**Introduction**

Include project objectives

**Data Description**

Include EDA findings

**Methodology**

*@ Include data preprocessing, model selection, and evaluation techniques*

Show evidence as to why we made this choice of model, i.e. links to articles etc justifying why this would be a good choice…………..

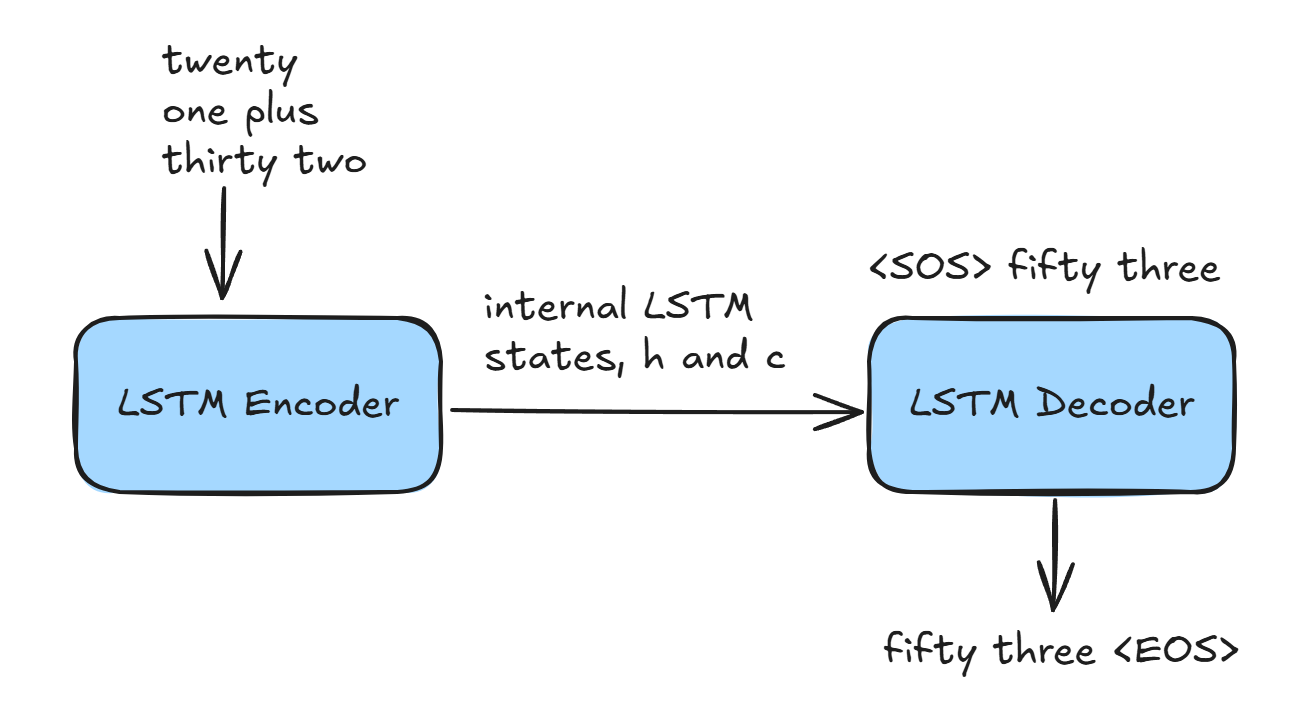
The approach taken leverages a ***Neural machine translation*** (NMT) approach that employs deep neural networks to perform machine translation from the source question to the target answer. This requires the neural translation mechanism to take in the question as text (that is, the source language) as a sequence of inputs and encodes these to a hidden representation, which is then decoded back to produce the translated text sequence as the target answer. One of the key advantages of this NMT system is that the whole machine translation system can be trained from end-to-end together.

Generally speaking, ***Recurrent Neural Networks*** (**RNNs**) architectures such as ***Long Short-Term Memory*** (**LSTMs**) and/or ***Gated Recurrent Units*** (**GRUs**) are used in the neural translation machine architecture.

Describe LSTM…..

Our prescribes approach adopts the encoder-decoder model approach,

The following diagram shows the high-level architecture of our proposed neural translation mechanism that uses one LSTM as the encoder to encode the input question sequence into final hidden states and final memory cell states .



The final hidden states and cell states [,] will capture the context of the whole input sequence, therefore [,] becomes a good candidate on which the decoder network can be conditioned. This hidden and cell state information, [,], is passed to the decoder LSTM model as the initial hidden and cell states. The decoder is then trained on the target sequence, with the input target sequence being one step behind the output target sequence. As per the decoder, the first word of the input sequence is the token word <SOS>, while the output label is the word *fifty*. The decoder network is just trained as a generative language model, where at any time step , the output label, is just the next word with respect to the input, that is, . The only new thing is that the final hidden and cell states of the encoder (that is, [,]) is fed to the initial hidden and cell states of the decoder to provide content for the translation.

**Results**

Discuss results and any conclusions and recommendations

**Conclusion**

Here is my conclusion

**References**

Ref ddddd