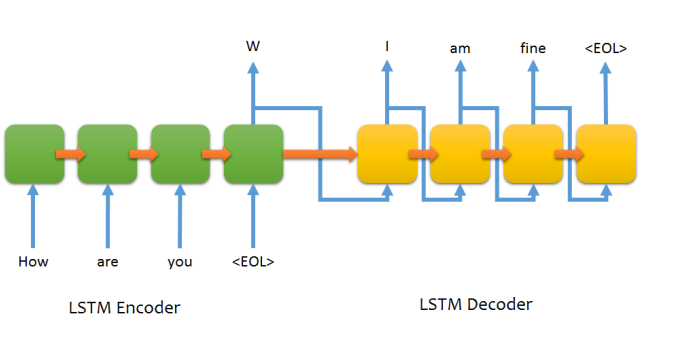
Sequence to Sequence Modelling

Sequence to Sequence Modelling is used in application like text classification, machine translation, and chat bot and so on. They are constructed by joining two sets of RNN, called as an encoder and decoder. Output of encoder is stored in a context vector. This vector acts as hidden state input values to the decoder.



Input is fed to the encoder and decoder with the help of a word embedding matrix. This embedding matrix is built based on Skip Gram/ Continuous Gram model. Input is fed to the decoder only during training phase. During the test phase, accuracy of model is tested based on only the output of context vector. A technique called as attention mechanism is used to decide how much of the input affects the output, i.e. input that doesn’t contribute to o/p is ignored.

Accuracy of model can be increased by using stacks of LSTMS on top of each other. Though this increases training time, accuracy of o/p is guaranteed. A typical Sequence to Sequence to has a very large training dataset, general in order of 100,000.

Pseudocode:

* Build word embedding matrix of both encoder and decoder inputs. Word embedding built can be based upon Skip Gram/ Continuous Bag of Words model or any other similar model.
* Initialize the weights of RNN’s i/p unit, hidden unit and o/p unit for both encoder and decoder.
* Feed input to encoder sequentially and store it’s output in context vector.
* Output of context vector acts as hidden unit input to the decoder.
* At decoder, o/p is displayed after passing through every decoder input unit. While in encoder, o/p is collected only at end of encoder unit.
* Here, programming an encoder is similar to building an RNN for sentiment analysis purpose while programming an encoder is similar to building an RNN for text generation purpose