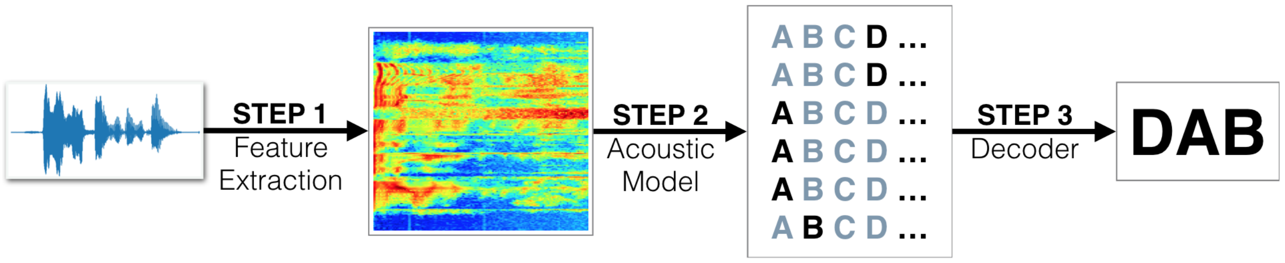
Speech Recognition

Disclaimer:

All the reports this far have been made by **Ashutosh Upadhye**. Siddhardha SST helped in Paper Review. Rahul Dhawan was busy with other assignments. I volunteered to do all the reports myself.

Introduction

We will build a deep neural network that functions as end-to-end speech recognition pipeline. The completed pipeline will accept raw audio as input and return a predicted transcription of the spoken language. The full pipeline is summarized in the following figure.



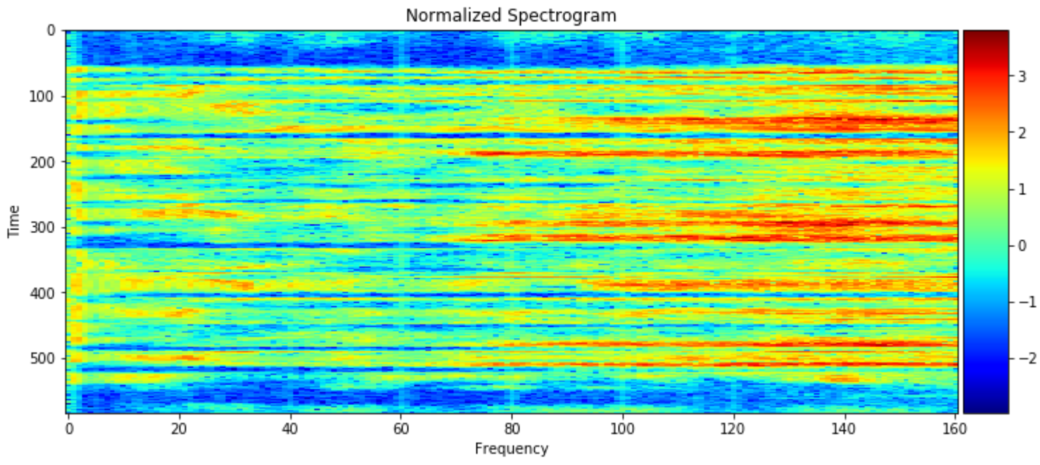
Step 1: Acoustic Features for Speech Recognition

STEP 1 is a pre-processing step that converts raw audio to one of two feature representations that are commonly used for ASR.

Spectograms:

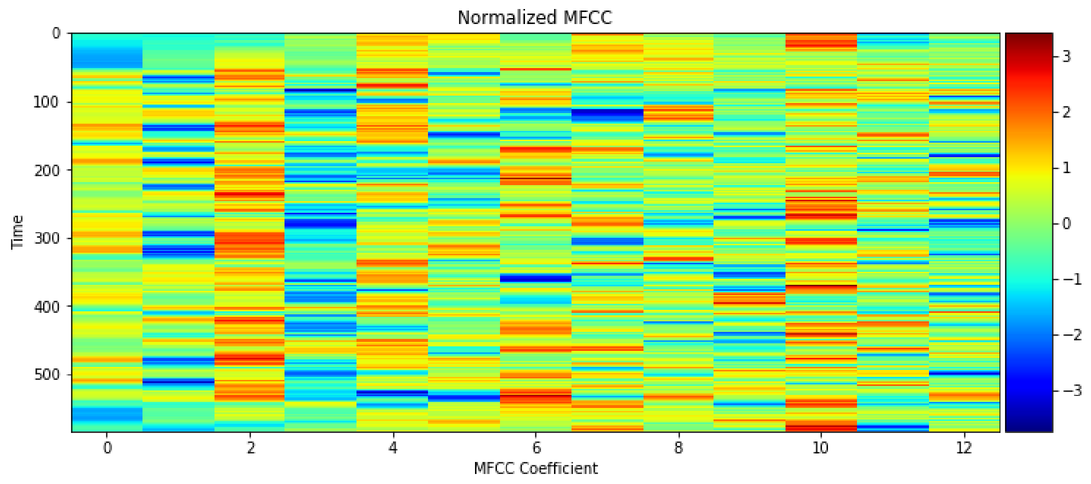
A spectrogram is a visual representation of the spectrum of frequencies of sound or other signal as they vary with time. Spectrograms are sometimes called spectral waterfalls, voiceprints, or voicegrams.

