Progress Report

Nanditha Merugu | 2020102061

Anantha Lakshmi Yadavalli | 2020101103

Harshavardhan Thatipamula | 2020101106

Translation Model	Paraphrasing	Encountering abuse
Translation model that translated Hindi to English sentences.	Designing the Paraphrasing model.	Encountering abuse and offensive language

Github Link: https://github.com/nandithamerugu/Honours_Final_Project.git

Translation Model using Seq2Seq and Transformer

The objective is to create a neural machine translation model that can accurately translate Hindi sentences into English.

Dataset

I have used the dataset provided by the IITK (Train Data). The training dataset we've been provided with is a CSV file containing pairs of Hindi sentences and their corresponding English translations. The dataset comprises 102,322 such pairs. Upon examination, it's evident that some Hindi sentences contain interspersed English words. We are also provided the final test data with actual translation. It contains 24,101 Hindi sentences. It was divided into four sets. Here is the link for the first set (Test Data).

Data Pre-processing

1. Filter sentences based on length:

- Check if sentence length is less than Max sentence length.

2.Check for null values:

- Remove pairs with null values in either 'Hindi' or 'English' senten

3.Lowercase conversion:

- Convert all 'English' sentences to lowercase characters.

4.Remove punctuations:

- Remove punctuations from both source and target sentences.

5.Create vocabularies:

- Initialize vocabularies for both source and target languages.
- Add special tokens like "start," "end," "pad," and "ukn" (for unknown words).
- Process each sentence and add new words to the corresponding vocabulary.
- Add "start" and "end" tokens to the sentence.
- Add required number of "pad" tokens to equalize the sentence length to Max length.

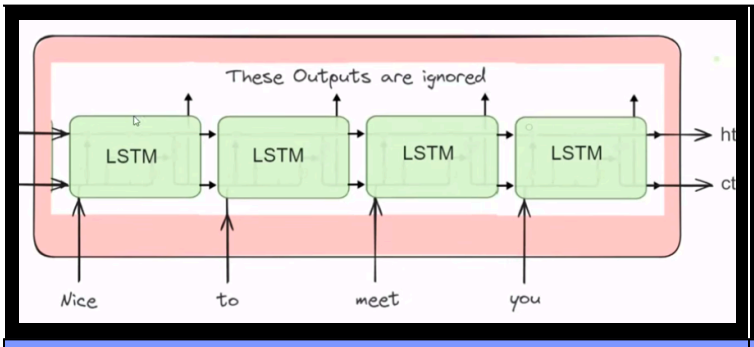
6.Convert sentences to tensors:

- Convert each sentence into a tensor using its index in the corresponding vocabulary for model input.

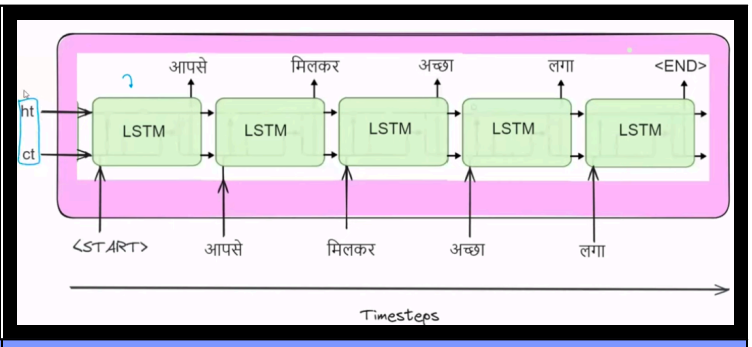
Model Description

1. Seq2seq

- In the initial phase, I delved into sequence-to-sequence (**seq2seq**) models, the cornerstone for neural machine translation. These models are comprised of two distinct components: the Encoder and the Decoder.
- The Encoder captures the essence of a sentence, condensing it into a context vector. Subsequently, the Decoder utilises this contextual information for accurate translation.



