TRANSLATION FROM ENGLISH TO INDIC LANGUAGE

Group members : Pragati Sinha (B19CSE065)
Sandip Kumar Burnwal (B19CSE075)

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.

Paper 1: Attention is All you need , 31st conference NIPS , Dec, 2017.

Introduction

In this research paper the authors proposed a new simple network architecture, transformer based on attention mechanism dealing with recurrence and convolutions entirely. A RNN is a type of artificial neural network that works with data that is presented in a sequential or time series format. Language translation, nlp , speech recognition, and image captioning are all examples of problems where deep learning techniques are used. Siri, voice search, and Google Translate are just a few examples of how this can be used.

Recurrent neural networks are the typical factor computation models which work along the symbol position of input and output sequence. Aligning the positions to step in computation time, they generate a sequence of hidden state h_{t-1} and the input for position t. Because of this, parallelization within training instances is impossible, which becomes essential as sequence lengths grow larger and memory limits limit batching among samples.

Another mechanisms which are used by the authors are Attention mechanisms. They have become an integral part of compelling sequence modeling. They allow the modeling of dependencies without regard to their distance in the input or output sequence. In this work the model architecture Transformer is mainly using the recurrence architecture as compared to attention mechanisms in order to draw global dependencies between input and output.

Model Architecture

Most of the neural sequence transduction models have an encoder and decoder structure. In this model the encoder maps an input sequence of symbols with the sequence of continuous representation z. (x1,x2,x3…..xn) is considered as an input sequence of symbols and z =(z1,z2,z3……zn) can be considered as a sequence of continuous representation. The decoder generates an output sequence y = (y1,y2,y3…..yn) of symbols. At each and every step the previously generated symbols are consumed as additional input and then it generates the next output sequence. This is called auto-regression. The image of the model architecture is attached below for better understanding.

Encoder: The encoder is made up of six layers that are all the same. For each layer, there are two sub-layers. The multi-head self-attention mechanism is the first layer, and the position-wise completely connected feed-forward network is the second. Every two sub layers, as well as the normalization layers, have a residual connection that is responsible for generating the output of each sub-layer, which is LayerNorm(x + Sublayer(x)). Sublayer(x) is a function that the sub-layer implements.

Decoder: The decoder, like the encoder, has six identical layers. For each layer, there are three sub-layers. The two sublayers perform the same job as the encoder layer, but the third sub-layer performs multi-head attention on the encoder stack's output. There are residual connections around each sub-layer, as well as the normalization layer, similar to the encoder. The self-attention sub-layer has been tweaked to prevent positions from attending to the positions after them.

