Bahu-Bhashi : The Speech Translator

Jhanvi Shah — shah.jh@husky.neu.edu
Gaurang Davda — davda.g@husky.neu.edu
Yashasvi Jariwala — jariwala.ya@husky.neu.edu

CSYE 7245, Fall 2018, Northeastern University

Abstract - In this global world we have to interact with a lot of people. They might not speak the language we understand and might not understand the language we speak. In this era, if we have a platform which translates the audio input from one language to another will be a great help. Many companies like Google, Amazon and Microsoft have worked on immense data and word embeddings to make Text-to-Text translation possible. These companies have spent millions of dollars on these problems and developed APIs for us people. The system basically aims at converting an audio input given in one language i.e. English to the audio output in target language i.e. German. The audio input is restricted to 1 word in English and it is then translated to German and spoken in German using Google TTS engine.

1. Introduction - Motivation

Recently many MNC's are facing an issue where a customer of a particular region calls in and the customer experiences long wait hours as all the representatives are busy, while their call centers in other regions are not experiencing busy wait hours. To mitigate this issue, there could be a system designed which takes in an audio input in one language and translates it into another language. This solution could be used where, if the call centers in one region are facing high call volumes the calls could be directed to the regions facing lower call volumes and BahuBhashi could be used so that person on both ends can converse in their own language and find a solution to their issue.

Background

The main goal behind our project is to ease a conversation and to break down the language barriers. Speech translation technology is basically being able to speak in one language and have those words translated into another language. In MNC's people have to interact with people speaking various languages. It might be easier if there is a platform that converts the input language, for example English, into language spoken by the other person, for example German. The whole process of speech translation technology is a mix up of three technologies: technology to recognize speech (speech recognition); technology to translate the recognized words (language translation); and technology to synthesize speech in the other person's language (speech synthesis). We have used the CNN/LSTM model for language translation from English language to German language.

2. Dataset

The paper implements two different datasets. The first dataset is the Google's Speech Command dataset in English Language. This dataset has 30 different words spread across 64,000 wav files recorded by different people. The second dataset is the European Parliament Proceedings Parallel Corpus 1996-2011. It has around 2,00,000 sentences in English and its equivalent German translation.

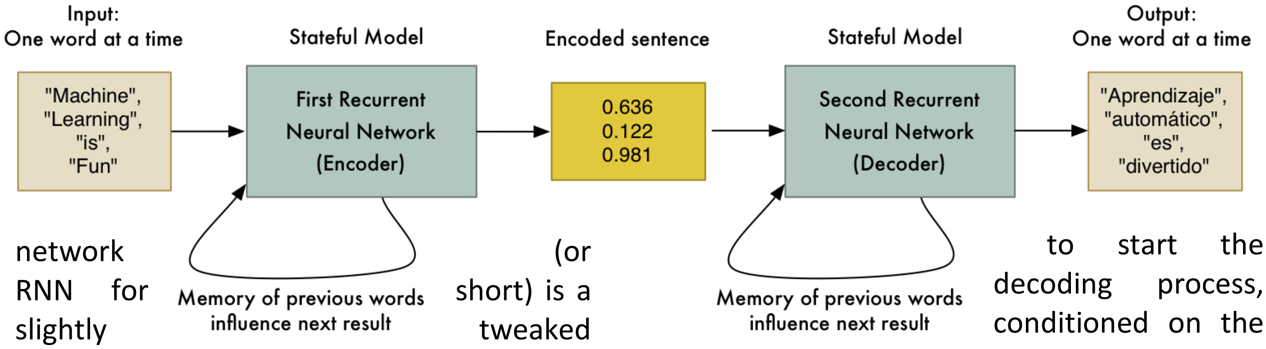
3. Methodology

Speech-to-text was modelled by using the features from the audio files. Package 'librosa' is used to load the audio files with a particular sample rate. Each of the samples are used as a feature input to a Convolution 1D layer. It is followed by a series of Batch Normalization, Max Pooling and Dropout layers. A complex model was built with around 16 sets of Convolution layers and 3 fully connected layers. The audio samples, when passed through this network, let the network extract relevant information about the MFCCs features and the amplitude of the waveform and train itself for various samples of a same class. A softmax activation is used at the last layer to get the max probability out of all the classes. The model has close to 138,000,000 parameters and is a complex model.

Text-to-Text translation. A recurrent neural network (or RNN for short) is a slightly tweaked version of a neural network where the previous state of the neural network is one of the inputs to the next calculation. This means that previous calculations change the results of future calculations! Because the RNN has a "memory" of each word that passed through it, the final encoding that it calculates represents all the words in the sentence. To generate this encoding, the sentence is fed into the RNN, one word at time. The final result after the last word is processed and will be the values that represent the entire sentence.