Encoder



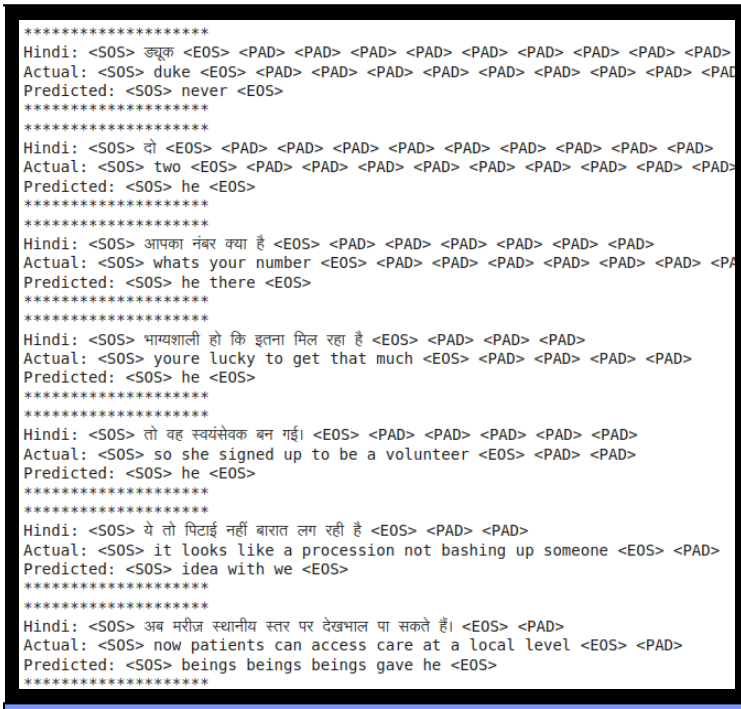
Decoder

- After experimenting with both GRU and LSTM architectures in both the Encoder and Decoder components, it was evident that GRU models trained faster, yet LSTM models yielded superior results overall.
- Subsequently, I explored the integration of Bi-directional LSTMs in the subsequent phase, but they fell short of anticipated performance levels. Generally, it became apparent that as sentence length

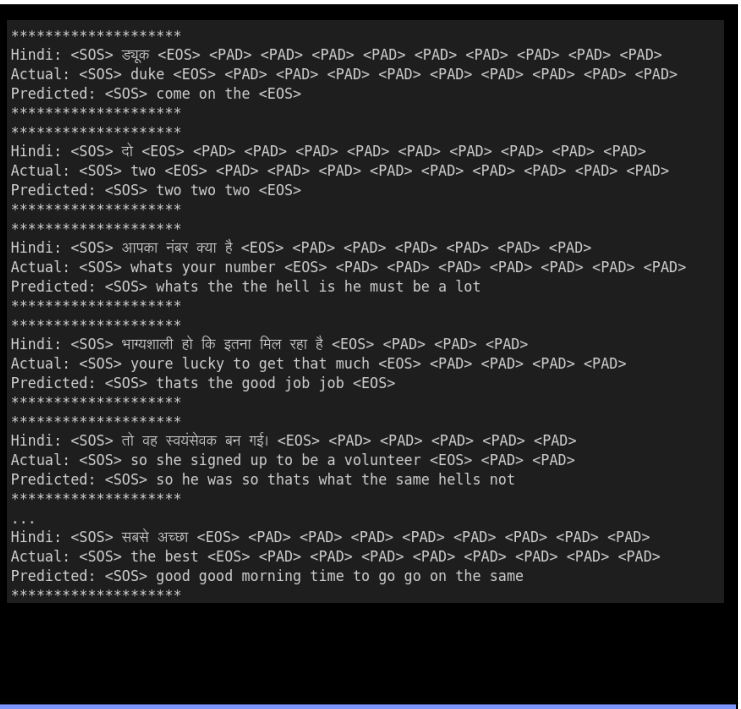
increased, simple seq2seq models struggled due to their inability to effectively utilize the original context vector at each decoding step.

Results:

- Initially i ran the model for 5 epochs and then for 10 epochs. Below are the results for the predicted and actual translated output.
- The predicted are not even very close to the actual translated.
- Then ran the model for 50 epochs. (which took lot of time around 8 to 10 hours for training). They are very little close as compared to the previous.
- The Bleu score of this is also in 0.01 range.



5 Epochs



50 epoch

Link for the code and result files:

I am hereby attaching the Github link for the code and output files.

