NEURAL MACHINE TRANSLATION

Abstract

This project aims to develop machine translation deep learning model to translate French language text to English language text. In Natural Language processing, Machine translation aims to translate one human native language to other human native language by training the computers with machine learning algorithms. The model built for neural machine translation is encoder-decoders and consists of an encoder that encodes the French sentence into a fixed length vector from which a decoder generates a translation to English. From this model we achieve a translation performance comparable to the existing state of the art phrase based system on the task of French to English translation. We summarized the resources, algorithms, tools that were used in this model. Finally we conclude with a discussion of possible improvements to the model.

Introduction

Neural machine translation is a recently proposed approach to machine translation. Unlike the traditional statistical machine translation, the neural machine translation aims at building a single neural network that can be jointly tuned to maximize the translation performance. The models proposed recently for neural machine translation often belongs to a family of encoder-decoders and consists of an encoder that encodes a source sentence into a fixed-length vector from which a decoder generates a translation. **[1].**

Neural Machine Translation (NMT) is an end-to-end learning approach for automated translation, with the potential to overcome many of the weaknesses of conventional phrase-based translation systems. Unfortunately, NMT systems are known to be computationally expensive both in training and in translation inference. Also, most NMT systems have difficulty with rare words. These issues have hindered NMT's use in practical deployments and services, where both accuracy and speed are essential. In this work, we present GNMT, Google's Neural Machine Translation system, which attempts to address many of these issues. **[2].**

Neural machine translation is a radical departure from previous machine translation approaches. On the one hand, NMT employs continuous representations instead of discrete symbolic representations in SMT. On the other hand; NMT uses a single large neural network to model the entire translation process, freeing the need for excessive feature engineering. The training of NMT is end-to-end as opposed to separately tuned components in SMT. Besides its simplicity; NMT has achieved state-of-the-art performance on various language pairs **[3].**

As a data-driven approach to machine translation, NMT also embraces the probabilistic framework. Mathematically speaking, the goal of NMT is to estimate an unknown conditional distribution $P(y|x)$ given the dataset D, where x and y are random variables representing source input and target output, respectively. We strive to answer the three basic questions of NMT:

Modelling - How to design neural networks to model the conditional distribution?

Inference - Given a source input, how to generate a translation sentence from the NMT model?

Learning - How to effectively learn the parameters of NMT from data? **[4].**

Encoder-It accepts a single element of the input sequence at each time step, process it, collects information for that element and propagates it forward.

Intermediate *vector*- This is the final internal state produced from the encoder part of the model. It contains information about the entire input sequence to help the decoder make accurate predictions.

Decoder- given the entire sentence, it predicts an output at each time step **[5].**

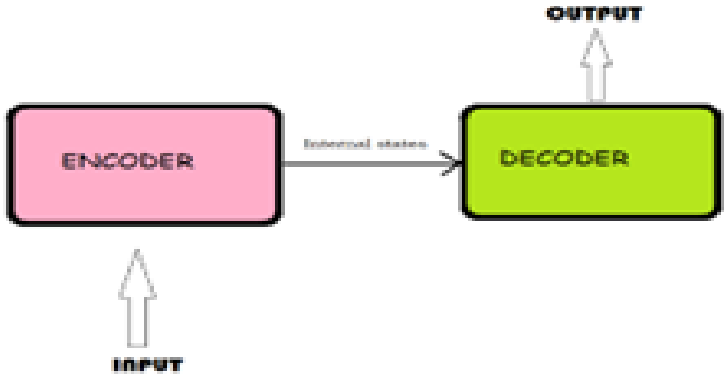


Fig.1 Encode Decoder model

Model Experiment:

Dataset

The data used in these projects is from Tatoeba Project data [6]. The dataset consist of 190206 translated sentences from French to English. From those sentences the outlier texts are identified and removed to get good quality of data for training.

