**Applications of NLP: Machine Translation**

Resmita Raju, 1948052, 3MDS.

**Introduction**

Natural Language Processing (NLP) has become an evolving technology that has taken various forms of Artificial Intelligence at present times. It has succeeded in establishing a smooth as well as interactive interface between humans and machines. This has become a top priority for today’s and tomorrow’s increasingly intellectual submissions. In this document we are going to see one of the applications of NLP is discussed.

**Machine Translation**

Translating one source language or text into another language is what is meant by Machine Translation. This is one of the prominent existing application of NLP. It is a popular topic in research with different methods being created, like rule-based, statistical and example-based machine translation. Neural networks have made a major contribution in the field of NLP too.

**Understanding the Problem Statement**

The objective here is to convert a German sentence to English using a Neural Machine Translation(NMT) system. The German-English sentence pairs data from <http://www.manythings.org/anki/> is used for the study.

**Methodology**

**Introduction to Sequence-to-Sequence (Seq2Seq) Modeling**

Sequence-to-Sequence (seq2seq) models are used for a variety of NLP tasks, such as text summarization, speech recognition, DNA sequence modeling, among others. The aim is to translate given sentences from one language to another. Here, both the input and output are sentences. In other words, these sentences are a sequence of words going in and out of a model. This is the basic idea of Sequence-to-Sequence modeling. The figure below tries to explain this method.