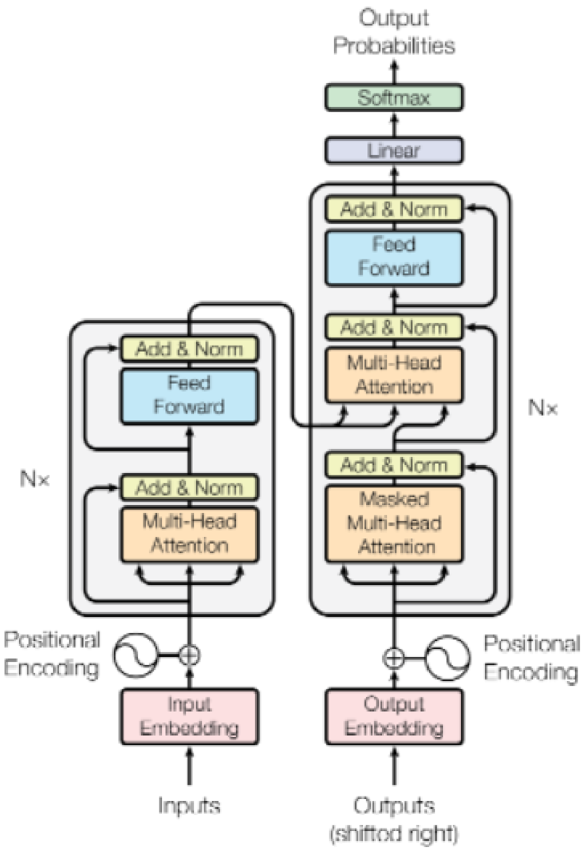


Figure 1: The Transformer - model architecture. Image Src

Attention

One of the most important achievements in Deep Learning research in the previous decade is the attention mechanism. It has produced a slew of recent innovations in natural language processing (NLP), including Google's BERT and the Transformer architecture. The attention function was employed in the model by the authors. A mapping query and a set of key-value pairs to an output can be defined as these. The output is generated as a weighted sum of the values, and the query, keys, and output values are all vectors. A compatibility function of the query with the relevant key is used to calculate the weight allocated to each value.

Application of Attention in the model:

The queries originate from the previous decoder layer, while the memory keys and values come from the encoder's output, as mentioned in the "encoder-decoder attention" layers. This causes every decoder location to pay attention to every position in the input sequence.

Training:

Training Data and Batching: The authors had trained their model on the standard WMT 2014 English-German dataset comprising 4.5 million sentence pairs. For English-French they had trained their model on WMT 2014 English-French dataset consisting of 36M sentences. The number of source tokens present in the training batch is around 25000 and the number of target tokens present in the training batch is approximately equal to the number of source tokens. Training a model on such a large dataset ensures a high accuracy value. They had used an optimizer in this model.

Optimizers are algorithms or strategies for minimizing an error function (loss function) or increasing production efficiency. They had used Adam optimizer with β1 = 0.9, β2 = 0.98 and ϵ = 10^(-9). They varied the learning rate over the course of training, by using this formula:

$$lrate = d_{\text{model}}^{-0.5} \cdot \min(step\_num^{-0.5}, step\_num \cdot warmup\_steps^{-1.5})$$

According to the formula, if we increase l.r. in the first warmup step and decrease it a/q to the inverse square root of the step number leads to overall decrease in l.r.

Results

The results achieved by this model are tremendous. On the WMT 2014 English-to-German translation task, the big transformer model had performed better than the best previously reported model. The new *Bilingual evaluation Understudy* score for this model was 28.4 which is more than by 2.0, scored by the previously reported model.

Paper 2 : Neural Machine Translation from English to Hindi, IJRASET, Volume 8, Issue V May 2020

Introduction:

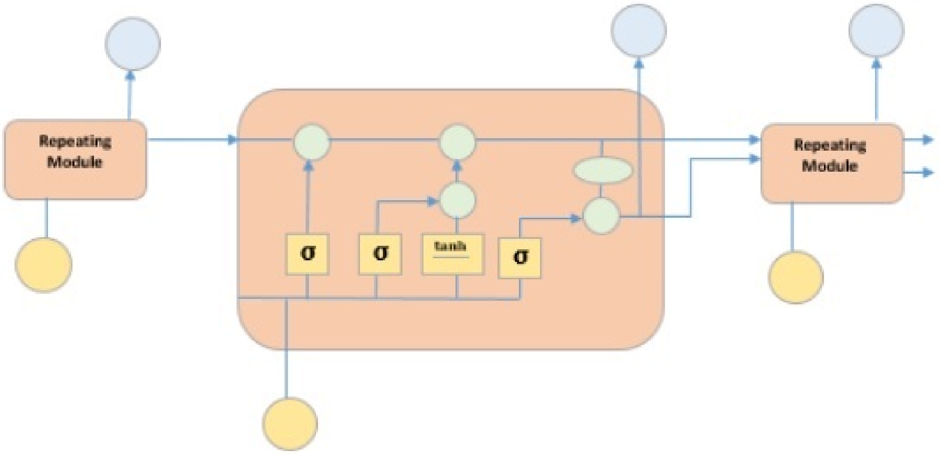
In this research paper, authors proposed a sequence to sequence learning mechanism using the LSTM model which means long and short term memory models. Machine Translation is one of the tasks which is taken by computer scientists and in this field research is going past 50 years. Most of the machine translations are based on statistical machine translation. In recent years, many big companies, such as Google have shifted its translation to Neural Machine Translation. This neutral network based translation system comprises encoder and decoder both having eight layers.

