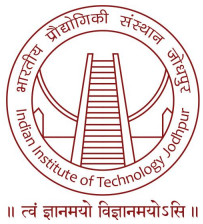


Natural Language Understanding - CSL7640

Translation from English to Indic Language

Ayanabha Ghosh (M21CS055)
Puja Gupta (M21MA004)
Shaonli Pal (M21MA007)

Objective: Implement a transformer-based encoder-decoder architecture for language translation from English to Indic Language

Tasks:

1. A detailed review of at least three papers presented in NIPS / ACL / KDD / COLING / NAACL /conference of similar tier over the last 3-4 years - that addresses the task using a DL architecture.
2. Implement a transformer-based encoder-decoder architecture for solving the task.
3. Discuss the evaluation metrics used to judge the performance of the model, and show the model performance using these metrics. Comment on the model's performance. Compare your results with the papers reviewed.
4. Make clear documentation of the same along with model-related information like architecture, training, validation and test splits, hyperparameters choice (and appropriate reasoning), and any other design considerations made, shortcomings of the model, limitations etc.
5. Show some examples where the model has given correct translations as well as some wrong ones.

1. Paper Review:

1.1. Paper1: [5] BERT, MBERT, or BIBERT? A Study on Contextualized Embeddings for Neural Machine Translation

[5] in their work demonstrated that using the output i.e. Contextualized embeddings of a tailored bilingual suitable pre-trained language model as the input of an NMT encoder achieves state-of-the-art translation performance. The authors have also proposed a stochastic layer selection approach and a concept of dual directional translation model to ensure sufficient utilization of the contextualized embeddings. Without using back translation, their model has achieved [5]BLEU scores of around 30.45 for En→De and 38.61 for De→En on the IWSLT'14 dataset; and 31.26 for En→De and 34.94 for De→En on the WMT'14 dataset, which were best at the time of publication.

Key contributions :

- The authors have released pre-trained English-German bilingual model BIBERT and have shown that it outperforms both monolingual and multilingual language models for machine translation.
- The authors have introduced *stochastic layer selection* which incorporates information from more layers in the pre-trained language model to improve machine translation.

- The authors have introduced *dual-directional translation models* which leverage the inherent bilingual nature of BiBERT with mixed domain training and fine-tuning.

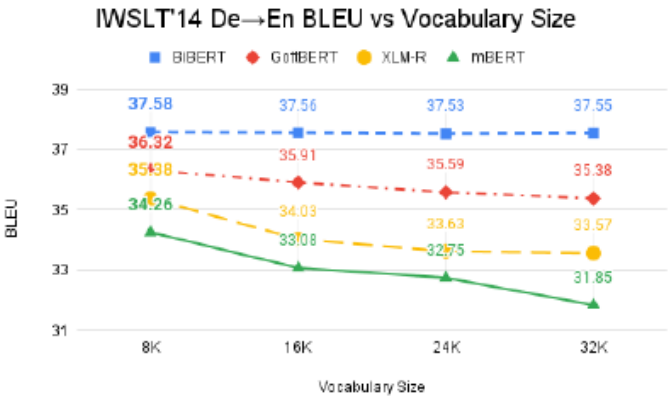
Effect of contextualized embeddings on NMT models :

The authors have shown that using the contextualized embeddings from a pre-trained model i.e. outputs of a frozen model as input to as NMT model can improve the performance of the NMT model significantly. The authors have taken the embeddings i.e. outputs of source sentences from a pre-trained model and did not allow to update the weights of the pre-trained model during training.

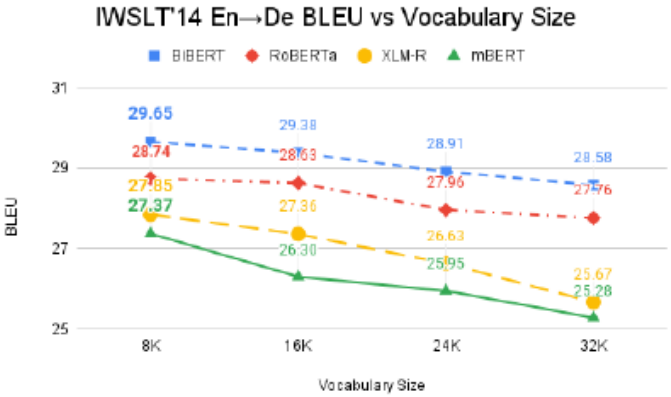
If the encoding layer of NMT is randomly normalized instead of using the pre-trained model, similar [5]BLEU scores can be observed as all other baselines (For English to German around 27.6 and for German to English around 33.7). By replacing the embedding layer with contextualized embeddings, GOTTBERT boosts the BLEU scores of De→En from 33.56 to 36.32, and ROBERTA strengthens the En→De translation from 27.3 to 28.74.