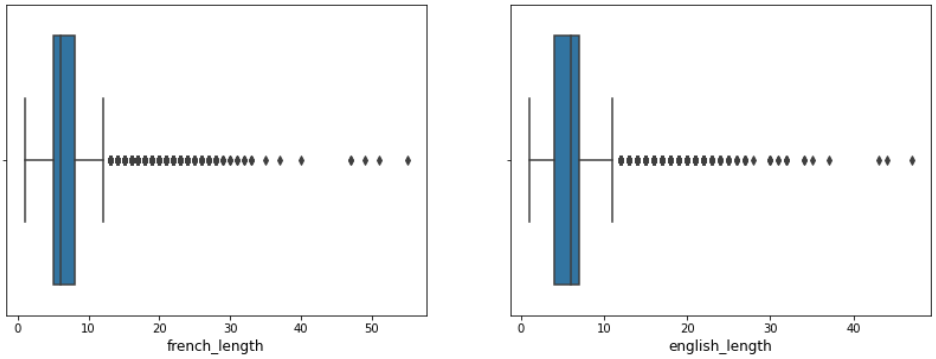


Fig.2. Data Outliers

Text Pre-processing

The data needs some minimal cleaning before being used to train a neural translation model.

Looking at some samples of text, some minimal text cleaning may include:

- Tokenizing text by white space.
- Normalizing case to lowercase.
- Removing punctuation from each word.
- Removing non-printable characters.
- Converting French characters to Latin characters.
- Removing words that contain non-alphabetic characters.

Feature Extraction

Machine learning algorithms operate on a numeric feature space, expecting input as a two-dimensional array where rows are instances and columns are features. In order to perform machine learning on text, we need to transform our documents into vector representations such that we can apply numeric machine learning. This process is called feature extraction or more simply vectorization, and is an essential first step toward language-aware analysis. **[11].**

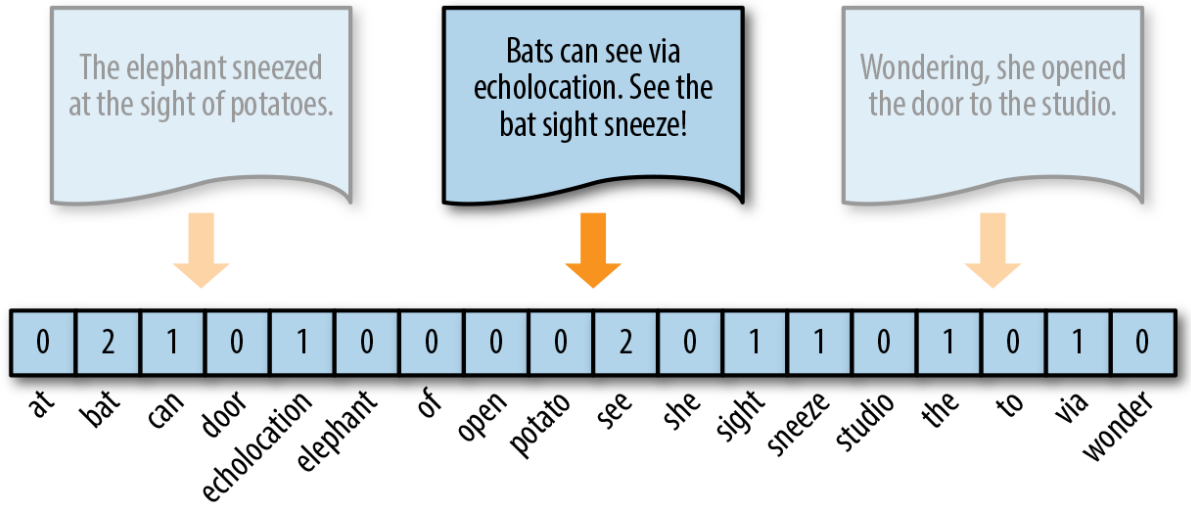


Fig.3 Tokenization

Teacher Forcing

Teacher forcing is a procedure for training RNNs with output-to hidden recurrence. It emerges from the maximum likelihood criterion. During training time model receives ground truth output y(t) as input at time t+1. The difference of teacher forcing are shown in below figure.