Neural Machine Translation

NMT uses a bidirectional recurrent neural network. It is also known as encoder which converts a source sentence into vectors and transfers it to a second recurrent neural network known as decoder. This decoder helps in predicting the word in target language. This machine translation approaches a very large artificial neural network predicting the sequence of words in the form of a complete sentence. Statistical machine translation consumes more space and is also not very time efficient. Neural machine translation trains it to parts end to end to increase the performance.

Long Short Term Memory

As the name suggests there must be a combination of short term and long term memory. The key recurrent components are LSTM memory cells. Suppose there are 10 units of memory cell then there may be two units that can carry their long term information and the rest of units can carry short term memory information that's why it is called as long short term memory. All the cells are controlled by gates. The gating is learnt in terms of linear combination of waited inputs and hidden units.



imagesource

There are four neural network layers in yellow boxes, point wise operators in green circles, yellow circles representing input, and blue color indicating cell state in the image. An LSTM model has three gates and a cell state, giving it the ability to selectively learn, unlearn, or retain information from each unit. Each unit contains an input, output, and a forget gate that can be used to add or remove data from the cell state. The forget gate decides which information from the previous cell state should be forgotten for which it uses a sigmoid function. The input gate uses a pointwise multiplication operation of sigmoid and 'tanh' to control the information flow to the current cell state. Finally, the output gate determines which data should be sent to the next concealed state.

Algorithm

The algorithm proposed by the authors can translate in real time and the probability of sequences of words. NMT performs the best quality translation produced by these systems containing 17% fewer lexical errors and 50% fewer words and 19% grammatical errors. Algorithm proposed by authors:

1. Pre-processing of Corpus:
   It involves basic text cleaning like removal of url , ip addresses , email and the copyright statements. Upper case words converted to lowercase and punctuation digits etc removed.

2. Preparing Training Data:
   They have to form it machine-ready for training their model. They have to perform Tokenizing and Indexing. Tokenization in NLP is a way of separating a piece of text into smaller units called tokens. For tokenization they will find all the unique words in both the languages. Dimensionality of the index array is determined by tokenization. Now they will make three numpy arrays with encoders for input, decoders for input, and target decoders.
   This will be used to index each word. The sizes 30 and 32 in the preceding step are related to the maximum sentence lengths we've decided on.
   The encoder (Hindi) has a value of 30 while the decoder(English) has a value of 32.

3. Word embedding:
   It refers to possessing a featurized representation for every word. The main motive is to find out a group of features and their values, in order to get dense vector representation for words. Consider a statement "Man-Woman-King-Queen", if 'Man' is at position 5545 during a vocabulary size of 10,000 the one-hot vector for 'Man' may be 10,000-dimensional vector of 0s, with just one entry 1 at 5545 denoted as O_{5545}. Now these words are defined by some features, say 50, a new matrix of all 50 features wll be defined. Each word of this vector is going to be a 50 dimensional vector space. To seek out the word

embedding of ‘Man’, the embedding matrix is multiplied with a one-hot encoded vector(O_{5545}) for ‘Man’.

4. Sequence to Sequence Learning:
   The vectors for source sequences in Hindi are given to the encoder network, each word at a time. The input sentences are encoded into a fixed dimensional vector, and encoder LSTM will provide hidden and cell state and then it will be given to decoder LSTM.

5. Prediction:
   If Beam Width = B = 3 is elected, the model $\hat{Y} = \arg\max P(Y|X)$ evaluates the probability of the primary word, given only input X i.e. $P(y^{<1>}|X)$. The input Hindi sentences run through the encoder LSTM and therefore the initiative of decoder LSTM is going to be a softmax output over all the chances within the English vocabulary. For the primary word, of all the likely translations, top three are chosen. These three choices are stored for subsequent steps. In this step, for every of the three options picked, subsequent choice is estimated i.e. $P(y^{\langle 2 \rangle}|X , y^{\langle 1 \rangle})$.