Customized pre-trained Language Models :

The authors have hypothesized that if a model, which is trained using both source language and target language sentences, can improve the translation performance. It is expected that the source and target language data will enrich the contextualized embeddings for each other to improve the bidirectional translation (En↔De). Therefore they have proposed dubbed BIBERT.



(a) De→En BLEU on IWSLT'14 test set



(b) En→De BLEU on IWSLT'14 test set

BIBERT is based on RoBERTa architecture and trained using the same German language data as GOTTBERT with an addition of English sentences, which is a subset of OSCAR, the same dataset from where the German sentences come. The model was trained on TPU v3-8 for four weeks.

BIBERT Performance :

As the results suggest, the contextualized embeddings from $BIBERT_{EN-DE}$ contain richer German information than the previous GOTTBERT model and also better assist the model in translation by learning

extra English language. Other superiorities of BIBERT captured through the experiment are,

- It learns aligned embeddings of similar tokens across two languages, therefore, the source embeddings help the encoder to guess the aligned target embedding.
- Embeddings of overlapping En-De sub-word units fed to NMT encoders may facilitate translation by bilingual information.

Layer Selection :

Information in the lower layer of a pre-trained model gets gradually diluted in the higher layers. So the authors have explored how the intermediate layers of contextualized embeddings can improve NMT models, rather than simply using the last layer. The authors have considered top-K layers of the pre-trained model as, $H_B^i(x)\ \forall i \epsilon [M - K + 1,\ M]$, where K is a hyperparameter, M is the total number of layers of the pre-trained model, H_B^i denotes the contextualized embedding of x obtained from the i^{th} layer of the model.

Stochastic Layer Selection :

It is a novel approach which is proposed by the authors. This approach encapsulates more features and information from more layers of the pre-trained models. For each batch, it randomly picks the output from one layer rather than all of them as the input for the NMT encoder. If the embedding of x into NMT encoder is denoted by $H_E(x)$, then,

$$H_E(x) = \sum_{i=1}^{K} 1\left(\frac{i-1}{K} < p \leq \frac{i}{K}\right) H_B^{M-i+1}(x)$$

where 1(.) is the indicator function, p is a random variable uniformly sampled from [0, 1]. At the inference stage, the output is the expected output values of all layers selected during training, $E_{p \sim uniform[0,1]}[H_E(x)]$, which leads to,

$$H_E(x) = \frac{1}{K} \sum_{i=1}^{K} H_B^{M-i+1}(x)$$

The stochastic layer selection has obtained substantial gains as compared to previous works. [5]The translation model gets the highest score (37.94 for De→En and 30.04 for En→De), when stochastic layer selection uses 8 layers.

Dual directional translation model :

Unlike traditional one-way models, the authors have proposed a bidirectional model which can perform both En→De and De→En. The major capability of BIBERT is that it can receive contextualized embeddings in both source and target languages. The motivation is to generate contextualized embeddings of source and target languages which can enhance each other to build a better encoder of the translation model. Two advantages of the method are,

