parameter named beam size, k. At time step 1, k tokens with the highest conditional probabilities are selected. Each of them will be the first token of k candidate output sequences, respectively. At each subsequent time step, based on the k candidate output sequences at the previous time step, we continue to select k candidate output sequences with the highest conditional probabilities from $k|v|$ possible choices where is $|v|$ is the vocabulary size [2].

The final model uses greedy decoding as the beam search decoding somehow does not perform well on the validation set (or maybe the implementation is not correctly done) and takes a long time to predict the target sentence. The prediction process continues till the <eos> token is encountered or maximum step length is reached. The method of prediction of output sequence used in phases 2, 3, and final phase is shown in Figure 6. The empty boxes in Figure 6 are either LSTM units or GRU units based on the model used. The prediction method used in phase 1 provides the context vector to the first RNN cell only. In every other phase, the context passed to the RNN cells is due to the self-attention mechanism.

5 Experiments

5.1 Data Pre-processing

To remove the noise described in the data analysis section, the simplest possible method is used. To process the Hindi sentences, 'regex' is used to substitute everything with space except the Hindi

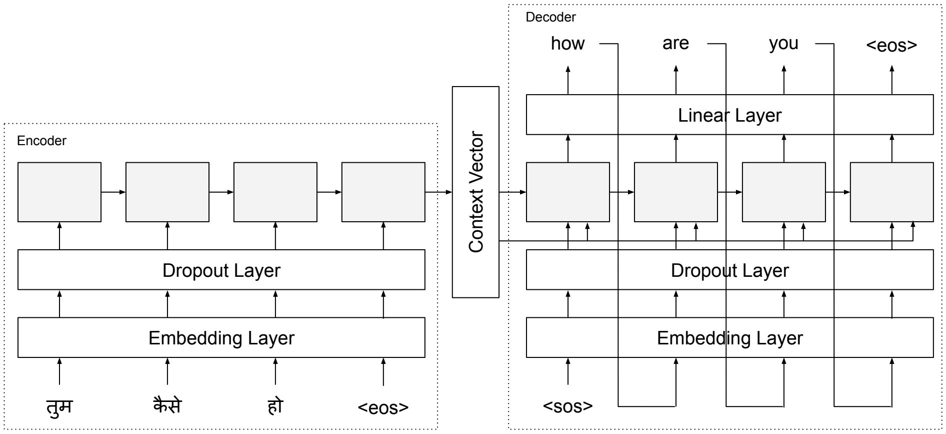


Figure 6: Prediction of target sentence

alphabets and space. Similarly, for the English sentences, ‘regex’ is used to substitute everything with space except the English alphabets and space. The case of letters is also changed to lowercase in the English sentences. One model is trained with data also consisting of four punctuation marks (.,!?). The tokens are generated by splitting the modified sentences with space. For training, the model tokens with frequency ≥ 2 are only used. The reason for performing this kind of data pre-processing is to improve the model performance by removing the unwanted symbols, numbers (as they were in less frequency). The next step of data pre-processing is to append <eos> token to every source sentence in the training dataset and then truncate or pad the sentences based on the step length. The sentences with length > step length are truncated, and those with length < step length are padded with <pad> tokens.

5.2 Training Procedure

The optimizer and learning rate used for different models is Adam optimizer and 0.001, respectively. Different learning rates such as 0.01, 0.005, and 0.0001 were tried, but the model performed the best for learning rate = 0.001. For every model, teacher forcing is used while decoding. The information of the training procedure of different models is given in Table 1.

Phase	**Model Architecture**	**Epochs**	**Train Time**
1	LSTM Seq2Seq model	300	3 hrs
2	LSTM Seq2Seq model (Bahdanau attention)	300	6 hrs
3	BiGRU Enc - GRU Dec (Bahdanau attention)	100	2 hrs
Test	BiGRU Enc - GRU Dec (Bahdanau attention)	200	$5\frac{1}{2}$ hrs

Table 1: Training procedure information of different models

5.3 Hyper-parameters

The dropout, batch size, and no. of layers used for different models are 0.25, 64, and 2. The chosen values of hyper-parameters dropout and batch size are decided after trying many other values. The hyper-parameter list of different models is shown in Table 2.

6 Results

The result of different models on the dev set in three phases of the competition is shown in Table 3. In the final phase of the competition, two models with the same architecture, i.e., BiGRU encoder - GRU decoder with Bahdanau attention but different hyper-parameters and different data pre-processing

Phase	Model Architecture	Embedding Size	Hidden Units	Step Length
1	LSTM Seq2Seq model	50	128	15
2	LSTM Seq2Seq model (Bahdanau attention)	50	128	15
3	BiGRU Enc - GRU Dec (Bahdanau attention)	50	128	15
Test	BiGRU Enc - GRU Dec (Bahdanau attention)	64	128	20

Table 2: Hyper-parameters of different models

techniques, are tried. In model 1, the punctuation marks (.,!?) were kept as part of tokens, while in model 2, the punctuation marks were removed. The parameters of the two models are given in Table 4. The result of the two models on the test set is shown in Table 5.