Results:

The results obtained from neural machine translation from English to Hindi machine translation are comparable with statistical or phrase based machine translation.For a long time, statistical phrase machine translation systems have struggled with the need for big data sets and accuracy. In this paper, we looked into the prospect of solving the machine translation problem with a shallow RNN and LSTM based neural machine translator. The system is evaluated using a BLEU score. In each the score of the translation is different.

Paper 3: Multilingual Indian Language Translation System at WAT 2018: Many-to-one Phrase-based SMT , 32nd Pacific Asia Conference on Language, Information and Computation, ACL 2018

Introduction:

The authors of this study offered a phrase-based statistical machine translation (PBSMT) system as a model. This methodology is intended for both Indic to English and English to Indic language translations. The authors have also presented statistical machine translation to a large number of people (SMT). In India, there are around 122 main languages and 1599 minor languages. These languages are divided into four groups. Although there are similarities between languages that come from the same region , family, there are also many common things between different language groups. These languages are linked because they have lexical, structural, and morphological similarities.

Many-to-one SMT model:

SMT model aims for tremendous transnational outcome in case of data scarcity. The learning of language models and translating models requires a much less data when compared with NMT

SMT can be represented Mathematically as:

$\hat{e}$ = arg max (P(e|f)) = arg max (P(e).P(f|e)) in this formula e is a sentence of the English language and f is a sentence of foreign language.

Given a foreign sentence f, the SMT system chooses the best translated English sentence e. The language model P(e) and the translation model P(f|e) are combined in the argmax computation. For a given sentence f, it generates the English sentence with the highest probability value.

The translation model in this model must be trained on a merged corpus containing all Indic->English language pairs, where all bilingual corpora is transliterated into a specific script pair. The language model is also trained on the target language corpus in its merged form.

NMT model:

NMT stands for Neural Machine Translation. In this model, multilingual transfer learning approaches, including many-to-one, one-to-many, and many-to-many translation, have shown significant improvements in translation quality with minimal increase in network complexity, especially in the case of resource poor languages. According to the authors, the multilingual many-to-one SMT system for Indic language to English translation required less amount of data as compared to NMT.

Pre-processing

Using Moses and the Indic NLP package normalizing , truecasting and tokenizing is done . Tokenization in NLP segregates the sentences into tokens/words. Applying these preprocessing steps, we generated a corpora of all Indian languages transliterated in

Devanagari script using the BrahmiNet transliteration system, which is based on the transliteration module in Moses. Many-to-one SMT systems are trained using this data.

Models trained

The model is trained with the Moses implementation and a 3-gram language model, as well as the grow-diag-final and heuristic for phrase extraction. 14 baseline SMT systems were trained utilizing 7 parallel corpora to train two types of SMT systems. To train the multilingual SMT system, we first combined all of these corpora into a single bilingual corpus in which Indic language sentences were transliterated into Devanagari script. In order to take advantage of the lexical commonality among Indian languages, transliteration is crucial. Following that, the translation model was trained using the seven Indic-English bilingual corpora stated before.

Result and Discussion

The results of the experiment are not very markable. Slight increase is shown by the many-to-one SMT approach as the BLEU score for 3 Malayalam, Urdu and Sindhi, and slight decrease in BLEU scores for Bengali and Tamil. In the case of Hindi and Telugu a noticeable reduction can be observed in BLEU scores.
As a result, just one model must be maintained and hosted. Of course, the multilingual model's phrase table is far larger than the individual models.

2. Implement a transformer-based encoder-decoder architecture for solving the task. & 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.

So the model we implemented was based on the first paper which involves encoder-decoder architecture with attention models taking the help of the transformer library.

1) **Architecture used** : encoder - decoder model with attention base mechanism which overcomes drawbacks of LSTM bases seq2seq architecture.
2) **Training** : For training we gradually increased our dataset from 5000 to 20000 and finally got the best result for the 1,00,000 dataset.
3) **Testing** : Testing was done on trained as well as non trained data (both 10000).
4) **No . of epochs** = 10 .

