**Long Homework 4**

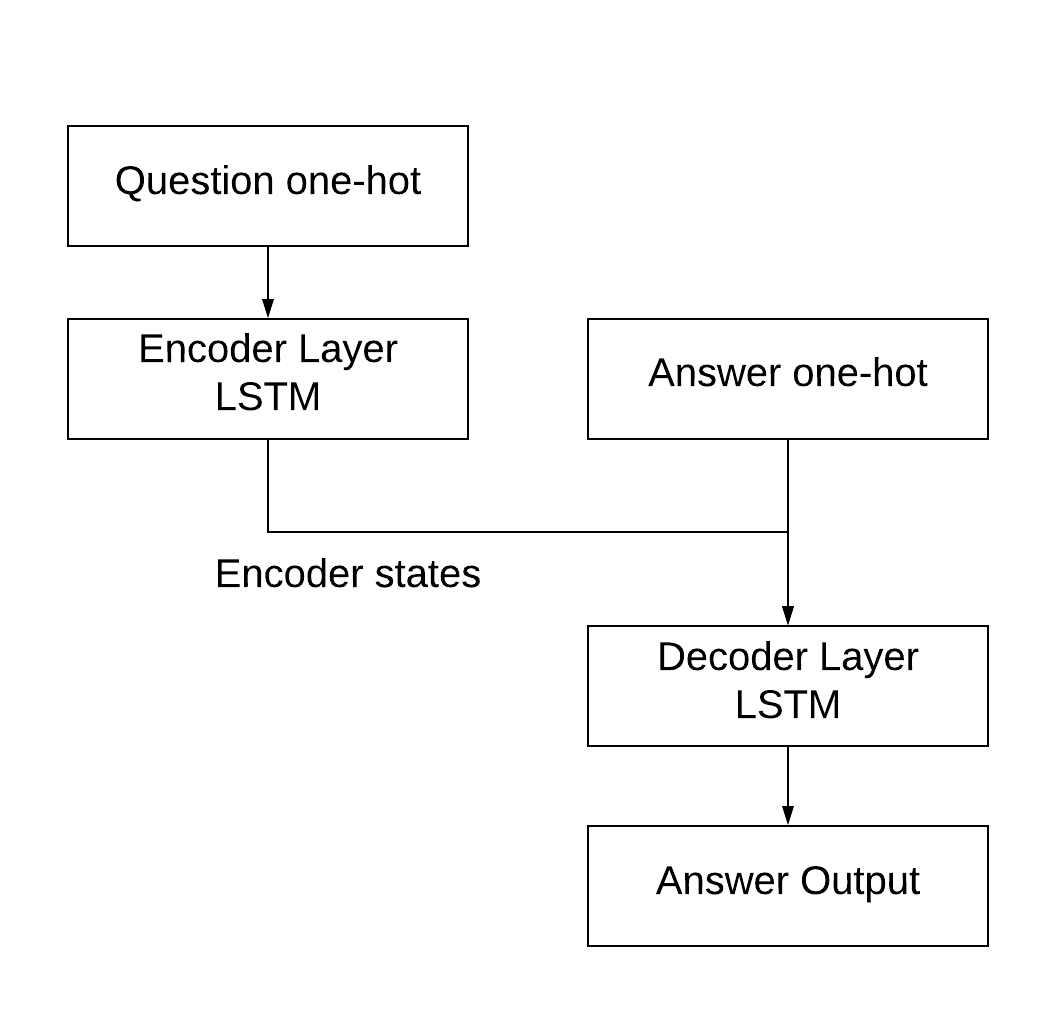
**Introduction**

We build this Chatbot by referencing the tutorial code [2]. We use word embedding rather than character level embedding to encode the sentences.

**Data Pre-processing**

1. Sort each sentence based on line ID in ascend order
2. Slice many stories based on the continuous line ID. For an example, L49, L50, L51, L60, L61, L62 will be classified to two groups [L49, L50, L51] and [L60, L61, L62]
3. Then, we assume that in a story, there are two characters having conversation. Therefore, we slice some story chunk to story. For example, [[L49, u0], [L50, u1], [L51, u0], [L52, u2], [L53, u3], [L54, u2]] will be divided into group1: [[L49, u0], [L50, u1], [L51, u0]], and group2: [[L52, u2], [L53, u3], [L54, u2]]
4. Extend some abbreviations to original form. For examples, I’m/He’s/I’ll/What’s/can’t to I am/He is/I will/What is/cannot.
5. Remove all the symbols "[-()\[#/@;:<>\*{}`+=~|.!?,]](mailto:#/@;:<>{}`+=~|.!?,])" in the text
6. lower the words
7. strip the sentence
8. Add <EOS>, <Start>, and <UNK> to vocab and modify the sentences by them
9. Convert the spaces which are more than one to one space between two words.
10. Select the questions and answers that the length is less than 15, and greater than 2
11. Select the words that the frequency is greater than 10

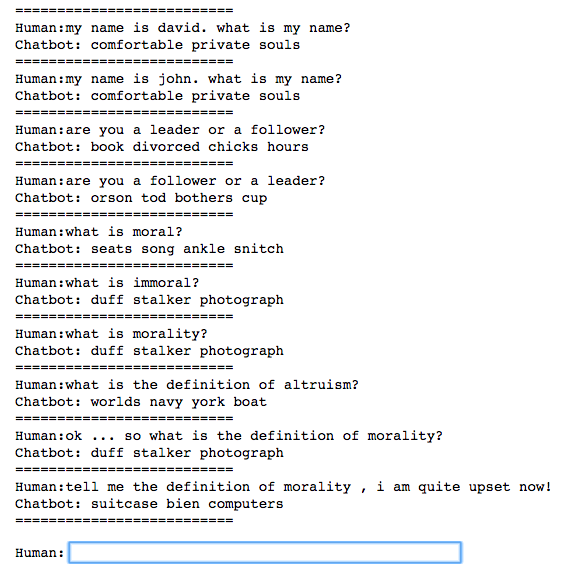
**Model**



We use LSTM as encoder and decoder layers. Our input for the encoder layer is one-hot for each word. The encoder layer encodes the input question one-hot sequence and sends its own internal states to the decoder layer as initial state of the decoder layer. Then the decoder layer is trained using the Answer one-hot input of the last state and the output answer data.

After the model is trained, we use the method to feed in the question data and generate predictions for each word in vocab. Then we sample from predictions using argmax as the next word.

**Sample Dialogue**



**Reference**

[**https://github.com/jkarimi91/chatbot**](https://github.com/jkarimi91/chatbot)

<https://blog.keras.io/a-ten-minute-introduction-to-sequence-to-sequence-learning-in-keras.html>

<https://github.com/keras-team/keras/blob/master/examples/addition_rnn.py>