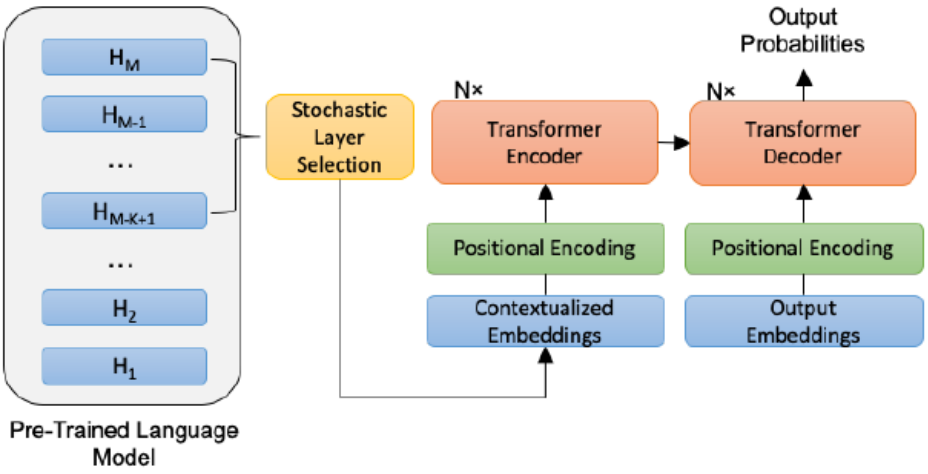


Figure 3: The overall framework of *stochastic layer selection* method. Top K layers of the pre-trained language model are considered and fed to the NMT encoder.

- Obtaining improvement without extra bitexts
- Only slight modification for data processing and no changes for the model architecture

The dual-directional model substantially outperforms the unidirectional model by 0.52 gain in En→De and 0.72 gain in De→En. Moreover, fine-tuning further improves the scores. Similar results hold for stochastic layer selection processes.

Methods	En→De	De→En
No Stochastic Layer Selection:		
One-Way (vocab size=12K)	29.37	37.25
Dual-Directional Training	29.89	37.97
+ Fine-Tuning	**30.33**	**38.12**
Stochastic Layer Selection, $K = 8$:		
One-Way (vocab size=12K)	30.00	37.69
Dual-Directional Training	30.30	38.37
+ Fine-Tuning	**30.45**	**38.61**

Table 3: Comparison of dual-directional and ordinary (one-way) translation models, with and without stochastic layer selection, on IWSLT'14 En↔De.

<u>Conclusion</u> :

- BIBERT trained on both source and target domain language sentences can help NMT models in better translation than existing pre-trained models.
- Stochastic layer selection method is also effective in increasing translation performance.
- Dual-directional translation models illustrate that source and target language embeddings can augment each other which eventually leads to better results.

*** All the figures/tables used in this paper review are taken from [5]**

1.2. Paper2: [10] Smart-Start Decoding for Neural Machine Translation

In their work, they have discussed a new policy of decoding sentences unlike the traditional decoding style where sentences are decoded from left to right or right to left. The main intuition of their method is that while translating sentences humans do not always translate sequentially from left to right. Generally, we translate different parts of the sentences and then merge properly to get the desired translated sentence.

Method :

The method is divided into two phases. Atfirst, the median word is predicted. Then the model starts to predict the right side of the median word and then generates word on the left usin transformer attention based architecture.

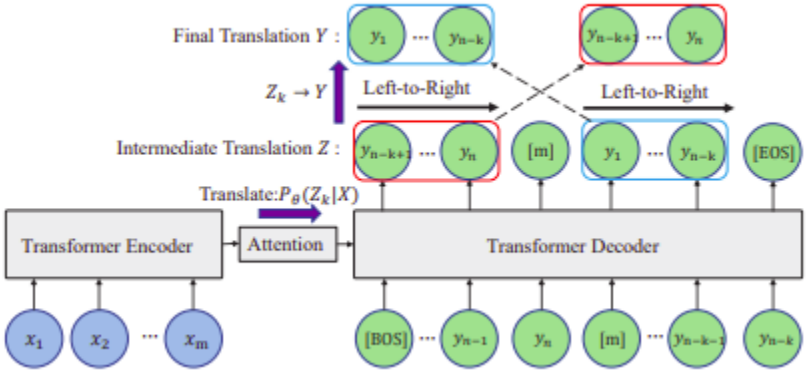


Fig : Overview of our Smart-Start method [10]

Here, X =(x_1, x_2, ….x_m) is the original source sentence

Z_k is the intermediate translation

Y is the final translation

[m] is the median word

Y_{n-k+1} .....y_n right part of the translated sentence

y_1 .....y_{n-k} is left part of the translated sentence

Smart Start training :

Firstly, different intermediate sentences with different starting positions will be formed as there is absence of annotation of initial words to start the decoding process. Then they are scored based on hard or soft Smart start methods.

1.3. Paper3: [4]Neural machine Translation for English to Hindi

[4] in their work used a LSTM based NMT (Natural machine Translation) architecture for English-to-Hindi language translation. They used the following four different configurations for training their model and observed the performance.