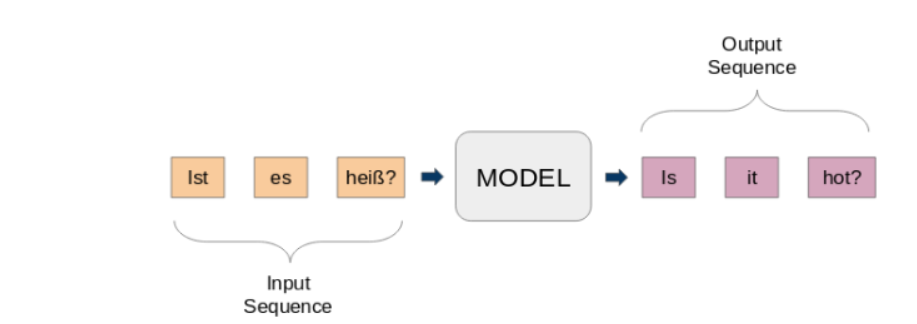


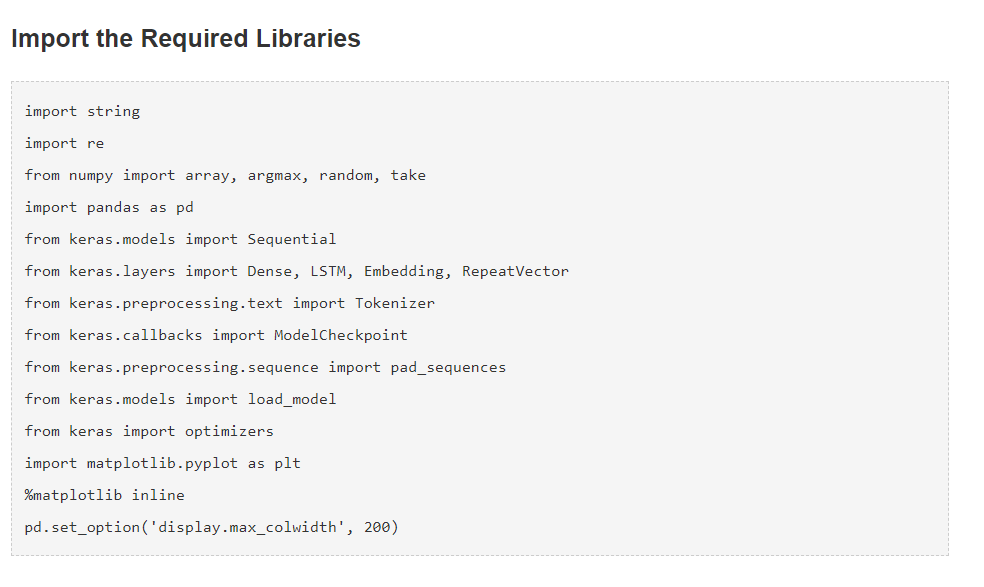
Fig 1. Model Method

**A typical seq2seq model has 2 major components –**

a) an encoderb) a decoder

Both these parts are essentially two different recurrent neural network (RNN) models combined into one giant network.

**Implementation in Python using Keras**



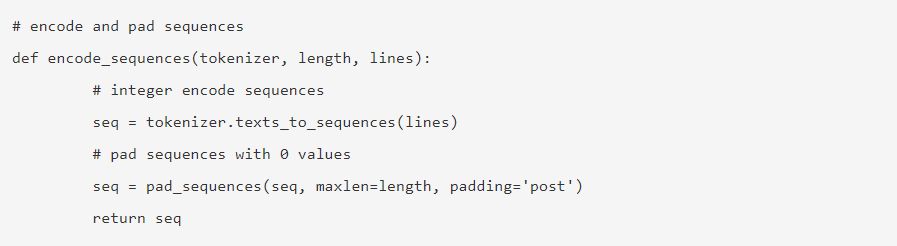
The next step to **read the data into the IDE.** The data consist of English and German sentences which needs to be split. T**he actual data contains over 150,000 sentence-pairs. However, the first 50,000 sentence pairs to reduce the training time of the model is only used.**

After loading the data, we can now move on to **text pre-processing**. Quite an important step in any project, especially so in NLP. The data we work with is often unstructured so there are certain things we need to take care of before jumping to the model building part. Here, in **text cleaning** the punctuations are removed, and the text is converted to lower case. It is to be noted that Seq2Seq model requires the conversion of both the input and the output sentences into integer sequences of fixed length. So, the next step is **text conversion**. For this we find the length of the sentences. We will capture the lengths of all the sentences in two separate lists for English and German, respectively. It is seen that the maximum length of the German sentences is 11 and that of the English phrases is 8.

Next step is to **vectorize the text** data by using Keras’s *Tokenizer()* class. It will turn sentences into sequences of integers. We can then **pad those sequences** with zeros to make

all the sequences of the same length. Tokenizers for both the German and English sentences are prepared for the dataset.

The below code block contains a function to prepare the sequences. It will also perform sequence padding to a maximum sentence length as mentioned above.



**Model Building**

We will now split the data into train and test set for model training and evaluation, respectively. Then the sentences are encoded. We will encode German sentences as the input sequences and English sentences as the target sequences. It will be done for both train and test datasets.

Now, let us define our Seq2Seq model architecture. We are using an Embedding layer and an LSTM layer as our encoder and another LSTM layer followed by a Dense layer as the decoder.