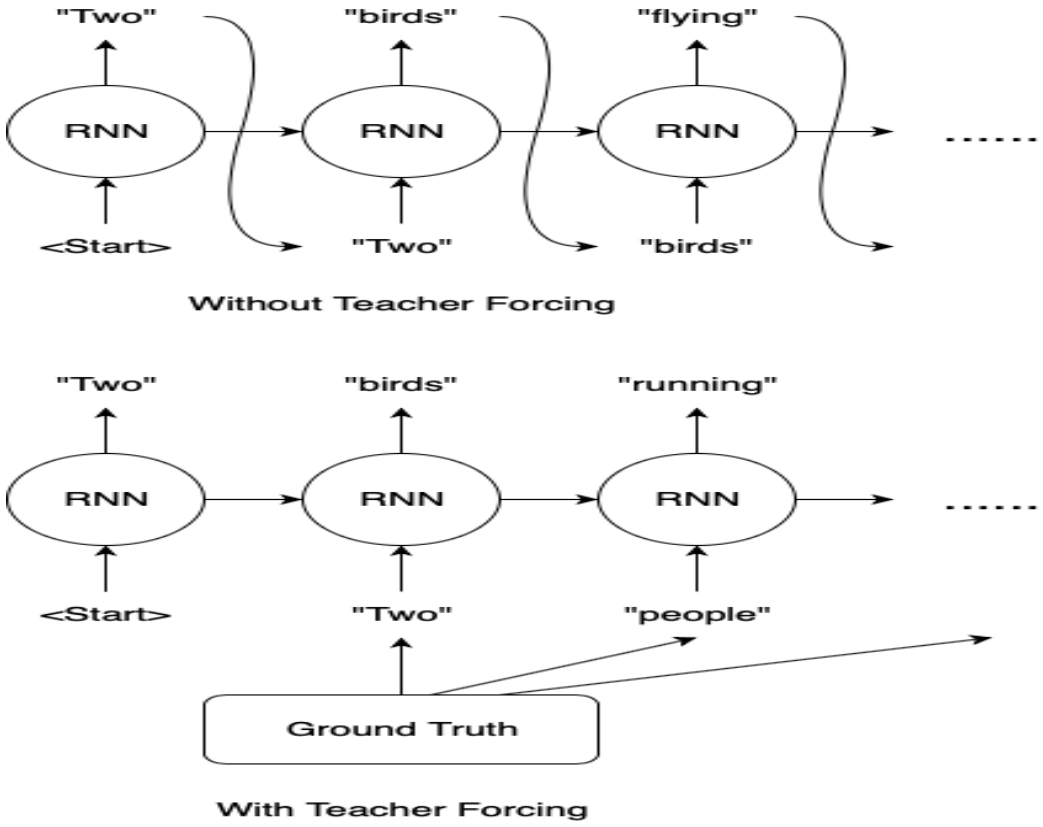


Fig.4 Teacher Forcing [7]

Model Architecture

In this model we used embedding layers and lstm layers for encoding and decoding the input text.

LSTM Architecture [10]

Long Short Term Memory (**LSTM**) is a special Recurrent Neural Network architecture, which was originally conceived by Hochreiter and Schmidhuber in 1997. This type of neural network has been recently rediscovered in the context of deep learning, because *it* is free from the problem of vanishing gradients, and offers excellent results and performance. The networks that are LSTM-based are ideal for prediction and classification of temporal sequences, and are replacing many traditional approaches to deep learning.