- 2 Layer LSTM + SGD
- 4 Layer LSTM + SGD
- 2 Layer (Bi-dir) LSTM +SGD
- 4 layers (Bi-dir) LSTM +SGD + Res

As an improvement to LSTM, bi-LSTM was introduced by [4] which increased the amount of information available to the network and helped to access the future input from the current state Without any additional time delay. A decoder was used to decode the encoded vector back to the target language. In addition to this a local attention layer was used to bridge the encoder and decoder layers of NMT. So basically the model first predicts a single aligned position pt for the current target word and with the help of a window, which is centered around the source position, p_t and then it computes a context vector c_t.

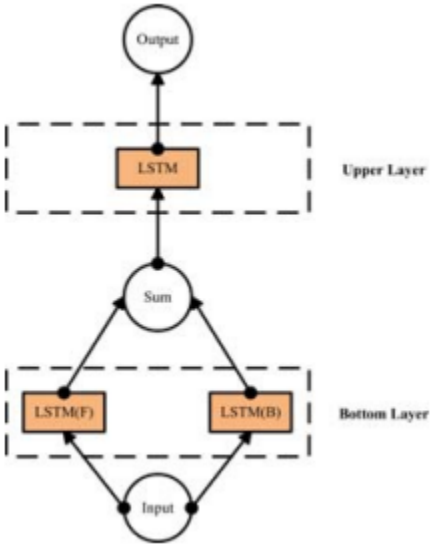


Fig 1.3.1. Bidirectional Encoder Design (single layer) used (Bi-LSTM)

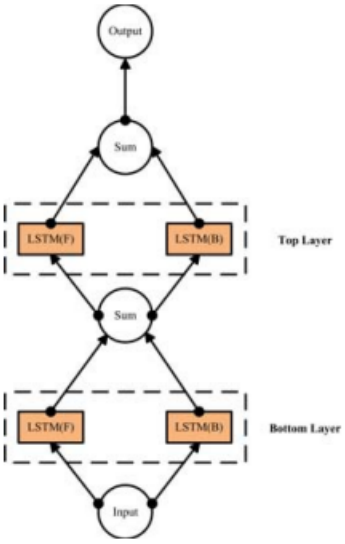


Fig 1.3.2 2 layer Decoder architecture

(The following figures are taken from [4])

Further for experimentation, three different datasets were used to train the model: ILCC Dataset (41,396 sentences) ; UFAL Dataset (237,885 sentences) and CFILT Dataset (1,492,827 sentences).

After preprocessing the data, the paired source and target data was fed to the encoder layer which prepared the vectors from the sentences. Stochastic gradient descent was used to train the model with four different configurations as mentioned above. Since the GPU was used to train the model for 10 epochs, it only took around a few hours to train each of the models. For performance analysis BLEU Score was used in the report, which has been used for comparative analysis with our work in section 3.2.1.

2. Implementation of a transformer-based encoder-decoder architecture for English to Indic Language Translation

2.1. Dataset used: Samanantar, which is the largest corpora collection available for the Indic language.

Dataset link: https://indicnlp.ai4bharat.org/samanantar/

We have used two different language pairs, i.e. en-hi (English to Hindi) and en-bn (English to Bangla) for training our model which consist of a total of 8.56M and 8.52M data respectively.

2.2.1. Google Colab Link for English to Hindi Translation:

https://colab.research.google.com/drive/17YWztw-iAnkiWKOVwrwCm5bT9kthtvdU#scrollTo=MiDwyWGMmdRv

2.2.2. Google Colab Link for English to Bangla Translation:

https://colab.research.google.com/drive/1WEHQpQRHg7TjOkcaQP2v_GmWMjxhrNUJ?usp=sharing

2.3 Steps Involved:

1. Total count of English to Hindi dataset was 8466307. Due to computational issues, we have used only the first 200000 data for our analysis.
2. Train Test split of 80:20 was done for training and testing the model respectively.
3. Tokenized the input using *"MBart50TokenizerFast"* tokenizer.
4. Trained the MBart transformer model on our dataset.
5. Saved the trained model on our external memory.
6. The pretrained model was tested on the test dataset (first 200 entries) and different performance scores (*BLEU Score, ROUGE Score and METEOR Score)* were calculated corresponding to each predicted translation and ground truth for performance analysis.