For all the dataset size we chose from 1st to 10th epoch training loss decreased , for our final dataset of 100000 sentences the decrease in training loss decreased from 1 st to 10 th epoch. Hence we concluded that 10 epochs would be best and further increase in no. of epochs can lead to overfit.

5) **Learning rate** = .0005, adjusting the weight of the network with respect to loss gradient.
   As you can see in the graph of training rate vs epochs this the best learning rate one could have where loss decreases with epoch.
6) **Batch_size** = 128 for 5,000 , 20,000 and 1,00,000 dataset. However it was decreased to 64 for 2,00000

We used **three encoders and three decoder layers** to get the best value.

<u>Oher design consideration:</u>

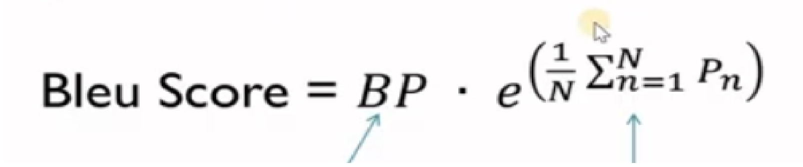
We had to decrease the batch size while training on a 200000 dataset because of GPU memory exceeding.
For hindi and english sentences preprocessing was different.

<u>Short comings & limitations of the model:</u>

1) If you provide any digit to input sentence in translation that digit will get deleted in preprocessing , hence not shown in output.
2) Similarly any URL will also not show up in the output as it gets removed in preprocessing.
3) If suppose there are some other language words in input that will also get removed in output.

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.**

So , the models in all the papers have been evaluated upon BLEU Score i.e. **Bilingual Evaluation Understudy .**



$$\text{Bleu Score} = BP \cdot e^{\left(\frac{1}{N}\sum_{n=1}^{N} P_n\right)}$$

Brevity Penalty Mean of all n-gram precision.

Here ,

$$BP = \begin{cases} 1 & if\ c > r \\ e^{(1-r/c)} & if\ c \leq r \end{cases}$$

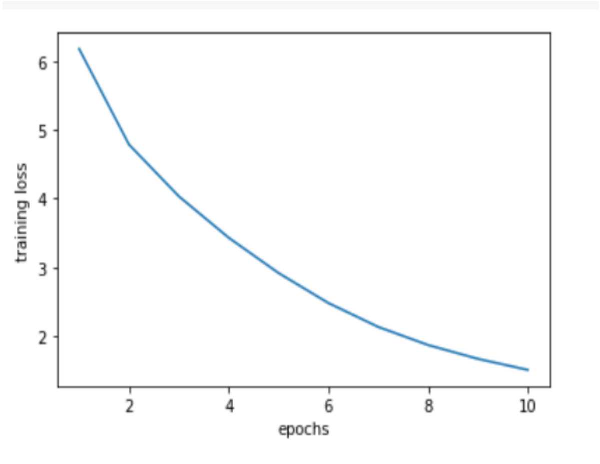
Here r = length of reference sentence from which translated sentence is being compared and c = length of translated sentence.

Higher the BLEU score better is the model performance.

1st Paper	2nd Paper	3rd PAPER	Our model
28.4 on WMT 2014 English-to-German translation task	----	En-bn 11.34 En-hi 26.49 En-ml 14.23 En-ta 15.87 En-te 21.02 En-ur 21.62 En-Si 11.71	**Trained on 200000:26.43** **Trained on 100000:25.71**

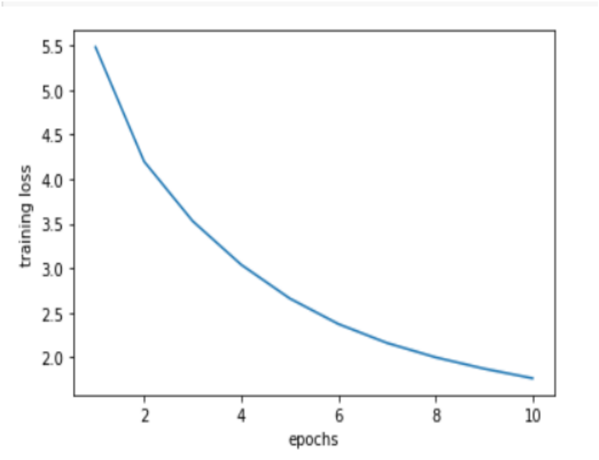
Also we made graphs of training loss after every epoch on a dataset of various sizes.

