Indic Language NLP model Tamil-English

Done By :

John Alphonse Raja V

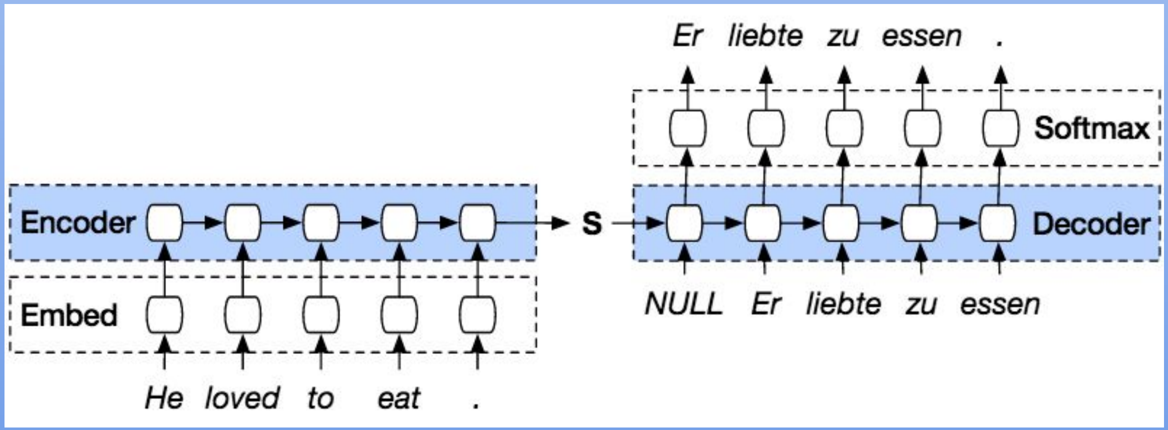
Deepak Pandian

Replicate Indic Language NLP model by AI4Bharat

- Most of the existing open source libraries use pre-built models or use sources that are either hard to reassemble. Also most of them focus on high resource languages.
- AI4bharat provides documentation and links to the source from where the models are generated.
- Replicating the process to build the model will help validating the approach and potentially helps understanding the end-to-end process of data collection, pre-processing, parameters involved.

What is Seq2Seq?

- **Seq2Seq** is a method of encoder-decoder based machine translation and language processing that maps an input of sequence to an output of sequence .
- The idea is to use 2 RNNs that will work together with a special token and try to predict the next state sequence from the previous sequence.
- Google used this model for machine translation in late 2016

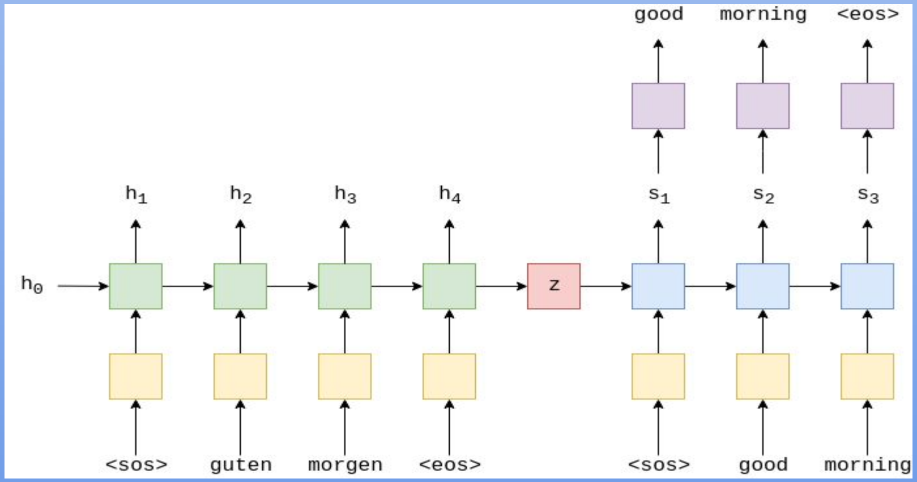


Text preprocessing

- **Tokenization:** It breaks the raw text into words, sentences called tokens. These tokens help in understanding the context or developing the model for the NLP.
- **Vocabulary:** The vocabulary is used to associate each unique token with an index (an integer). The vocabularies of the source and target languages are distinct.
- **Word embedding** is a term used for the representation of words for text analysis, typically in the form of a real-valued vector that encodes the meaning of the word such that the words that are closer in the vector space are expected to be similar in meaning.

Seq2seq Working:

- **Encoder**
- The encoder RNN iterates through the input sentence one token at a time, at each time step outputting an "output" vector and a "hidden state" vector.
- **Decoder**
- The decoder RNN generates the response sentence in a token-by-token fashion. It uses the encoder's context vectors to generate the next word in the sequence.



Network & Training followed by AI4Bharat

- We use fairseq (Ott et al., 2019) for training transformer-based models. We use 6 encoder and decoder layers, input embeddings of size 1536 with 16 attention heads and feedforward dimension of 4096.
- We optimized the cross entropy loss using the Adam optimizer with a label-smoothing of 0.1 and gradient clipping of 1.0.
- We use an initial learning rate of 5e-4, 4000 warmup steps and the learning rate annealing schedule as proposed in Vaswani et al.
- We use a global batch size of 64k tokens. We train each model on 8 V100 GPUs and use early stopping with the patience of 5 epochs.
- Trained for 30k epochs

Drawback and Future Plans:

- The output sequence relies heavily on the context defined by the hidden state in the final output of the encoder, making it challenging for the model to deal with long sentences.
- In the case of long sequences, there is a high probability that the initial context has been lost by the end of the sequence.
- To improve upon this model we'll use an attention mechanism, which lets the decoder learn to focus over a specific range of the input sequence.

References

- Ai4Bharat repository https://github.com/AI4Bharat/indicTrans
- Corpus to build the model https://indicnlp.ai4bharat.org/samanantar
- https://www.geeksforgeeks.org/seq2seq-model-in-machine-learning/
- https://www.youtube.com/watch?v=EoGUlvhRYpk
- https://github.com/bentrevett/pytorch-seq2seq/blob/master/1%20-%20Sequence%20to%20Sequence%20Learning%20with%20Neural%20Networks.ipynb
- https://arxiv.org/ftp/arxiv/papers/2104/2104.05596.pdf