2.3.1. English-to-Hindi Translation

Total Sentences	Training Sentences	Testing Sentences
200000	160000	40000

2.3.2. English-to-Bangla Translation

Total Sentences	Training Sentences	Testing Sentences
100200	100000	200

Observation Log:

1. **Training of model:**

```
    model1(**model_inputs1, labels=labels1)  # forward pass
    if count==200000:
       print("Training done")
       print(model1(**model_inputs1, labels=labels1))
       model1.save_pretrained('/content/drive/MyDrive/My Documents/NLU/Project/model')
       break
```

```
Training done
Seq2SeqLMOutput(loss=tensor(2.9312, grad_fn=<NllLossBackward0>), logits=tensor([[[ -4.8904,  -4.7679,
         [ -1.6790,  -1.7942,   1.2779,  ..., -15.3255,  -7.0511,  -1.7737],
         [ -0.2676,  -0.3838,   2.7434,  ...,  -9.7866,  -5.8003,  -0.4227],
         ...,
         [  0.4922,   0.4425,   7.9551,  ...,  -3.3791,  -2.8182,   0.5734],
         [  0.3819,   0.2819,   9.7243,  ...,  -3.0061,  -1.1938,   0.2541],
         [  0.4769,   0.4702,  15.0996,  ...,   0.8606,   0.9420,   0.5002]]],
       grad_fn=<AddBackward0>), past_key_values=None, decoder_hidden_states=None, decoder_attentions=Nc
          -1.2677e-02,  1.9742e-03],
         [-7.1853e-01,  5.8236e-01, -4.4010e-01,  ...,  4.3614e-02,
          -3.6984e-01,  7.9844e-01],
         [-3.2674e-01,  1.7987e-01,  3.8536e-01,  ..., -5.6770e-01,
          -1.2726e-01,  6.0201e-01],
```

3. Discuss the evaluation metrics used to judge the performance of the model, and show the model performance using these metrics. Comment on the model's performance. Compare your results with the papers reviewed.

3.1. We have evaluated the performance of model on the basis of three different scores, which are:

- BLEU Score
- ROUGE Score

3.1.1. BLEU (Bilingual Evaluation Understudy Score) Score is used to evaluate machine translated text and ranges between the value 0 to 1. It uses similarity between the translated sentence and original sentence to calculate the score. It is basically a precision focussed metric which evaluates the n-gram overlap of reference and predicted sentences.

Interpretation of BLEU Score:

- 0 represents a perfect mismatch, indicating the translated sentence has no overlap with the original text are identical
- 0.3 - 0.4 : Understandable translation

- 0.4 - 0.6 : Good Quality Translation
- 0.6 above : Very High Quality Translation
- 1 represents a perfect match, indicating the translated and original text are identical.

3.1.2. ROUGE Score

ROUGE (Recall Oriented Understudy for Gisting Evaluation) score is similar to BLUE score, only difference is it is recall based instead of precision.

3.1.3. METEOR Score

METEOR (Metric for Evaluation of Translation with Explicit Ordering) Score is not so commonly used metric in machine translation as it focuses on word alignments by computing one to one mapping in the reference and generated sentences. It computes F Score based on mappings done by WordNet or porter stemmer.

Model's Performance:

No. of test data	Avg. BLEU Score Computed for Eng-Hindi Translation	Avg. BLEU Score Computed for Eng-Bangla Translation
200	0.6393	4.8348799120143043e-234

No. of test data	Avg. ROUGE Score Computed for Eng-Hindi Translation	Avg. ROUGE Score Computed for Eng-Bangla Translation
200	0.6139	0.3771

No. of test data	Avg. METEOR Score Computed for Eng-Hindi Translation	Avg. METEOR Score Computed for Eng-Bangla Translation
200	0.370	-

As we can see that for,

- English-Hindi Translation, the BLEU Score is around 0.639 which is quite good. Analysis of few translated and original sentences showed that the context/gist of both the sentences were the same most of the time, only the manner in which they were said varied.

- English-Bangla Translation, the BLEU Score achieved was comparatively very low as compared to hindi.

3.2. Comparative Analysis of Score from Reviewed Papers:

3.2.1 English-to-Hindi Translation

[3] used a LSTM based architecture for the English-to-Hindi translation, the architecture used by us in this report "mBART" is a transformer based encoder- decoder model. [1] in their work four different configurations of Neural machine translation and calculated the BLEU score for three different datasets.

* represents the model used in [3] for English-to-Hindi translation.

