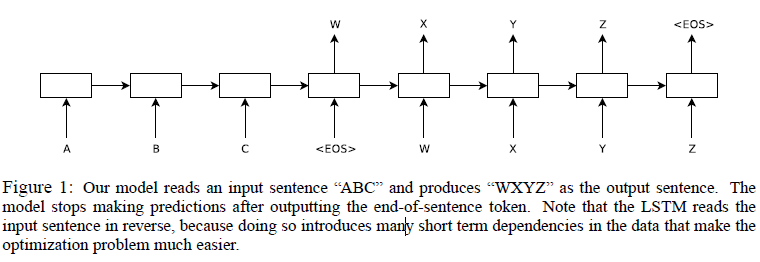
**Sequence to Sequence Learning with Neural Networks**

**Abstract**

Our method uses a multilayered Long Short Term Memory (LSTM) to map the input sequence to a vector of a fixed dimensionality, and then another deep LSTM to decode the target sequence from the vector.

**1 Introduction**

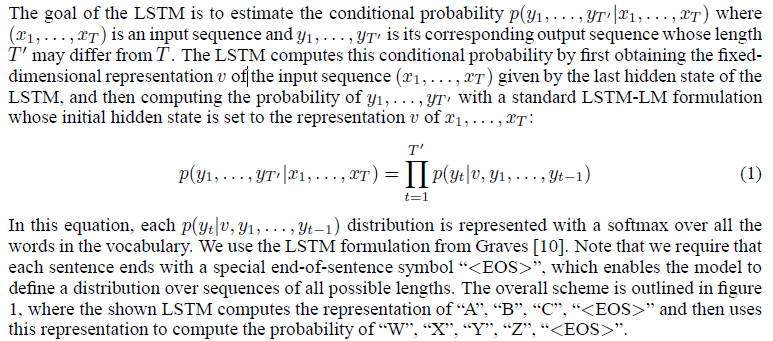
Sequences pose a challenge for DNNs because they require that the dimensionality of the inputs and outputs is known and fixed. The idea is to use one LSTM to read the input sequence, one timestep at a time, to obtain large fixed-dimensional vector representation, and then to use another LSTM to extract the output sequence from that vector (fig. 1). The second LSTM is essentially a recurrent neural network language model except that it is conditioned on the input sequence. The LSTM’s ability to successfully learn on data with long range temporal dependencies makes it a natural choice for this application due to the considerable time lag between the inputs and their corresponding outputs (fig. 1).

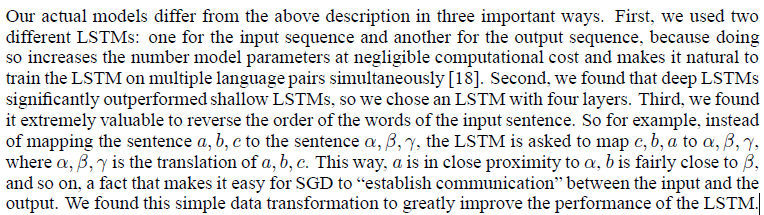


A useful property of the LSTM is that it learns to map an input sentence of variable length into a fixed-dimensional vector representation.

**2 The model**

The simplest strategy for general sequence learning is to map the input sequence to a fixed-sized vector using one RNN, and then to map the vector to the target sequence with another RNN.





**3 Experiments**

**3.1 Dataset details**

We used the WMT’1**3.4 Training details**4 English to French dataset. As typical neural language models rely on a vector representation for each word, we used a fixed vocabulary for both languages.

**3.2 Decoding and Rescoring**

**3.3 Reversing the Source Sentences**

While the LSTM is capable of solving problems with long term dependencies, we discovered that the LSTM learns much better when the source sentences are reversed (the target sentences are not reversed

While we do not have a complete explanation to this phenomenon, we believe that it is caused by the introduction of many short term dependencies to the dataset.

**3.4 Training details**

**3.5 Parallelization**

**3.6 Experimental Results**

We used the cased BLEU score to evaluate the quality of our translations.

**3.7 Performance on long sentences**

**3.8 Model Analysis**

One of the attractive features of our model is its ability to turn a sequence of words into a vector of fixed dimensionality.

**4 Related work**

**5 Conclusion**