1,00,000 dataset:



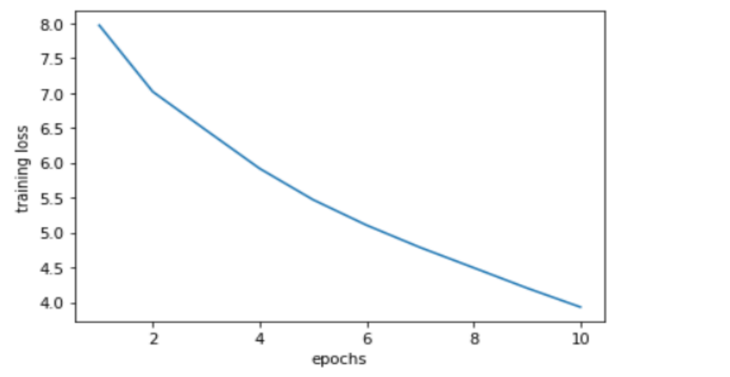
Range of loss = (1,6)

2,00,000 dataset



Range of loss = (1,5.5)

5000 dataset



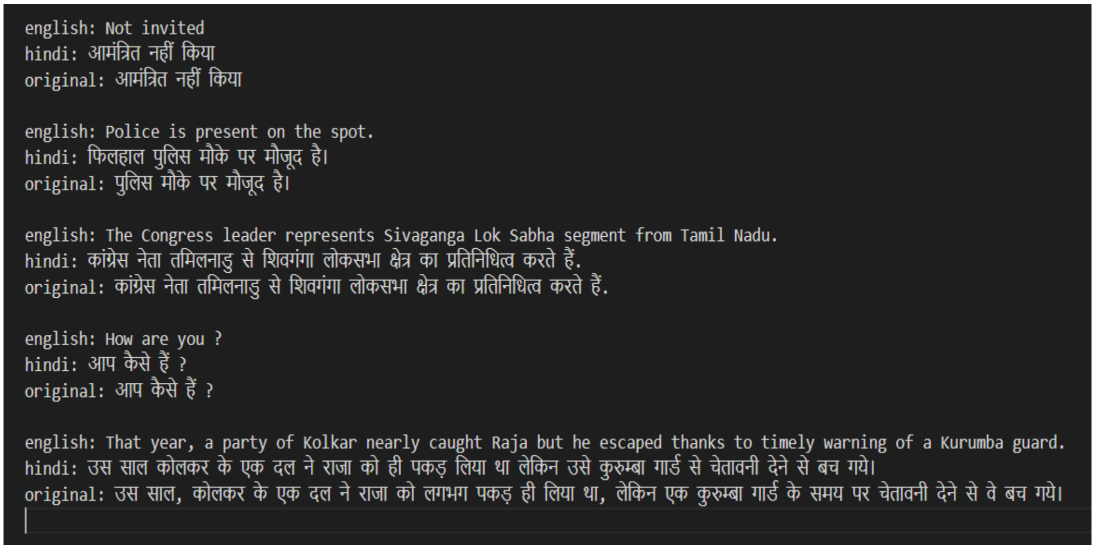
Range of loss = (3,8)

<u>Observations:</u>

- **You can see how performance of the model improved from 5000 to 100000 and 200000 training samples.**
- **However we cannot further increase the dataset as we see there is only a slight difference in slope and range of 1,00000 and 2,00000 dataset. Increasing the dataset will now cause overfit.**