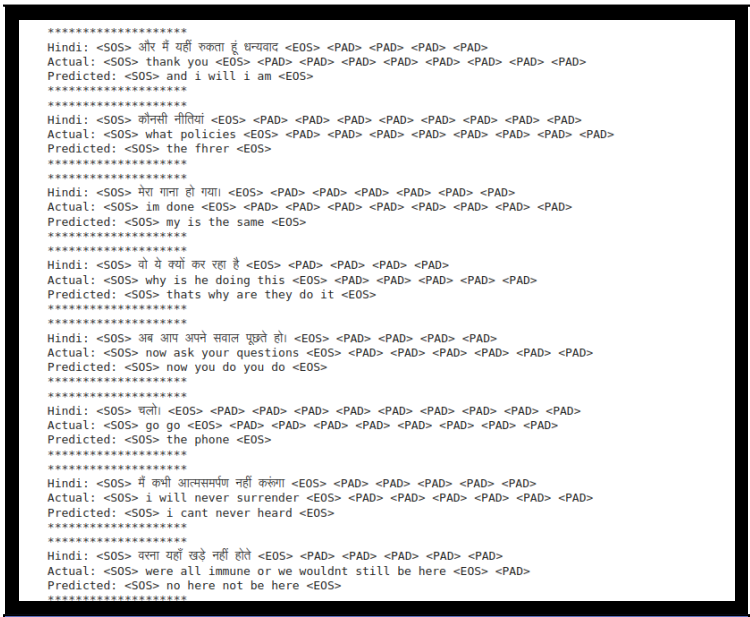
[Translation Model Using Seq2Seq]

2.Transformer

1. **Word and positional embeddings**:
- Word embeddings represent vocabulary in dense form.
- Positional embeddings encode word positions to address translation order.
- Implemented using nn.Embedding to generate a lookup table.
- Final embedding for the Transformer is the sum of word and positional embeddings.
2. **Mask generation**:
- Masks are created for both source (Hindi) and target (English) languages.
- Source mask: (batch size, max length) with '1' for pad tokens.
- Target mask: Lower triangular matrix to allow model to focus on one word at each step.
3. **Transformer model:**
- Embeddings and masks are passed to the Transformer model.
- Output is fed to a fully connected layer to predict target language words.
4. **Training**:
- Trained for 100 epochs, faster than seq2seq architecture.
- Used batch size of 64 and Adam optimizer.
- Cross-Entropy Loss employed for calculating loss during training.
- Addressed Gradient Explosion problem using nn.utils.clip_grad_norm() function in PyTorch.

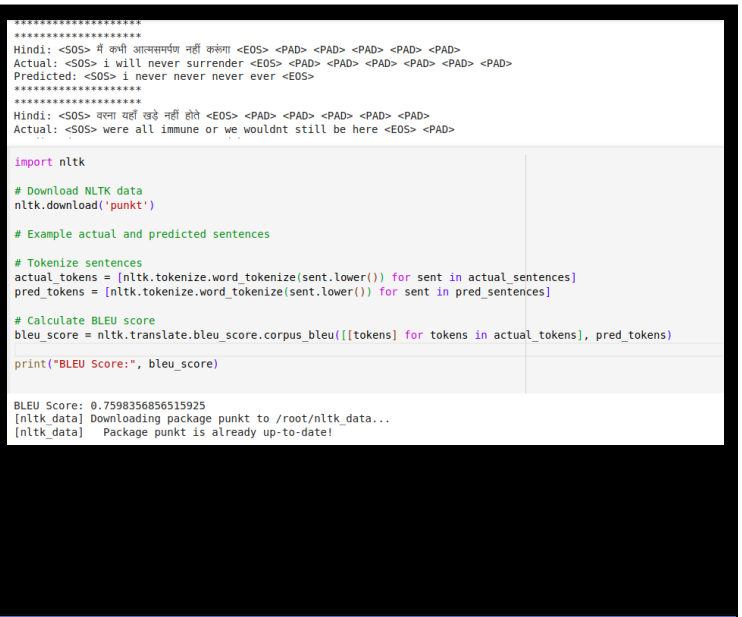
Result

- The below is the result for the above model.
- Bleu score for this model is 0.75.



```
********************
Hindi: <SOS> और मैं यहीं रुकता हूं धन्यवाद <EOS> <PAD> <PAD> <PAD> <PAD>
Actual: <SOS> thank you <EOS> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD>
Predicted: <SOS> and i will i am <EOS>
********************
********************
Hindi: <SOS> कौनसी नीतियां <EOS> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD>
Actual: <SOS> what policies <EOS> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD>
Predicted: <SOS> the fhrer <EOS>
********************
********************
Hindi: <SOS> मेरा गाना हो गया। <EOS> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD>
Actual: <SOS> im done <EOS> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD>
Predicted: <SOS> my is the same <EOS>
********************
********************
Hindi: <SOS> वो ये क्यों कर रहा है <EOS> <PAD> <PAD> <PAD> <PAD>
Actual: <SOS> why is he doing this <EOS> <PAD> <PAD> <PAD> <PAD> <PAD>
Predicted: <SOS> thats why are they do it <EOS>
********************
********************
Hindi: <SOS> अब आप अपने सवाल पूछते हो। <EOS> <PAD> <PAD> <PAD> <PAD>
Actual: <SOS> now ask your questions <EOS> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD>
Predicted: <SOS> now you do you do <EOS>
********************
********************
Hindi: <SOS> चलो। <EOS> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD>
Actual: <SOS> go go <EOS> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD>
Predicted: <SOS> the phone <EOS>
********************
********************
Hindi: <SOS> मैं कभी आत्मसमर्पण नहीं करूंगा <EOS> <PAD> <PAD> <PAD> <PAD> <PAD>
Actual: <SOS> i will never surrender <EOS> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD>
Predicted: <SOS> i cant never heard <EOS>
********************
********************
Hindi: <SOS> वरना यहाँ खड़े नहीं होते <EOS> <PAD> <PAD> <PAD> <PAD> <PAD>
Actual: <SOS> were all immune or we wouldnt still be here <EOS> <PAD>
Predicted: <SOS> no here not be here <EOS>
********************
```