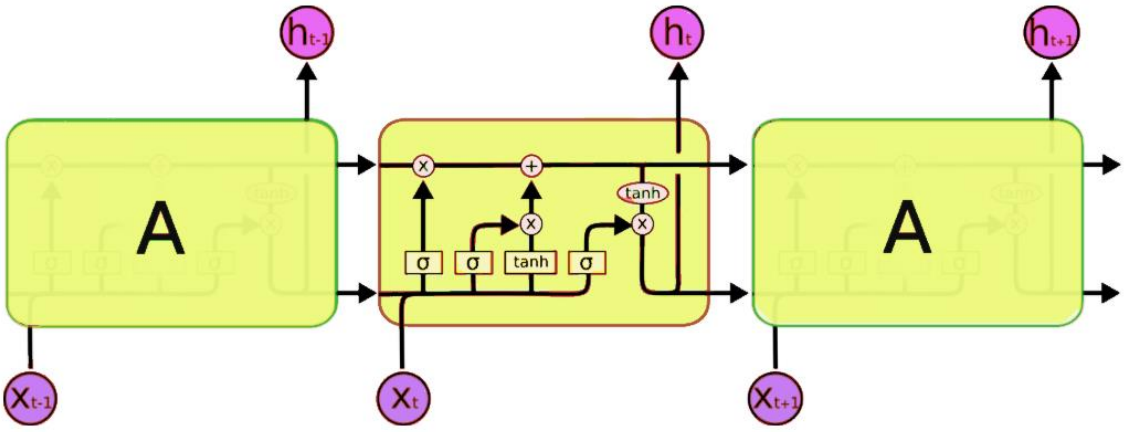


Fig.5 LSTM Architecture

Model training

A hyper parameter is a parameter whose value is used to control the learning process.

The below hyper parameters should be modified to improve the model performance.

- Number of nodes and hidden layers
- Number of units in a dense layer
- Activation function
- Learning rate
- Momentum
- Number of epochs
- Batch size

Model Summary

Trainable parameters are those which value is adjusted/modified during training as per their gradient. In Batch Normalization layer we have below mentioned trainable parameters:

Gamma: It's a scaling factor

Beta: a learned offset factor

Non trainable parameters are those which value is not optimized during the training as per their gradient. In batch normalization layer, we have below trainable parameters:

Mean

Standard deviation

```
Model: "encoder_decoder"
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 encoder (Encoder)            multiple                  4321792

 decoder (Decoder)            multiple                  3177728

 dense (Dense)                multiple                  7474410

=================================================================
Total params: 14,973,930
Trainable params: 14,973,930
Non-trainable params: 0
```

Fig.6 Model summary

Model Validation:

Model validation is the process of determining whether the model accurately represents the behaviour of the system. Model validity should be evaluated both operationally and conceptually. The figure shows the reduction in loss for every epoch.