This is a way to represent an entire sentence as a set of unique numbers. It is not known what each number in the encoding means, but it doesn't really matter. As long as each sentence is uniquely identified by it's own set of numbers, there is no need to know exactly how those numbers were generated. Now it is known how to use an RNN to encode a sentence into a set of unique numbers. How does that help? What if two RNNs were taken and hooked end-to-end? The first RNN could generate the encoding that represents a sentence. Then the second RNN could take that encoding and just do

The Long Short-Term Memory recurrent neural network is commonly used for the encoder and decoder. The encoder output that describes the source sequence is used to start the decoding process, conditioned on the words already generated as output so far. Specifically, the hidden state of the encoder for the last time step of the input is used to initialize the state of the decoder. The model is trained on these datasets using LSTM or GRU. The



encoder-decoder model is a way of organizing recurrent neural networks to tackle sequence-to-sequence prediction problems where the number of input and output time steps differ. The model was developed for the problem of machine translation, such as translating sentences in English to German.

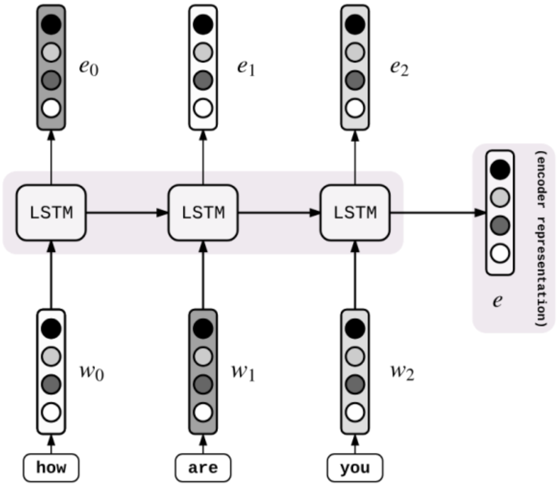
The model involves two sub-models, as follows:
Encoder: An RNN model that reads the entire source sequence to a fixed-length encoding.
Decoder: An RNN model that uses the encoded input sequence and decodes it to output the target sequence.

Text-to-Speech. The googles Text-To-Speech is used to convert the translated text to speech. The Playsound is used to play that audio.

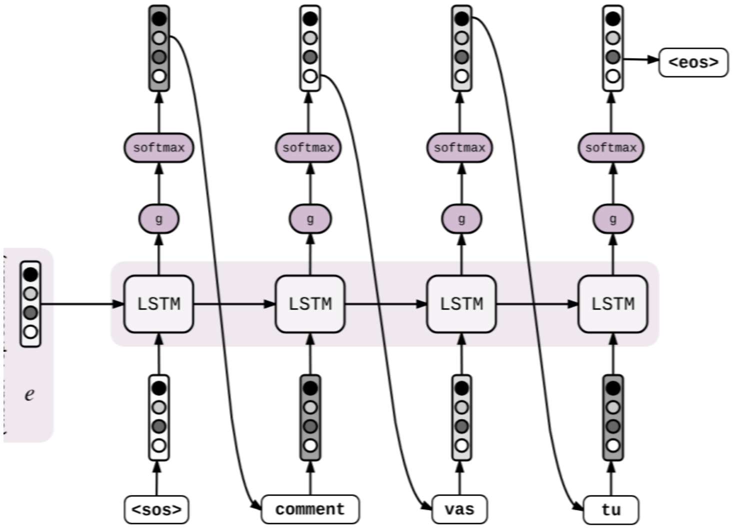
4. **Architecture**

There are three steps to convert the audio input from on language to another. Speech-To-Text, Text-To-Text, Text-To-Speech. For Text-To-Text we have used the dataset: European Parliament Proceedings Parallel Corpus ("http://www.statmt.org/europarl/v7/de-en.tgz"). There are three python files model class + necessary for preprocessing and training and one Jupyter notebook that shows the results of using the network. There is a *'sequence2sequence'* model which uses bidirectional RNN (LSTM or GRU). The encoder-decoder model provides a pattern for using recurrent neural networks to address challenging sequence-to-sequence prediction problems such as machine translation.

