NLP PROJECT

Made By:
Rishabh Jain - J021
Lovish Kanther - J026

Machine Translation

Machine translation is the task to translate a text from a source language to its counterpart in a target language.
Given a sequence of text in a source language, there is no one single best translation of that text to another language. There are many challenging aspects of Machine Translation:

1. The large variety of languages, alphabets and grammars
2. The task to translate a sequence (a sentence for example) to a sequence is harder for a computer than working with numbers only
3. There is no *one* correct answer (e.g.: translating from a language without gender-dependent pronouns, *he* and *she* can be the same).

Rule Based Machine Translation :

A rule-based system requires experts' knowledge about the source and the target language to develop syntactic, semantic and morphological rules to achieve the translation.

An RBMT system contains a pipeline of Natural Language Processing (NLP) tasks including Tokenization, Part-of-Speech tagging and so on. Most of these jobs have to be done in both source and target language.

Statistical Machine Translation :

Statistical machine translation starts with a very large data set of good translations, that is, a corpus of texts (e.g., United Nations documents) which have already been translated into multiple languages, and then uses those texts to automatically infer a statistical model of translation. That statistical model is then applied to new texts to make a guess as to a reasonable translation. Using Bayes' theorem, we can transform this maximisation problem to the product of Pr(S) and Pr(T|S), where Pr(S) is the language model probability of S (S is the right sentence in that place) and Pr(T|S) is the translation probability of T given S. In other words, we are seeking the most likely translation given how correct a candidate translation is and how well it fits in the context.

$$\Pr(S|T) = \frac{\Pr(S)\Pr(T|S)}{\Pr(T)}$$

Neural Machine Translation :

Neural Machine Translation (NMT) is the most powerful algorithm developed till date for machine translation . While Google Translate is the leading industry example of NMT, tech companies all over the globe are going all in on NMT. This state-of-the-art algorithm is an application of deep learning in which massive datasets of translated sentences are used to train a model capable of translating between any two languages. With the vast amount of research in recent years, there are several variations of NMT currently being investigated and deployed in the industry. One of the older and more established versions of NMT is the Encoder Decoder structure.

Encoder - Decoder Model

The *encoder-decoder architecture* is a neural network design pattern. The encoder's role is to encode the inputs into state, which often contains several tensors. Then the state is passed into the decoder to generate the outputs. In machine translation, the encoder transforms a source sentence, e.g., "Hello world.", into state, e.g., a vector, that captures its semantic information. The decoder then uses this state to generate the translated target sentence, e.g., "Bonjour le monde.".

It was initially developed for machine translation problems, although it has proven successful at related sequence-to-sequence prediction problems such as text summarization and question answering.The approach involves two recurrent neural networks, one to encode the input sequence, called the encoder, and a second to decode the encoded input sequence into the target sequence called the decoder.

Encoder :

A stack of several recurrent units (LSTM or GRU cells for better performance) where each accepts a single element of the input sequence, collects information for that element and propagates it forward.

Decoder :

A stack of several recurrent units where each predicts an output y_t at a time step t. Each recurrent unit accepts a hidden state from the previous unit and produces and output as well as its own hidden state.

