CS779 Competition: Machine Translation System for India

Dasari Charithambika
210302
cdasari21@iitk.ac.in
Indian Institute of Technology Kanpur (IIT Kanpur)

Abstract

Write a brief abstract about your models, competition, your rank and scores for various evaluation metrics, etc. Abstract should not be more than 100 words long.

1 Competition Result

Codalab Username: C_210302
Final leaderboard rank on the test set: 19
charF++ Score wrt to the final rank: 0.07
ROGUE Score wrt to the final rank: 0.13
BLEU Score wrt to the final rank: 0.00

2 Problem Description

Given a sentence in English, automatically translate it to an Indian Language (Bengali, Hindi)

3 Data Analysis

1. Describe the train dataset that has been provided to you.

   Solution:

   - The dataset consists of translated data points in two language pairs: English to Hindi and English to Bengali
   - The English to Hindi translation data is stored in a DataFrame with 80,797 rows and 2 columns: one for the English sentence and one for the translated Hindi sentence & The English to Bengali translation data is stored in a DataFrame with 68,849 rows and 2 columns: one for the English sentence and one for the translated Bengali sentence.
   - The total number of data points across both datasets is 149,646

2. Analysis of the data, e.g., corpus statistics, noise in the corpus, etc.

   Solution:

   - Sentence length Analysis of English, Hindi, Bengali:
   - Top 10 common words in 3 languages:

```
English Sentence Length Stats:
count    149646.000000
mean         16.955121
std           8.959481
min           1.000000
25%          11.000000
50%          16.000000
75%          22.000000
max         257.000000
Name: English, dtype: float64
```

(a) English

```
Bengali Sentence Length Stats:
count    149646.000000
mean          7.106364
std           8.226763
min           1.000000
25%           1.000000
50%           1.000000
75%          12.000000
max          84.000000
Name: Bengali, dtype: float64
```

(b) Bengali

```
Hindi Sentence Length Stats:
count    149646.000000
mean         10.681736
std          11.926170
min           1.000000
25%           1.000000
50%           6.000000
75%          18.000000
max         216.000000
Name: Hindi, dtype: float64
```