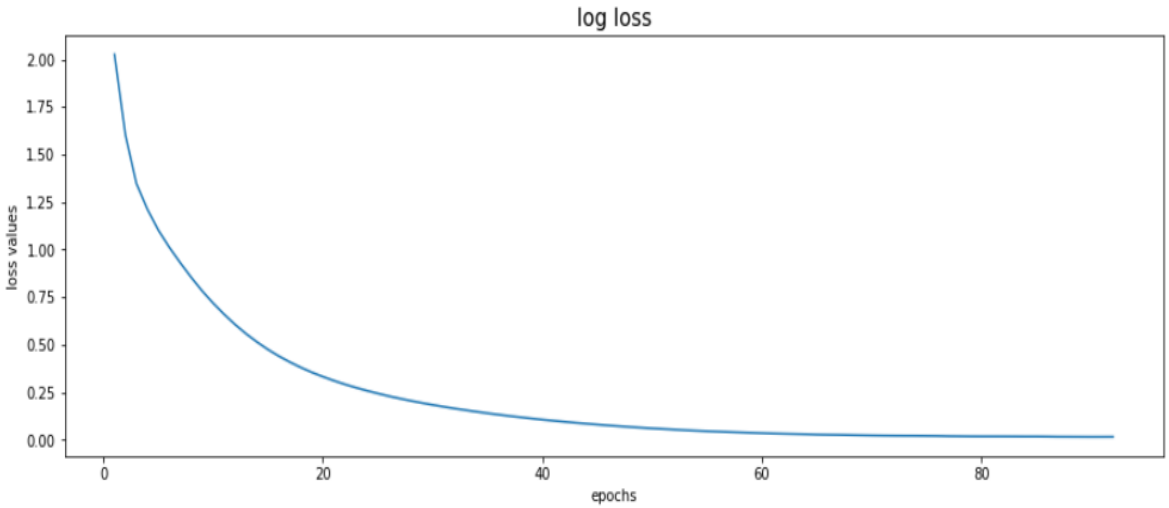


Fig.7. Model Validation against Loss Plot

Evaluation Metric

BLEU (Bilingual Evaluation Understudy) is an algorithm for evaluating the quality of text which has been machine-translated from one natural language to another. Quality is considered to be the correspondence between a machine's output and that of a human: "the

closer a machine translation is to a professional human translation, the better it is" – this is the central idea behind BLEU. BLEU was one of the first metrics to claim a high correlation with human judgements of quality, and remains one of the most popular automated and inexpensive metrics [9].

The Model score is about 68%.

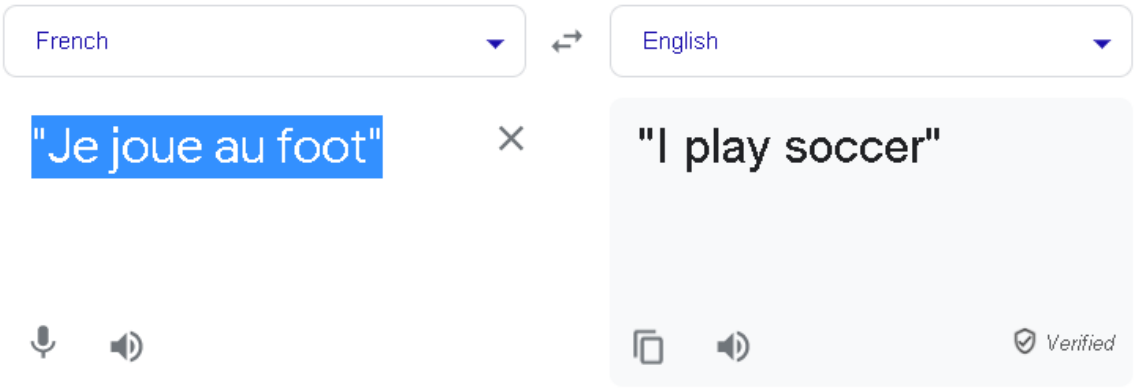
Results and Discussion:

Model prediction:

```
1 predict("Je joue au foot", threshold, model, fre_token, eng_token)
```

```
'i m playing football .'
```

Google Translator Output:



The above figure shows that, the model performing better. It needs little more improvement. The model can be optimized by tuning hyper parameters, adding more data for training, improvement in pre-processing.

Future works is to revealing insights on which hyper-parameters matter most in terms of performance, such as words processed per second, convergence rates, and translation accuracy, and provides insights on how to best achieve high-performing NMT systems.

GitHub Repository : https://github.com/OmkarAvinash/Neural-Machine-Translation-using-Deep-Learning

References:

1. Neural Machine Translation by Jointly Learning to Align and Translate
2. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation
3. Junczys-Dowmunt et al., 2016
4. https://www.sciencedirect.com/science/article/pii/S2666651020300024
5. https://medium.com/analytics-vidhya/machine-translation-encoder-decoder-model-7e4867377161
6. http://www.manythings.org/anki/
7. https://towardsdatascience.com/what-is-teacher-forcing-3da6217fed1c
8. https://www.maheshswami.com/2021/02/what-are-non-trainable-parameters.html
9. https://huggingface.co/spaces/evaluate-metric/bleu
10. http://colah.github.io/posts/2015-08-Understanding-LSTMs/
11. https://www.oreilly.com/library/view/applied-text-analysis/9781491963036/ch04.html