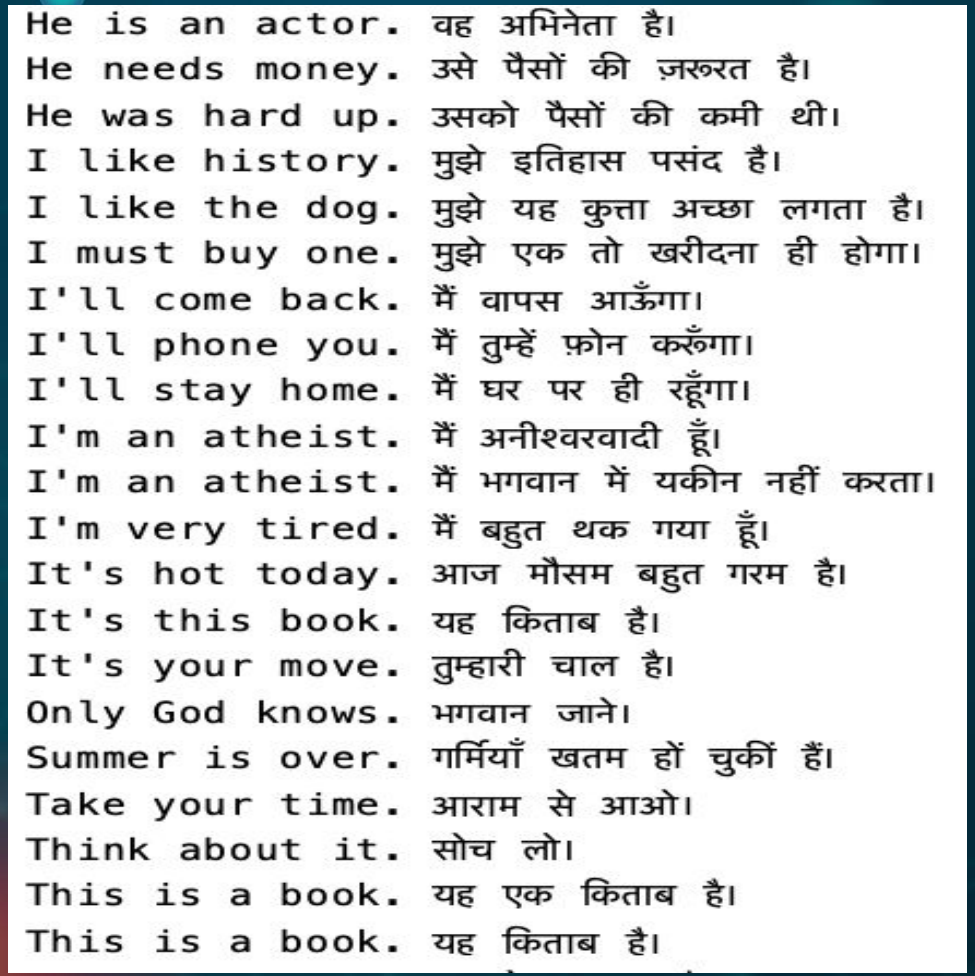
```
%tensorflow_version 1.x
```

```
import pandas as pd
import tensorflow
import gzip
import numpy as np
import h5py
from tensorflow.python.keras.models import Model
from tensorflow.python.keras.layers import Input, LSTM, Dense, InputLayer, Embedding
import numpy as np
import io

from sklearn.utils import shuffle
from sklearn.model_selection import train_test_split
import warnings
warnings.filterwarnings("ignore")

import string
from string import digits
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.utils import shuffle
```

Libraries Used



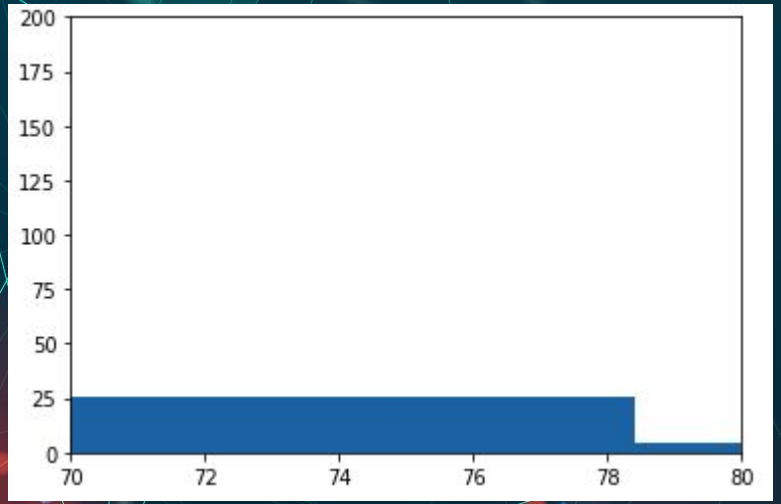
He is an actor.	वह अभिनेता है।
He needs money.	उसे पैसों की ज़रूरत है।
He was hard up.	उसको पैसों की कमी थी।
I like history.	मुझे इतिहास पसंद है।
I like the dog.	मुझे यह कुत्ता अच्छा लगता है।
I must buy one.	मुझे एक तो खरीदना ही होगा।
I'll come back.	मैं वापस आऊँगा।
I'll phone you.	मैं तुम्हें फ़ोन करूँगा।
I'll stay home.	मैं घर पर ही रहूँगा।
I'm an atheist.	मैं अनीश्वरवादी हूँ।
I'm an atheist.	मैं भगवान में यकीन नहीं करता।
I'm very tired.	मैं बहुत थक गया हूँ।
It's hot today.	आज मौसम बहुत गरम है।
It's this book.	यह किताब है।
It's your move.	तुम्हारी चाल है।
Only God knows.	भगवान जाने।
Summer is over.	गर्मियाँ खतम हों चुकीं हैं।
Take your time.	आराम से आओ।
Think about it.	सोच लो।
This is a book.	यह एक किताब है।
This is a book.	यह किताब है।