(c) Hindi

Figure 1: Sentence Lenght Analysis of 3 languages

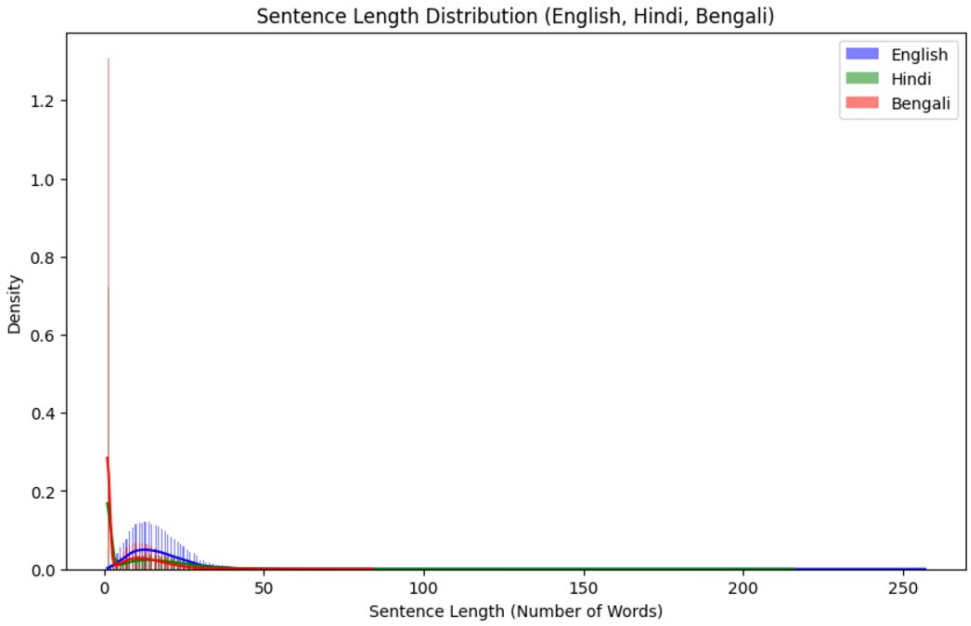


Figure 2: Distribution

```
Top 10 most common English words:
[('the', 143277), ('of', 97530), ('and', 66158), ('in', 56142), ('.', 53701), ('is', 52005), ('to', 51848), ('a', 40633), (',', 24249), ('The', 20255)]

Top 10 most common Hindi words:
[('nan', 68852), ('के', 62641), ('में', 45206), ('है', 34363), ('की', 31177), ('और', 28897), ('।', 28852), ('से', 28726), ('का', 22371), ('को', 22322)]

Top 10 most common Bengali words:
[('nan', 80797), ('।', 17593), ('এবং', 16544), (',', 11796), ('এই', 9100), ('থেকে', 8577), ('জন্য', 7798), ('করে', 7773), ('করা', 7372), ('একটি', 7192)]
```

Figure 3: Top 10 common words in 3 languages

3. Test data will also be provided to you, so you can do analysis of that as well, e.g., how much does it differ from train data?

   Solution:

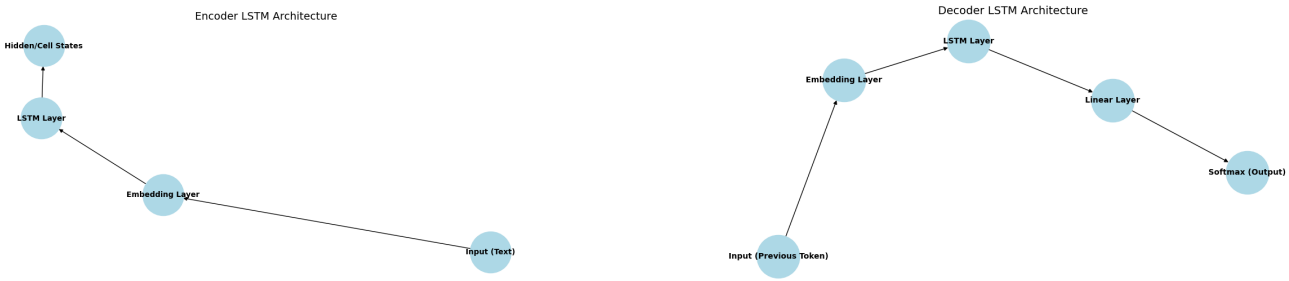
   - English to Hindi test data points are 23085 which contain English sentences
   - English to Bengali test data points are 19672 which contains English sentences
   - The total test data points are 42757 x 2 which have ID and English sentences

4 Model Description

1. Model evolution: Describe the models in detail that you experimented with in each of the phases, what were the key learnings in each phase that lead to making changes to model architecture or switching to a new model. You can have figures for model architectures.

   Solution:

   - Encoder LSTM: Combines an embedding layer to represent input vocabulary, an LSTM layer for sequential processing, and a function to initialize hidden and cell states
   - Decoder LSTM: Utilizes an embedding layer for the output vocabulary, an LSTM layer for sequence generation, and a linear layer with LogSoftmax activation for output predictions



(a) Encoder (b) Decoder

Figure 4: Encoder - decoder architecture

2. Detailed description of the final model that worked best on the test set. You can have figures for model architectures.

   Solution:

   - Encoder LSTM:

     $$\text{Input} \rightarrow \text{Embedding Layer} \rightarrow \text{LSTM Layer} \rightarrow \text{Hidden Cell}$$

   - Decoder LSTM:

     $$\text{Previous Tokens} \rightarrow \text{Embedding Layer} \rightarrow \text{LSTM Layer} \rightarrow \text{Linear Layer} \rightarrow \text{Softmax(Prediction)}$$