Model output



```
********************
********************
Hindi: <SOS> मैं कभी आत्मसमर्पण नहीं करूंगा <EOS> <PAD> <PAD> <PAD> <PAD> <PAD>
Actual: <SOS> i will never surrender <EOS> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD>
Predicted: <SOS> i never never never ever <EOS>
********************
********************
Hindi: <SOS> वरना यहाँ खड़े नहीं होते <EOS> <PAD> <PAD> <PAD> <PAD> <PAD>
Actual: <SOS> were all immune or we wouldnt still be here <EOS> <PAD>
```

```
import nltk

# Download NLTK data
nltk.download('punkt')

# Example actual and predicted sentences

# Tokenize sentences
actual_tokens = [nltk.tokenize.word_tokenize(sent.lower()) for sent in actual_sentences]
pred_tokens = [nltk.tokenize.word_tokenize(sent.lower()) for sent in pred_sentences]

# Calculate BLEU score
bleu_score = nltk.translate.bleu_score.corpus_bleu([[tokens] for tokens in actual_tokens], pred_tokens)

print("BLEU Score:", bleu_score)
```

```
BLEU Score: 0.7598356856515925
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

BLEU score

Link for the code and result files:

I amhereby attaching the Github link for the code and output files.

[Translation Model using transformer]

Paraphrasing Model

Overview:

Rewriting the text to: preserver the text meaning, eliminate the toxic style.

Example:

They are all communists who hate the USA

They are all communists who dislike the USA

Understanding the problems:

Problem 1

Despite the aim of maintaining the original content, the process of altering the sentence's style often leads to substantial changes in its meaning.

Problem 2