The Dataset Used

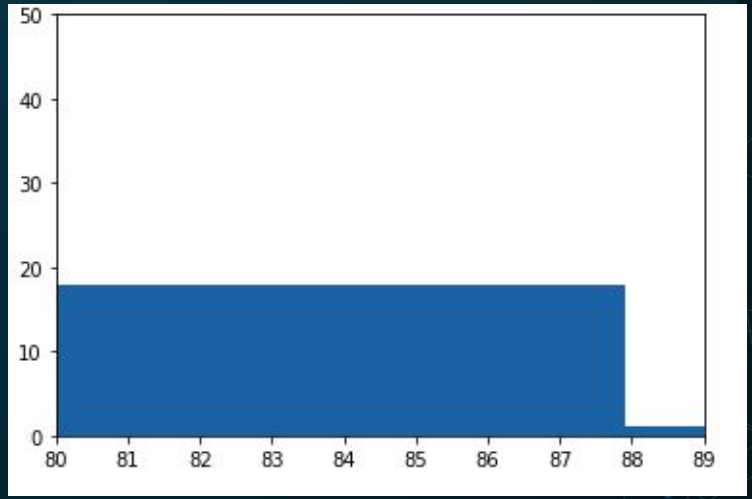
The dataset consist of 2869 English phrases along with their Hindi translations.

Input Length

After loading the data, we need to determine the input length of a sentence (best length) as we don't want sentences that are too long is because the computation becomes trickier for longer sentences and the performance also degrades. However we also want as many sentences in our dataset as possible. Thus it is important to choose the right length and discard sentences longer than this. As such we plot histograms to determine the best lengths.



For English



For Hindi

Model

Encoder

Encoder takes the English data as input and converts it into vectors that is passed to an LSTM model for training. We discard the encoder output and only keep the states.

```
encoder_eng = Input(shape=(None, encoder_tokens))
encoder = LSTM(latent_dim, return_state=True)
encoder_hin, state_h, state_c = encoder(encoder_eng)
encoder_states = [state_h, state_c]
```

Model

Decoder :

The decoder takes in Input the states of the encoder and the Hindi data points corresponding to the English input of Encoder. It trains an LSTM to produce the translated phrase in output. The decoder used softmax layer.

```
decoder_eng = Input(shape=(None, decoder_tokens))
decoder = LSTM(latent_dim, return_sequences=True, return_state=True)
decoder_hin, _, _ = decoder(decoder_eng,initial_state=encoder_states)
decoder_dense = Dense(decoder_tokens, activation='softmax')
decoder_hin = decoder_dense(decoder_hin)
```

Compiling The Model

```
model.compile(optimizer='rmsprop', loss='categorical_crossentropy')
```

Here we compile the model using ‘rmsprop’ optimizer as :

- RMSProp is an optimizer that tries to resolve Adagrad’s radically diminishing learning rates by using a moving average of the squared gradient. It utilizes the magnitude of the recent gradient descents to normalize the gradient.
- In RMSProp learning rate gets adjusted automatically and it chooses a different learning rate for each parameter.
- RMSProp divides the learning rate by the average of the exponential decay of squared gradients

Training The Model

```
model.fit([encoder_english, decoder_hindi], decoder_hindi,batch_size=batch_size,epochs=5,validation_split=0.2)
```

The model is fitted over a batch size of 64 and 5 epochs .

The model is trained on 2284 samples and validated on 572 samples (validation data is 20%).

OUTPUT :

```
English:        He is an actor.
Hindi:          वह अब चल रहा है।
English:        He needs money.
Hindi:          उसे पैसों की ज़रूरत है।
English:        He was hard up.
Hindi:          वह दरवाज़े खेल ले कुतवा भी लगता है।
English:        I like history.
Hindi:          मुझे इतिहास पसंद है।
English:        I like the dog.
Hindi:          मुझे यह कुत्ता अच्छा लगता है।
English:        I must buy one.
Hindi:          मुझे एक तो खरीदना ही होगा।
English:        I'll come back.
Hindi:          मैं दस बजे वापस आऊँगा।
English:        I'll phone you.
Hindi:          मैं तुम्हें फ़ोन करूँगा।
English:        I'll stay home.
Hindi:          मैं घर पर ही रहूँगा।
English:        I'm an atheist.
Hindi:          मैं भगवान में यकीन नहीं करता।
```

REFERENCES

- https://towardsdatascience.com/machine-translation-a-short-overview-91343ff39c9f
- https://medium.com/analytics-vidhya/machine-translation-encoder-decoder-model-7e4867377161
- https://en.wikipedia.org/wiki/Machine_translation
- https://machinelearningmastery.com/introduction-neural-machine-translation/
- https://medium.com/datadriveninvestor/overview-of-different-optimizers-for-neural-networks-e0ed119440c3

THANK YOU