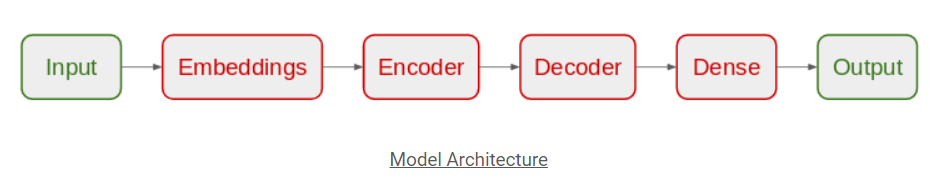
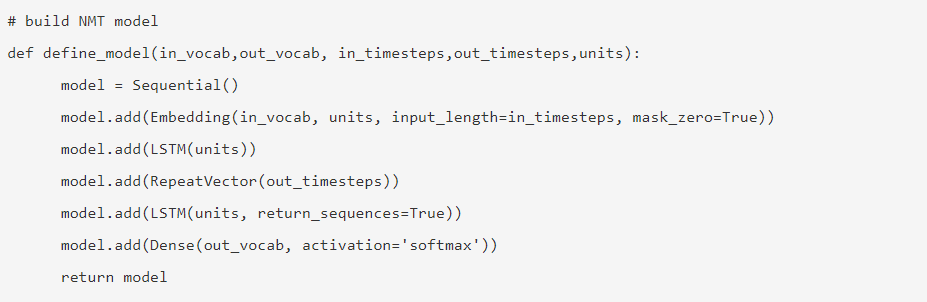
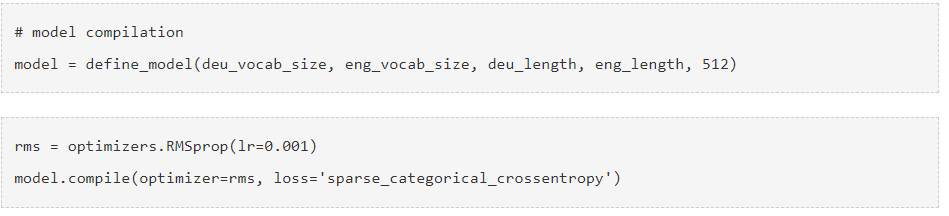


Fig2. Model Architecture





The model is then trained for 30 epochs and with a batch size of 512 with a validation split of 20%. 80% of the data will be used for training the model and the rest for evaluating it. Then we can compare the training loss and the validation loss of the model.

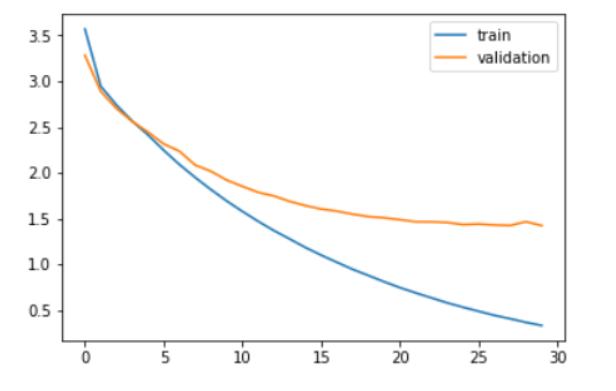


Fig.3 Plot of Training loss vs Validation loss

As seen in the above plot, the validation loss stopped decreasing after 20 epochs. Finally, we can load the saved model and make predictions on the unseen data – testX. These predictions are sequences of integers. We need to convert these integers to their corresponding words and then to English text.

Now, let us put the original English sentences in the test dataset and the predicted sentences in a data frame and print some actual vs predicted instances to see how our model performs:



Fig.4 Table for predicted words.

**Conclusion:**

It is observed that Seq2Seq model does a really decent job in translating. But there are several instances where it misses out on understanding the key words. For example, it translates “im tired of boston” to “im am boston”.

These are the challenges you will face on a regular basis in NLP. But these aren’t immovable obstacles. We can mitigate such challenges by using more training data and building a better (or more complex) model.

**References.**

[1]<https://github.com/prateekjoshi565/machine_translation/blob/master/german_to_english.ipynb>

[2] <https://www.analyticsvidhya.com/blog/2019/01/neural-machine-translation-keras/>

[3]<https://www.tutorialspoint.com/natural_language_processing/natural_language_processing_applications_of_nlp.htm>

[4]<https://github.com/sjayakum/nlp-machine-translation/blob/master/Hindi%20to%20English/BLUEScore.ipynb>

[5] S. Saini and V. Sahula, "A Survey of Machine Translation Techniques and Systems for Indian Languages," *2015 IEEE International Conference on Computational Intelligence & Communication Technology*, Ghaziabad, 2015, pp. 676-681, doi: 10.1109/CICT.2015.123.