+ Observations on CIFLT dataset @ Samanantar dataset

Model	BLEU Score
2 Layer LSTM + SGD*	16.854 [+]
4 Layer LSTM + SGD*	17.124 [+]
2 Layer (Bi-dir) LSTM +SGD*	18.100 [+]
4 layers (Bi-dir) LSTM +SGD + Res*	18.215 [+]
MBartForConditionalGeneration (transformer model with 12 encoder and decoder layers (each) - used by us)	63.693[@]

We can clearly see that there is a considerable increase in performance using the transformer based architecture.

In paper review 1, [5] calculated BLEU scores of De→En using GOTTBERT which was obtained around 36.32 and for En→De using ROBERTA which scored around 28.74. As the language used for translations are different from the ones used by us i.e., En→Hn and En→Bn translation, we can't make a direct comparison in the models.

In paper review 2, [10] calculated BLEU-4 scores of De→En using smart start decoding which was obtained around 35.61 and for En→De using smart start decoding which scored around 30.01. As the language used for translations are different from the ones used by us i.e., En→Hn and En→Bn translation, we can't make a direct comparison in the models.

4. Make clear documentation of the same along with model-related information like architecture, training, validation and test splits, hyperparameters choice (and appropriate reasoning), and any other design considerations made, shortcomings of the model, limitations etc.

4.1. Architecture of the model used:

Sequence-to-Sequence Transformer Architecture with 12 encoder and decoder layers (each) were used in the model with a model dimension of 1024 on 16 heads. An additional layer normalization on top of encoder and decoder layers were used.

architectures	MBartForConditionalGeneration
model_type	mbart
transformers_version	4.18.0
d_model	1024
decoder_attention_heads	16
decoder_ffn_dim	4096
dropout	0.1
decoder_layers	12
encoder_attention_heads	16
encoder_ffn_dim	4096
encoder_layers	12
encoder_layerdrop	0.0

The model consists of Encoder and Decoder of 12 layers each.

<u>First layer of Encoder looks like:</u>

```
MBartEncoder(
  (embed_tokens): Embedding(250054, 1024, padding_idx=1)
  (embed_positions): MBartLearnedPositionalEmbedding(1026, 1024)
  (layers): ModuleList(
    (0): MBartEncoderLayer(
      (self_attn): MBartAttention(
        (k_proj): Linear(in_features=1024, out_features=1024, bias=True)
        (v_proj): Linear(in_features=1024, out_features=1024, bias=True)
        (q_proj): Linear(in_features=1024, out_features=1024, bias=True)
        (out_proj): Linear(in_features=1024, out_features=1024, bias=True)
      )
      (self_attn_layer_norm): LayerNorm((1024,), eps=1e-05, elementwise_affine=True)
      (activation_fn): ReLU()
      (fc1): Linear(in_features=1024, out_features=4096, bias=True)
      (fc2): Linear(in_features=4096, out_features=1024, bias=True)
      (final_layer_norm): LayerNorm((1024,), eps=1e-05, elementwise_affine=True)
    )
    (1): MBartEncoderLayer(
```

First layer of Decoder looks like:

```
MBartDecoder(
  (embed_tokens): Embedding(250054, 1024, padding_idx=1)
  (embed_positions): MBartLearnedPositionalEmbedding(1026, 1024)
  (layers): ModuleList(
    (0): MBartDecoderLayer(
      (self_attn): MBartAttention(
        (k_proj): Linear(in_features=1024, out_features=1024, bias=True)
        (v_proj): Linear(in_features=1024, out_features=1024, bias=True)
        (q_proj): Linear(in_features=1024, out_features=1024, bias=True)
        (out_proj): Linear(in_features=1024, out_features=1024, bias=True)
      )
      (activation_fn): ReLU()
      (self_attn_layer_norm): LayerNorm((1024,), eps=1e-05, elementwise_affine=True)
      (encoder_attn): MBartAttention(
        (k_proj): Linear(in_features=1024, out_features=1024, bias=True)
        (v_proj): Linear(in_features=1024, out_features=1024, bias=True)
        (q_proj): Linear(in_features=1024, out_features=1024, bias=True)
        (out_proj): Linear(in_features=1024, out_features=1024, bias=True)
      )
      (encoder_attn_layer_norm): LayerNorm((1024,), eps=1e-05, elementwise_affine=True)
      (fc1): Linear(in_features=1024, out_features=4096, bias=True)
      (fc2): Linear(in_features=4096, out_features=1024, bias=True)
      (final_layer_norm): LayerNorm((1024,), eps=1e-05, elementwise_affine=True)
    )
    (1): MBartDecoderLayer(
```

4.2. Train Test Split:

Due to computational issues, we have used only the first 200000 data for our analysis.