Spectrograms may created from a time-domain signal in one of two ways: approximated as a filterbank that results from a series of band-pass filters, or calculated from the time signal using the Fourier transform. These two methods actually form two different time–frequency representations, but are equivalent under some conditions.



Mel-Frequency Cepstral Coefficients (MFCCs):

In sound processing, the mel-frequency cepstrum (MFC) is a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency.



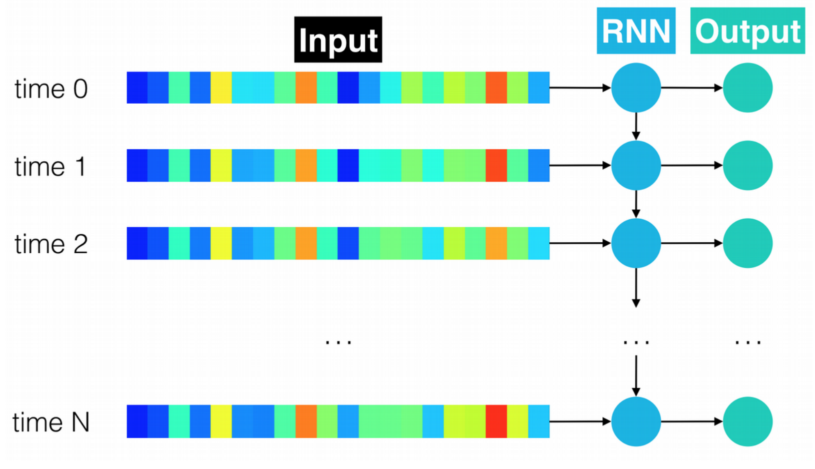
MFCCs are commonly used as features in speech recognition[6] systems, such as the systems which can automatically recognize numbers spoken into a telephone.

Step 2: Deep Neural Networks for Acoustic Modeling

STEP 2 is an acoustic model which accepts audio features as input and returns a probability distribution over all potential transcriptions.

Recurrant Neural Networks:

A recurrent neural network (RNN) is a class of artificial neural network where connections between units form a directed graph along a sequence. This allows it to exhibit dynamic temporal behavior for a time sequence. Unlike feedforward neural networks, RNNs can use their internal state (memory) to process sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition.



Other Models:

Others models can be applied and tested.

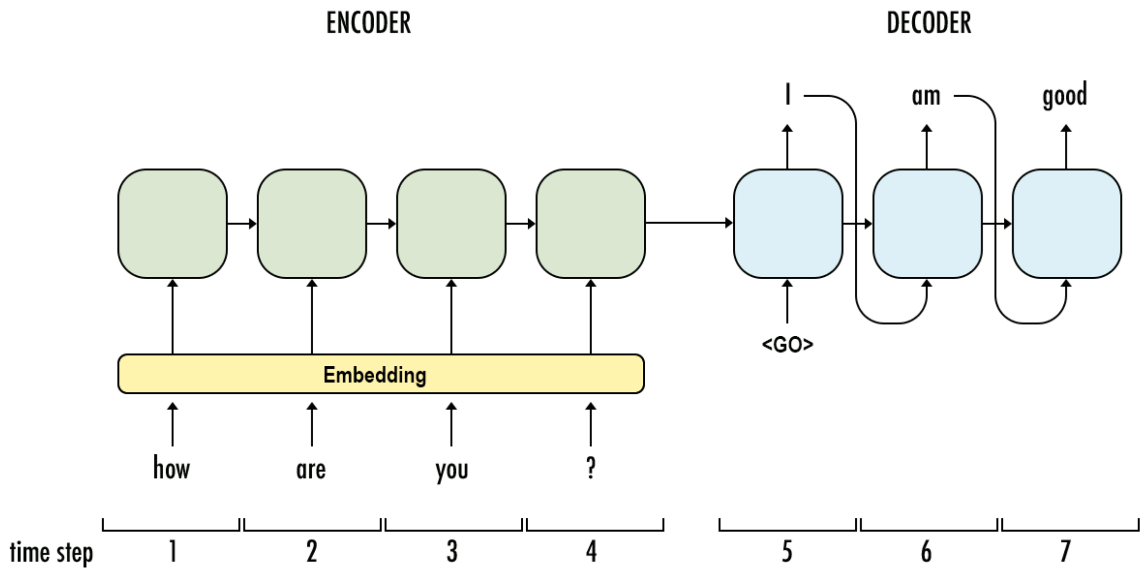
Step 3: Obtaining Predictions

STEP 3 in the pipeline takes the output from the acoustic model and returns a predicted digit. This is rather trivial.