Phase	Model Architecture	BLEU Score	METEOR Score	Rank
1	LSTM Seq2Seq model	0.0039	0.179	18
2	LSTM Seq2Seq model (Bahdanau attention)	0.0178	0.173	20
3	BiGRU Enc - GRU Dec (Bahdanau attention)	0.0312	0.203	12

Table 3: Results of different models on dev set

Parameters	Model 1	Model 2
Embedding Size	128	64
Hidden Units	128	128
Layer	2	2
Dropout	0.25	0.25
Learning Rate	0.001	0.001
Epochs	150	200
Step Length	15	20
Batch Size	64	64

Table 4: Parameters for two models used for test set

As there is gradual improvement in the models, there is gradual improvement in the results of evaluation metrics. In phase 1, a 2-layer LSTM Seq2Seq model is used. After that, in phase 2, a self-attention mechanism is added, leading to improved performance. After exhausting many different values for model parameters, in phase 3, the decision to use the Bidirectional GRU encoder and GRU decoder with a self-attention mechanism lead to further improvement. Due to the Bidirectional encoder, the information from the source language (‘Hindi’) is encoded in a more precise manner as it encodes the sequence information from both forward and backward directions. Due to the self-attention mechanism, the decoding of the information to the target language (‘English’) improved as the self-attention mechanism uses the context generated by all the encoder outputs and the previous hidden state information to predict the word instead of just the encoder output by the last RNN cell. In the final phase, the model used is the same as the model in phase 3 but with different parameters. According to my understanding, in the final phase, model 2 performed better than model 1 due to the increased step length and small embedding size. Also, the reason behind it could be the elimination of punctuation marks in model 2.

7 Error Analysis

The BLEU [4] score calculates precision over the n-grams in the surface form. It matches the no. of n-grams currently present in the predicted sentence. It also takes the length of the predicted sentence and original sentence into consideration. It generally checks up to 4-grams. Whereas, METEOR [5] score

Model	BLEU Score	METEOR Score	Rank
1	0.0817	0.336	-
2	0.0898	0.349	8

Table 5: Result of different models on test set

only calculates harmonic mean over unigram recall and precision but uses stemming and synonyms matching, along with standard exact word matching. For the different models, the METEOR score of the model is better than the BLEU score because it only matches unigram, and the chances of unigram matching are much higher than bigrams and trigrams matching. The other reason for the METEOR score to be higher can be that it also considers stems and synonyms.

The reasons why models are not perfect:

- The first and foremost reason is the size of the dataset. Due to limitations in the size of the dataset, many key features like sequence, bigrams, trigrams, etc., cannot be learned by the model.
- The test dataset contains a total of 19542 unique tokens while the vocabulary size of the training dataset is 20543. But after comparing the vocabularies of both datasets, 6943 new tokens are found leading to OOV (Out-of-vocabulary) words. The prediction of such token will be <unk> token, leading to poor performance.
- The annotation errors in the dataset shown in Figure 2 is another reason for lower evaluation metrics scores. In some cases, the model prediction is correct according to the real translation, but the model's evaluation metrics score reduces due to the annotation errors.
- Finding the optimal parameter setting for any model is time-consuming and requires expertise in training the models. So, the model parameters chosen may not be optimal, hindering the performance of the model.

The analysis based on the true translation of the test dataset:

- As the maximum step length in the model is 20. The sentences longer than 20 lengths are poorly translated.
- The model translates small sentences very precisely, but it often predicts similar words (synonym or stem) and not the exact word. This can affect the BLEU score of the model.
- The number and punctuations in the sentences are not predicted as they are not part of the vocabulary. A future step can be to include them in the vocabulary.
- Proper nouns such as names of places and persons are not properly translated due to a small dataset.

8 Conclusion

The different phases in the competition helped improve the NMT model. Starting from the simple LSTM Seq2Seq model, after trying the self-attention mechanism and bidirectional models, the best model landed to be a 2-layer BiGRU encoder with a self-attentive 2-layer GRU decoder with a BLEU sccore of 0.089 and a METEOR score of 0.348. The self-attention mechanism helps the model mimic human brain actions. The bidirectional model helps better encode the source sentence into a context vector by concatenating the hidden units of both directions. The possible future work can be to use different pre-trained word embeddings like Word2vec [6], Glove [7], etc. Different decoding strategies like beam search, sampling, top-k sampling, etc., can be tried out. Scheduled sampling can be implemented instead of teacher forcing. And finally, to improve the model's performance, Transformer [8] models or hybrid models can be implemented. Above future works are to change the model setting ifself, but improvement can be done in the current model by doing Named Entity Recognition (NER) for predicting the proper nouns correctly.

References

[1] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, p. 1735–1780, Nov. 1997.

[2] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, *Dive into Deep Learning*. 2020. `https://d2l.ai`.

[3] Q. Li, X. Zhang, J. Xiong, W.-m. Hwu, and D. Chen, "Implementing neural machine translation with bi-directional gru and attention mechanism on fpgas using hls," in *Proceedings of the 24th Asia and South Pacific Design Automation Conference*, pp. 693–698, 2019.

[4] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, (Philadelphia, Pennsylvania, USA), pp. 311–318, Association for Computational Linguistics, July 2002.

[5] S. Banerjee and A. Lavie, "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments," in *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, (Ann Arbor, Michigan), pp. 65–72, Association for Computational Linguistics, June 2005.

[6] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[7] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.