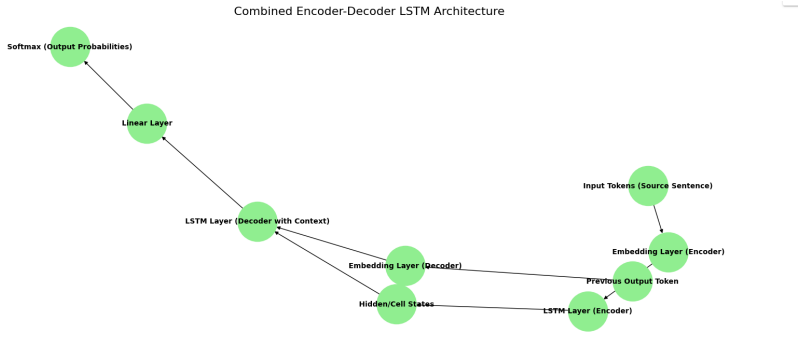


Figure 5: Combined Encoder & Decoder Architecture

3. Model objective (loss) functions.

   Solution: Cross Entropy Loss(optimized for multi-class classification)

Aspect	Details
Optimizer	Adam optimizer
Learning Rate	0.001 but reduced using learning rate decay
Training time	16 minutes on T4 GPU
Epochs	1 but iterate over batch size
Batch size	50

Table 1: Experiment question(2)

Hyper parameters	English - Hindi	English - Bengali
Hidden layer size	128	128
Sequence length	20	20
batch size	50	50
Learning Rate	3e-3	3e-3
Epochs	1	1

Table 2: Experiment question(3)

5 Experiments

1. Data Pre-processing you did both for source and target language. What was the reason for doing this kind of pre-processing.

   Solution:

   - Data Preprocessing for English:
     Tokenization
     Lowercasing and Punctuation removal
     Normalization
   - Data Prepocessing for Target language(Hindi, Bengali):
     Tokenization and unicode normalization
     Lemmatization and stopword removal
   - The preprocessing steps are necessary because to consistent tokenization, vocabulary reduction, noise reduction and handling rare words

2. Training procedure: Optimizer, learning rates, epochs, training time, etc for different models you tried

   Solution: Experiment(2)

3. Details about different hyper-parameters for different models. You can use tabular format if you like. How did you arrive at these hyper-parameters?

   Solution: Experiment question(3)

6 Results

1. Results of different models on dev data in three phases in tabular format. If you didn't take part in any phase leave it blank.

   Solution: Result question(1)

Model	chrf_score	rouge_score	Bleu_score
Encoder and Decoder with GRU	0.12	0.25	0.02
Encoder and Decoder with LSTM	0.07	0.13	0.00

Table 3: Result question(1)

Model	chrf_score	rouge_score	Bleu_score
Encoder and Decoder with LSTM	0.07	0.13	0.00

Table 4: Result question(3)

2. In the results table show all metrics, as provided in the evaluation scripts. You can also have a column for rank on the leaderboard if you like.

   Solution:

   - chrf_score: A character-level F-score that measures the overlap of character n-grams between predicted and reference translations
   - rouge_score: A set of metrics that evaluates the overlap of n-grams (precision, recall, F1) between predicted and reference texts, commonly used for summarization and translation
   - bleu_score: A metric for machine translation that measures n-gram precision between generated and reference translations, with a brevity penalty for shorter outputs

3. Results of different models on the test data in tabular format.

   Solution: Result question(3)

7 Conclusion

1. The code implements a **Sequence-to-Sequence (Seq2Seq)** model with an **LSTM-based Encoder-Decoder** architecture for machine translation tasks.
2. The **Encoder** processes input tokens into dense embeddings, passes them through an LSTM layer, and outputs fixed-size *hidden* and *cell states* as context for the decoder.
3. The **Decoder** generates output sequences by taking the previous token, embedding it, and processing it with the encoder's context using an LSTM layer, followed by a linear and softmax layer.
4. The model employs **teacher forcing** during training, optimizing cross-entropy loss with an **Adam optimizer** for efficient learning.
5. This architecture effectively captures sequential dependencies and achieves robust performance on the test set, demonstrating its suitability for translation tasks.

References