English-Hindi Translation	English-Bangla Translation
80:20	80:20

4.3. Hyperparameters used:

tokenizer_class	MBart50Tokenizer
activation_function	relu
activation_dropout	0.0
attention_dropout	0.0
max_length	200
max_position_embeddings	1024
use_cache	true
Learning rate	0.002
vocab_size	250054

4.4. Shortcomings/Limitations of the model:

The model showed good performance for English to Hindi Translation, but for English to Bangla Translation, the performance was not so good.

For Bangla language translation, the model was performing well on simple words/ sentences, but the performance dropped significantly for the words where complex alphabets were used. The model was showing the wrong translation for those cases. Also the model was not able to translate long sentences properly.

Sentence in English	Sentence in Bengali	Translation in Bengali
NewYork Times announced this.	এ খবর জানিয়েছে মার্কিন দৈনিক দ্য নিউইয়র্ক টাইমস।	নিউ ইয়র ্ ক টাইমস ্ বেশ তা প ্ রকাশিত করেছিল ।
But it will not show up.	কিন্তু সেটা দেখা হবে না।	কিন ্ তু এটি দেখাবে না ।
the Horticulture Department	উদ্যানতত্ব বিভাগ	জঙ ্ গমন ্ ত ্ রী
We hope that he will consider our request.	আমরা আশাবাদী, তিনি আমাদের দাবিকে মান্যতা দেবেন"।	আমরা আশা করি যে, তিনি আমাদের অনুরোধ পর ্ যবেক ্ ষণ করবেন ।
I asked him.	আমি তাকে জিজ্ঞেস করেছিলাম।	আমি ওকে জিজ ্ ঞেস করলাম

Even though the meaning derived from the predicted translation was quite understandable, the sentence formulation was not good.

Sentence in English	Sentence in Bengali	Translation in Bengali
I followed all instructions given by the doctors.	চিকিৎসকদের সব ধরনের পরামর্শ মেনে চলছি।	আমি ডাক ্ তারদের দেওয়া সব প ্ রক ্ রিয়া অনুসরণ করি ।
Was he entering politics?	তবে কি তিনি রাজনীতিতেই আসছেন?	তিনি কি রাজনীতিতে প ্ রবেশ করছিলেন?
He has had a heart attack.	তার হার্ট অ্যাটাক হয়েছিল।	সে হৃদরোগ আক ্ রান ্ ত ছিল ।

To overcome this shortcoming, we should train our model on ever larger corpus (more than 200000 lakh data points, which we used due to computational restraints.) This will help the model to learn all the complex features of the language and thus the model will perform better.

5. Show some examples where the model has given correct translations as well as some wrong ones.

5.1. English-Hindi Translations:

Sentence in English	Translation in Hindi
I enjoy doing my work and I give my best to it.	मैं अपने काम में आनंद लेता हूं और उसमें अपना पूरा प्रयास करता हूं।
This Annual Athletic Meet gave the students an opportunity to prove their sporting abilities and win laurels.	इस वार्षिक एथलेटिक्स मीट ने विद्यार्थियों को अपने खेल-कूद की क्षमता को सिद्ध करने और लारल जीतने का अवसर दिया।
The film will have Aamir Khan playing the titular role.	इस फिल्म में अमीर खान का शीर्षक भूमिका निभाया जाएगा।
This reflects the fact that in many programming languages these are the characters that may be used in identifiers.	यह इस तथ्य को प्रतिबिंबित करता है कि कई प्रोग्रामिंग भाषाओं में ये वे अक्षर हैं जिन्हें पहचानकर्ताओं में प्रयोग किया जा सकता है।

```
This Annual Athletic Meet gave the students an opportunity to prove their sporting abilities and win laurels.
Hindi Predicted Translation:  इस वार्षिक एथलेटिक्स मीट ने विद्यार्थियों को अपने खेल-कूद की क्षमता को सिद्ध करने और लारल जीतने का अवसर दिया।
Ground Truth:  इस वार्षिक खेल आयोजन ने छात्रों को अपनी खेल प्रतिभाओं को प्रदर्शित करने और पुरस्कार जीतने का अवसर प्रदान किया।
BLEU Score:  0.6920355067556669
ROUGE Score: 0.569620253164557
METEOR Score: 0.446

**************************************************************************

I enjoy doing my work and I give my best to it.
Hindi Predicted Translation:  मैं अपने काम में आनंद लेता हूं और उसमें अपना पूरा प्रयास करता हूं।
Ground Truth:  मेहनत और लगन से काम करती हूं और अपने काम को एन्जॉय करती हूं।
BLEU Score:  0.7324912081306231
ROUGE Score: 0.5238095238095238
METEOR Score: 0.319

**************************************************************************

"""All evidence has been collected."
Hindi Predicted Translation:  "सभी साक्ष्य एकत्र किया गया है।"
Ground Truth:  सारे साक्ष्य एकत्र कर लिए गए हैं।
BLEU Score:  0.8274377299117183
ROUGE Score: 0.6333333333333333
METEOR Score: 0.264
```