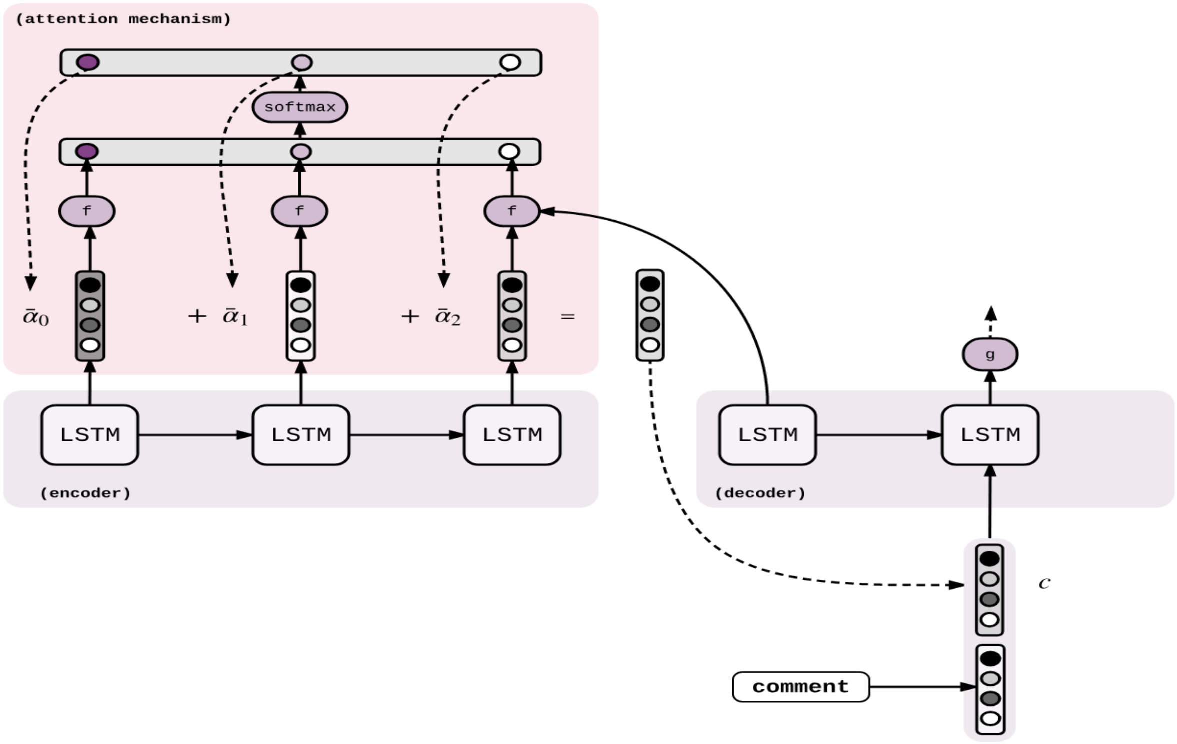
Encoder



Decoder



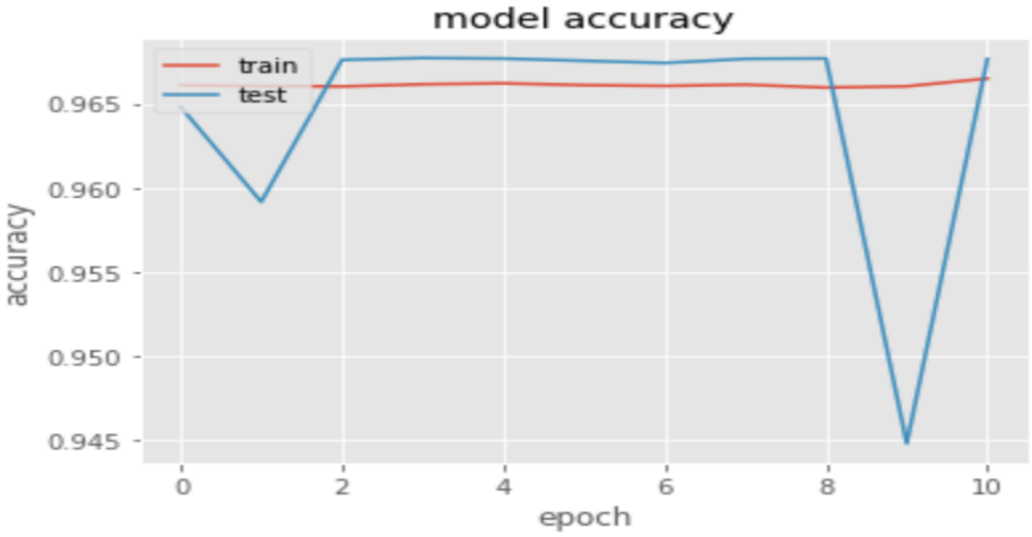
Attention is an extension to the encoder-decoder model that improves the performance of the approach on longer sequences. Adam optimizer is used to optimize the network and to generate the desired output. Beam Search or Greedy Decoding