Natural Language Processing

Introduction

A chatbot could be summed up as a blackbox which takes as the input a sequence of words and gives a sequence of words as output. It can be thought of as a sequence to sequence decoder.



Step 1: Preprocessing

We have decided to use Cornell Movie Dialogue Corpus dataset. This corpus contains a large metadata-rich collection of fictional conversations extracted from raw movie scripts.

Vocaboulary Extraction:

We have to extract the vocablulary from the training dataset and map each word to an integer.

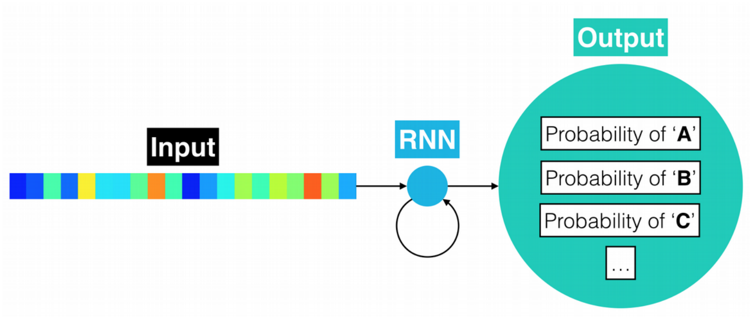
Tokenizer:

All the words in the speech have to be tokenised in order to generate a sequence of words as input.

Step 2: Model

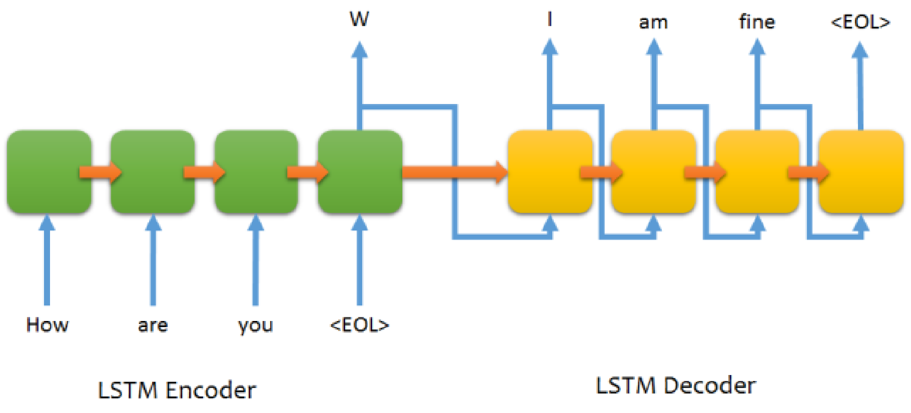
Recurrent Neural Networks:

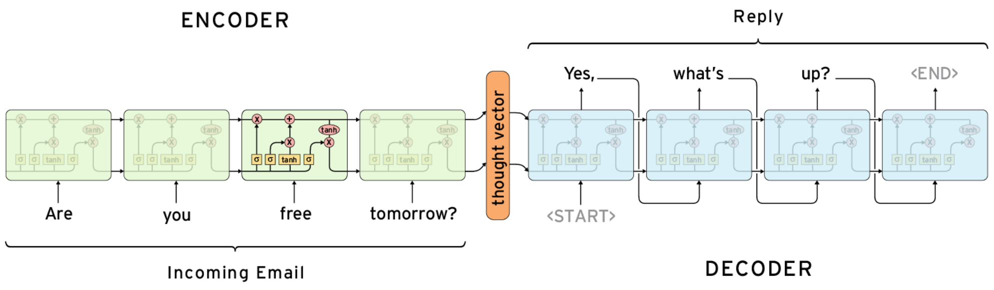
A recurrent neural network (RNN) is a class of artificial neural network where connections between units form a directed graph along a sequence. This allows it to exhibit dynamic temporal behavior for a time sequence. Unlike feedforward neural networks, RNNs can use their internal state (memory) to process sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition.



Sequence to Sequence model: RNN Encoder-Decoder

RNN Encoder-Decoder that consists of two recurrent neural networks. One RNN encodes a sequence of symbols into a fixed-length vector representation, and the other decodes the representation into another sequence of symbols.





There are various proposals made in various papers. Reversing the input sequence is proven to be better than the ordinary one. Padding the input supposedly yields better output and so on. These could be tried out. Sequence to Sequence model is the best for chatbot kind of applications.