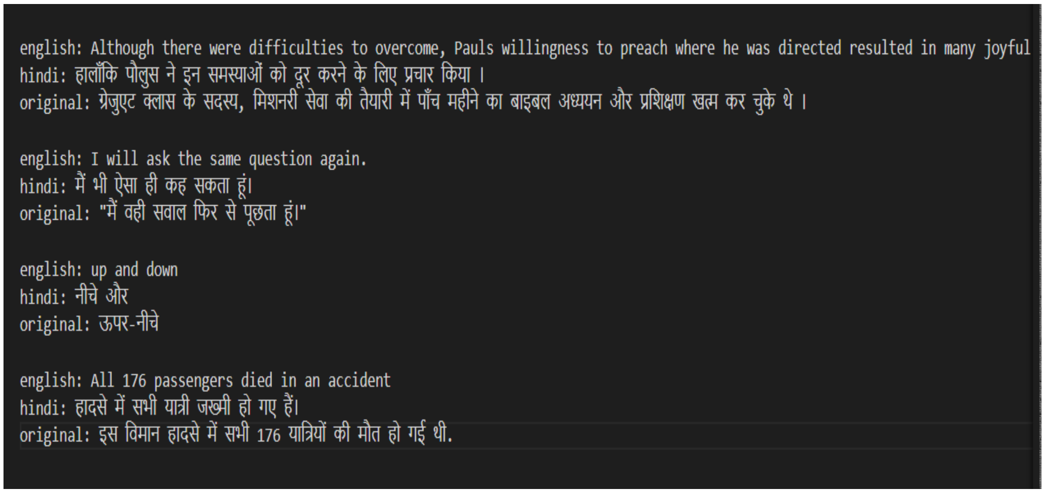
5. Show some examples where the model has given correct translations as as well as some wrong ones.

Correct



```
english: Not invited
hindi: आमंत्रित नहीं किया
original: आमंत्रित नहीं किया

english: Police is present on the spot.
hindi: फिलहाल पुलिस मौके पर मौजूद है।
original: पुलिस मौके पर मौजूद है।

english: The Congress leader represents Sivaganga Lok Sabha segment from Tamil Nadu.
hindi: कांग्रेस नेता तमिलनाडु से शिवगंगा लोकसभा क्षेत्र का प्रतिनिधित्व करते हैं.
original: कांग्रेस नेता तमिलनाडु से शिवगंगा लोकसभा क्षेत्र का प्रतिनिधित्व करते हैं.

english: How are you ?
hindi: आप कैसे हैं ?
original: आप कैसे हैं ?

english: That year, a party of Kolkar nearly caught Raja but he escaped thanks to timely warning of a Kurumba guard.
hindi: उस साल कोलकर के एक दल ने राजा को ही पकड़ लिया था लेकिन उसे कुरुम्बा गार्ड से चेतावनी देने से बच गये।
original: उस साल, कोलकर के एक दल ने राजा को लगभग पकड़ ही लिया था, लेकिन एक कुरुम्बा गार्ड के समय पर चेतावनी देने से वे बच गये।
```

Wrong:



```
english: Although there were difficulties to overcome, Pauls willingness to preach where he was directed resulted in many joyful
hindi: हालाँकि पौलुस ने इन समस्याओं को दूर करने के लिए प्रचार किया ।
original: ग्रेजुएट क्लास के सदस्य, मिशनरी सेवा की तैयारी में पाँच महीने का बाइबल अध्ययन और प्रशिक्षण खत्म कर चुके थे ।

english: I will ask the same question again.
hindi: मैं भी ऐसा ही कह सकता हूं।
original: "मैं वही सवाल फिर से पूछता हूं।"

english: up and down
hindi: नीचे और
original: ऊपर-नीचे

english: All 176 passengers died in an accident
hindi: हादसे में सभी यात्री जख्मी हो गए हैं।
original: इस विमान हादसे में सभी 176 यात्रियों की मौत हो गई थी.
```

Contribution:

Text preprocessing: Hindi : Sandip
English : Pragati

Model (Training , Translating , other utility codes): Pragati and sandip