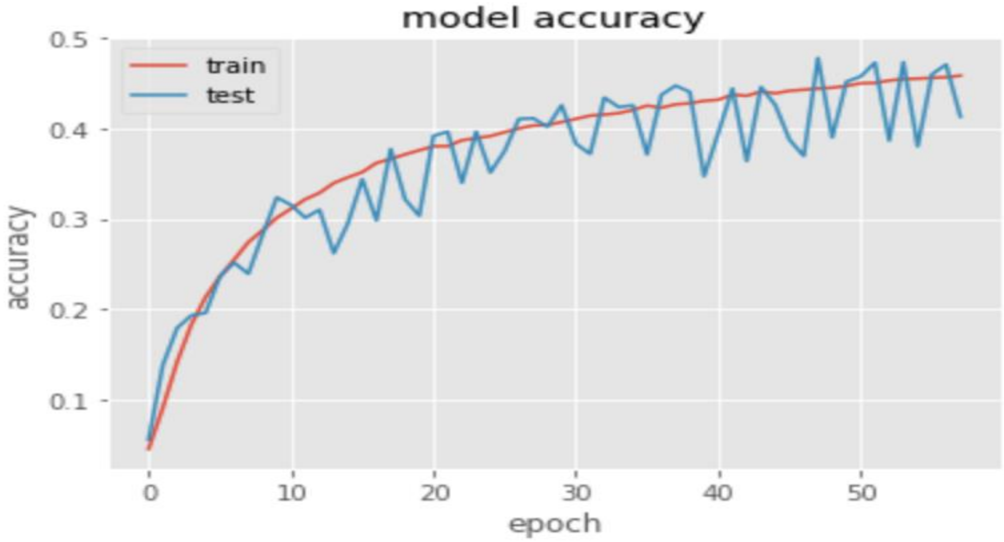
Results

Speech-to-text:-

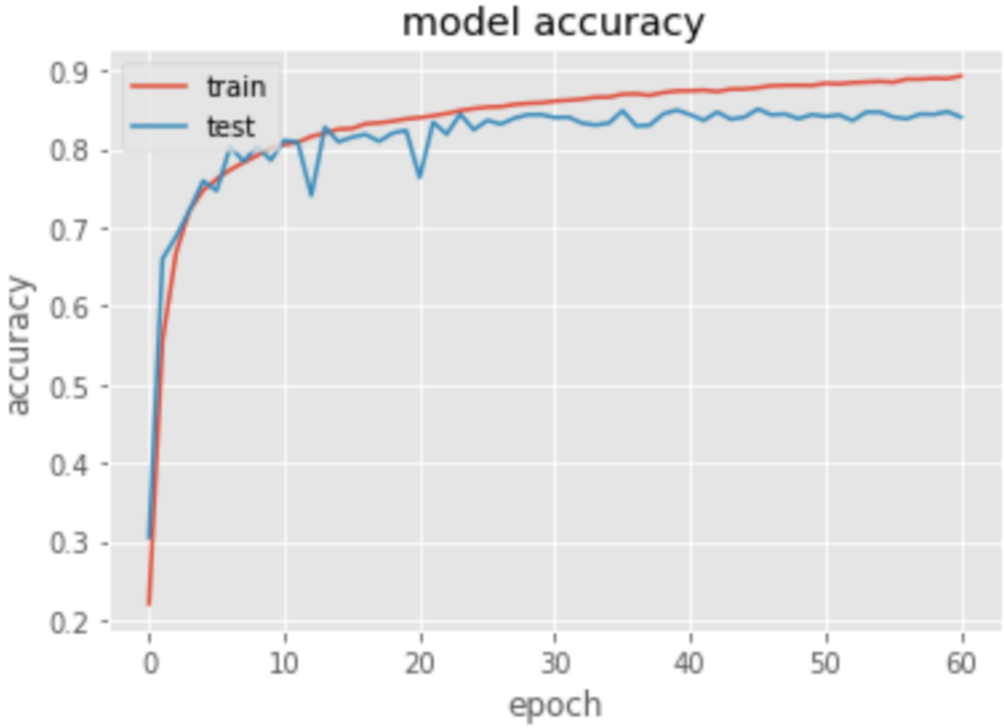
Model 1:

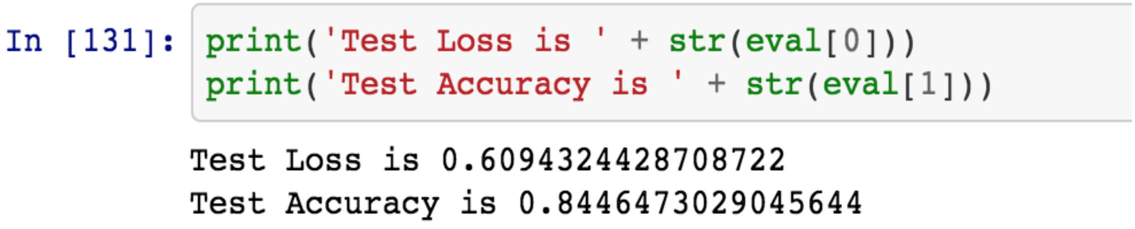


Model 2:



Model 3:



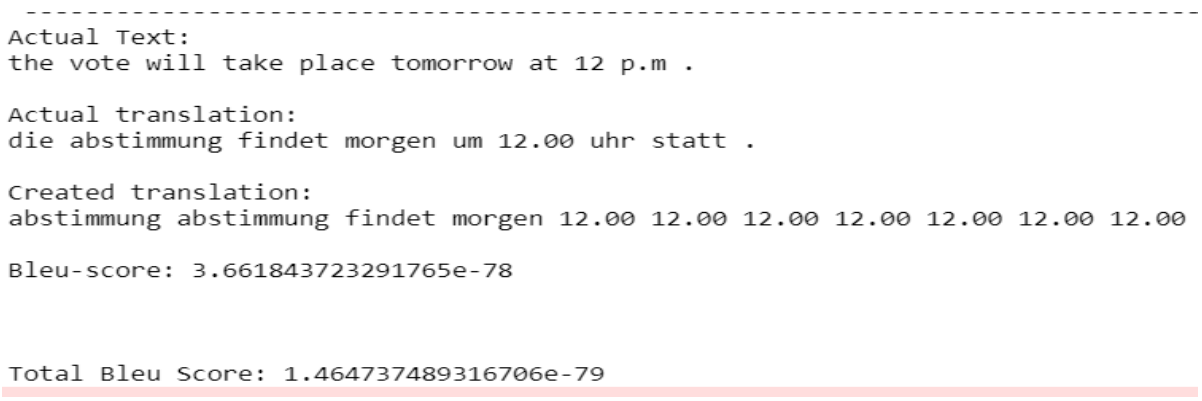


```
In [131]: print('Test Loss is ' + str(eval[0]))
          print('Test Accuracy is ' + str(eval[1]))

          Test Loss is 0.6094324428708722
          Test Accuracy is 0.8446473029045644
```

As it can be seen that the Model 3 is the best speech to text model. Below is its test accuracy.

Text-to-Text model:



```
 ------------------------------------------------------------------------------
Actual Text:
the vote will take place tomorrow at 12 p.m .

Actual translation:
die abstimmung findet morgen um 12.00 uhr statt .

Created translation:
abstimmung abstimmung findet morgen 12.00 12.00 12.00 12.00 12.00 12.00 12.00

Bleu-score: 3.661843723291765e-78


Total Bleu Score: 1.464737489316706e-79
```

As it can be seen from the Bleu score that the text-to-text model has a lot of scope for improvement as the Bleu score of 0 is the worst and 1 is the best.

Conclusions:

There is a lot of scope in improving the text-to-text model. A better approach to solving the word embedding problem or maybe by using pre-trained glove model might improve the accuracy. However, the Speech-to-text model seems to be working properly. A proper evaluation of the model can be done by retraining the model weights on a larger dataset with continuous audio data and not just words.

References:

1. https://machinelearningmastery.com/global-attention-for-encoder-decoder-recurrent-neural-networks (Gentle Introduction to Global Attention for Encoder-Decoder Recurrent Neural Networks)
2. https://guillaumegenthial.github.io/sequence-to-sequence.html (Seq2Seq with Attention and Beam Search)
3. http://www.statmt.org/europarl/v7/de-en.tgz (European Parliament Proceedings Parallel Corpus 1996-2011)
4. https://pypi.org/project/gTTS/
5. https://pypi.org/project/playsound/
6. https://github.com/tensorflow/nmt (Neural Machine Transfer Tensorflow Implementation)
7. https://github.com/thomasschmied/Neural_Machine_Translation_Tensorflow (Neural Machine Translation Implementation)
8. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.472.1019&rep=rep1&type=pdf (Overcoming the language barrier by Speech Translation Technology)
9. https://pdfs.semanticscholar.org/0fa1/911622a6c0a3dd43fefbdf2695ebdb7e10fa.pdf (Speech to Speech Translation – a review)
10. https://www.cs.cmu.edu/~awb/papers/eurospeech2003/speechalator.pdf (Speechalator - two way speech to speech translation on a consumer PDA)