Almost all the hindi translations obtained were correct conveying the required meaning. Though the way of speaking was a bit different.

5.1. English-Bangla Translations:

5.1.1. Correct translation:

Sentence in English	**Translation in Bengali**
As happened to me once.	যেমনটি একবার আমার সাথে ঘটেছে ।
I am terrified too.	আমিও ভয় পেয়েছি ।
The girl is currently undergoing treatment.	মেয়েটা এখন চিকিৎসার মধ ্য যে আছে ।

Finally you see.	অবশেষে আপনি দেখলেন ।

5.1.2. Incorrect translations:

Sentence in English	Translation in Bengali
Please don't leave!	তোমরা, চলুন না!
He was not disappointed.	তিনি খুব খুশি হলেন না ।
The trial in the case is yet to begin.	এক ◌ ষেত ◌ রে বিচারটা মুহূর ◌ তের মধে ◌ য থেকে শুরু হবে'
Similar is the case with children.	এরকমই বাচ ◌ চাদের ক ◌ ষেত ◌ রে ।
FIFA boss Gianni Infantino had invited the boys Wild Boars football team to Sunday's World Cup final last week.	ঀ৩ বড় দাননি ইন ◌ বান ◌ টিনো গত সপ ◌ তাহে ছেলেদের বিয ◌ োশ ◌ ব অফল ◌ যাল ◌ যান ◌ ড সিন ◌ টেমকে আমন ◌ ত ◌ রণ জানিয়েছিল তান ◌ দ ◌ র ◌ যাক ◌ তি বিশ ◌ ব পর ◌ যায়ের প ◌ রতিনিয়ত ।

References:

[1]Samanantar: The Largest Publicly Available Parallel Corpora Collection for 11 Indic Languages

[2]https://huggingface.co/docs/transformers/model_doc/mbart

[3] Abhirut Gupta, Aditya Vavre, and Sunita Sarawagi. 2021. Training Data Augmentation for Code-Mixed Translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5760–5766, Online. Association for Computational Linguistics.

[4] S. Saini and V. Sahula, "Neural Machine Translation for English to Hindi," *2018 Fourth International Conference on Information Retrieval and Knowledge Management (CAMP)*, 2018, pp. 1-6, doi: 10.1109/INFRKM.2018.8464781.

[5] Xu, Haoran & Durme, Benjamin & Murray, Kenton. (2021). BERT, mBERT, or BiBERT? A Study on Contextualized Embeddings for Neural Machine Translation. 6663-6675. 10.18653/v1/2021.emnlp-main.534.

[6]https://neptune.ai/blog/hugging-face-pre-trained-models-find-the-best

[7]https://cloud.google.com/translate/automl/docs/evaluate#:~:text=BLEU%20(BiLingual%20Evaluation%20Understudy)%20is,of%20high%20quality%20reference%20translations.

[8] https://towardsdatascience.com/the-most-common-evaluation-metrics-in-nlp-ced6a763ac8b

[9]https://towardsdatascience.com/how-to-evaluate-text-generation-models-metrics-for-automatic-evaluation-of-nlp-models-e1c251b04ec1

[10]https://aclanthology.org/2021.naacl-main.312.pdf
Jian Yang1 , Shuming Ma, Dongdong Zhang, Juncheng Wan, Zhoujun Li1 , Ming Zhou.(2021) “Smart-Start Decoding for Neural Machine Translation” . DOI : 10.18653/v1/2021.naacl-main.312