There are TST(Text Style transfer) models. Both content and style are ambiguous concepts. TST often changes the original meaning.

Solution

Detoxification requires better preservation of the original meaning than other style transfer tasks, so it should be done differently.

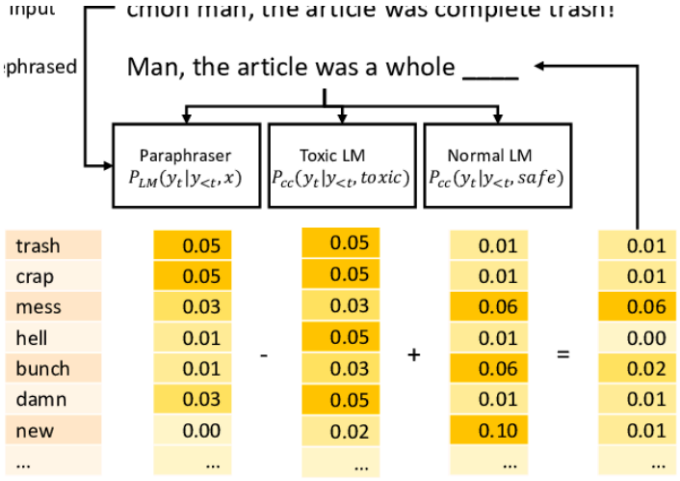
Methodology:

Our method combines two recent ideas:

Two novel methods for text detoxification:

(a) ParaGedi (paraphrasing GeDi)

(b) CondBERT (conditional BERT)



The method is based on two ideas:

- A large pre-trained paraphraser model can preserve the meaning.
- A pre-trained language model conditional on style can control the style.

Then the two models can be combined using the Bayes rule.

$$P(y_t|y_{<t}, x, c) \propto P_{LM}(y_t|y_{<t}, x)P(c|y_t, y_{<t}, x)$$
$$\approx P_{LM}(y_t|y_{<t}, x)P_D(c|y_t, y_{<t})$$

Flow of code:

Implementing and training of Generative Discriminator

Mining the dataset and Finetuning of the Paraphraser

Implementing the J metric for evaluation

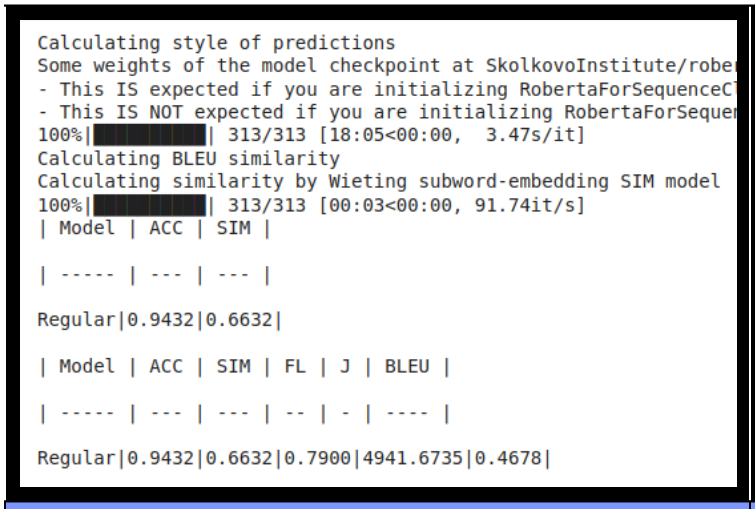
ACC: mean score of RoBERTA-based toxicity classifer.

SIM: is evaluated based on the similarity between sentence-level embeddings of the original and transformed texts. This is achieved by employing the trained SIMILE model.

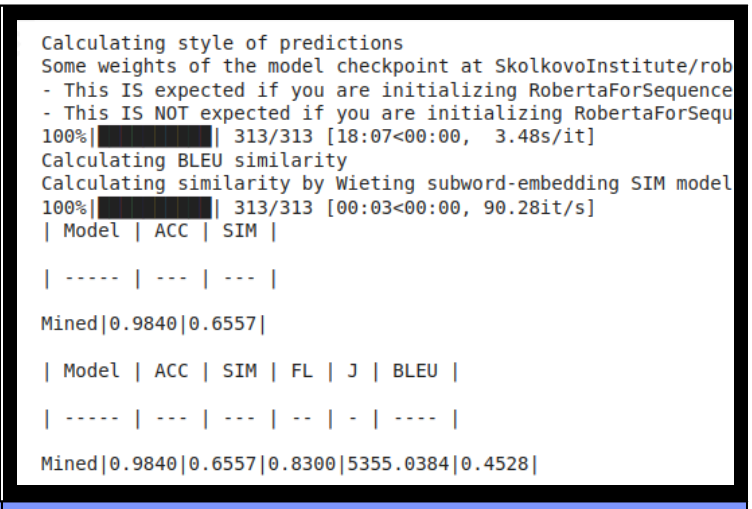
Fluency (FL): is measured using a classifier of linguistic acceptability that has been trained on the CoLA dataset.

J : mean product of sentence level ACC, SIM and FL.

Results:

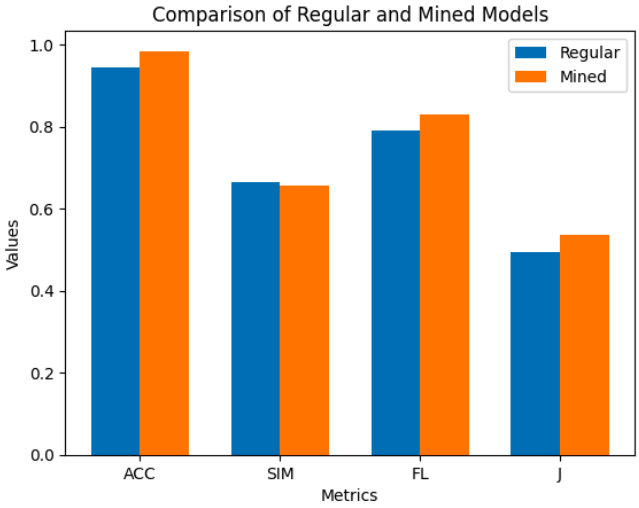


Regular Model



Mined Model

Comparison of Both the model:


Comparison of Regular and Mined Models
Regular
Mined
Values
1.0
0.8
0.6
0.4
0.2
0.0
ACC
SIM
FL
J
Metrics