

Introduction

Before 2017, the LSTM was the most optimal architecture for neural machine translation tasks. Subsequently, usually if a model is hailed as the 'best' for a certain task, as humans, we strive to come up with another model that will compete and even beat said 'best' model. Thus, the Transformer model was explored as an alternative within the past two years. The Transformer model is based on a self-attention mechanism. The Transformer architecture has been evaluated to out-perform the LSTM within these neural machine translation tasks. In this work, we propose that the Transformer out-performs the LSTM within our specific neural machine translation task: Semantic Parsing.

Materials and Methods

The Long-Short-Term-Memory or LSTM are units of an RNN. An LSTM unit has a cell, an input gate, an output gate and a forget gate. These three gates regulate the flow of information into and out of the cell, which is distributed over various time intervals. The Transformer is a model architecture that gets rid of recurrence and relies entirely on an attention mechanism to execute dependencies between input and output. Thus, the transformer allows for significantly more parallelization and can reach a new state of the art in translation quality. The decoder, like the encoder, is composed of a stack of $N = 6$ identical layers and also has three additional sub-layers. The decoder has a additional third sub-layer, unlike the encoder, because this layer performs multi-head attention over the output of the encoder stack. The decoder's sub-layers, like the encoder, apply residual connects around each of the sub-layers and then do layer normalization. Also, we mask the sub-layers in the decoder stack, so that we can prevent positions from attending to subsequent positions, which allows us to be sure that the predictions for position x can only depend on the known outputs at positions less than x .

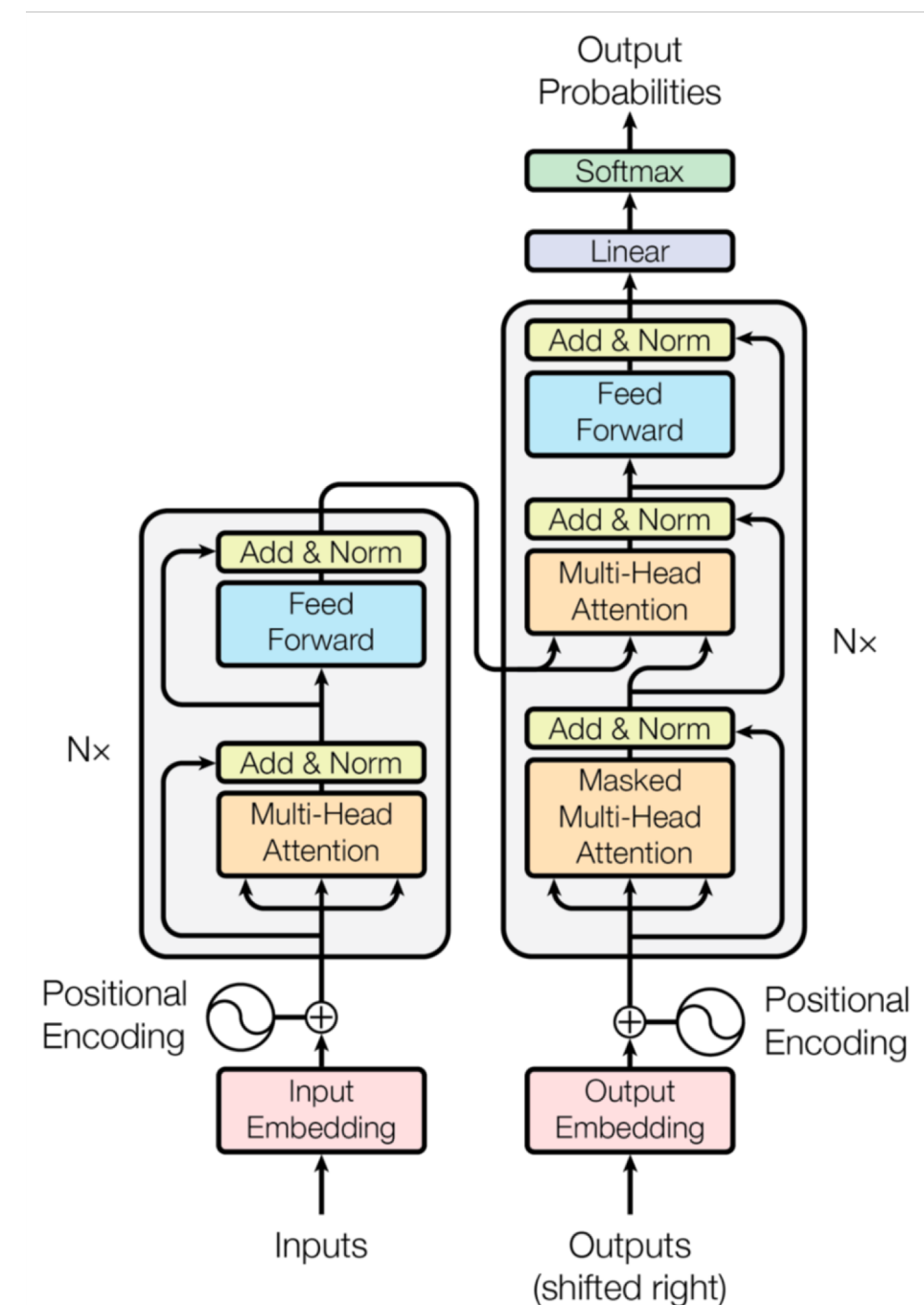


Figure 1: The Transformer - model architecture.

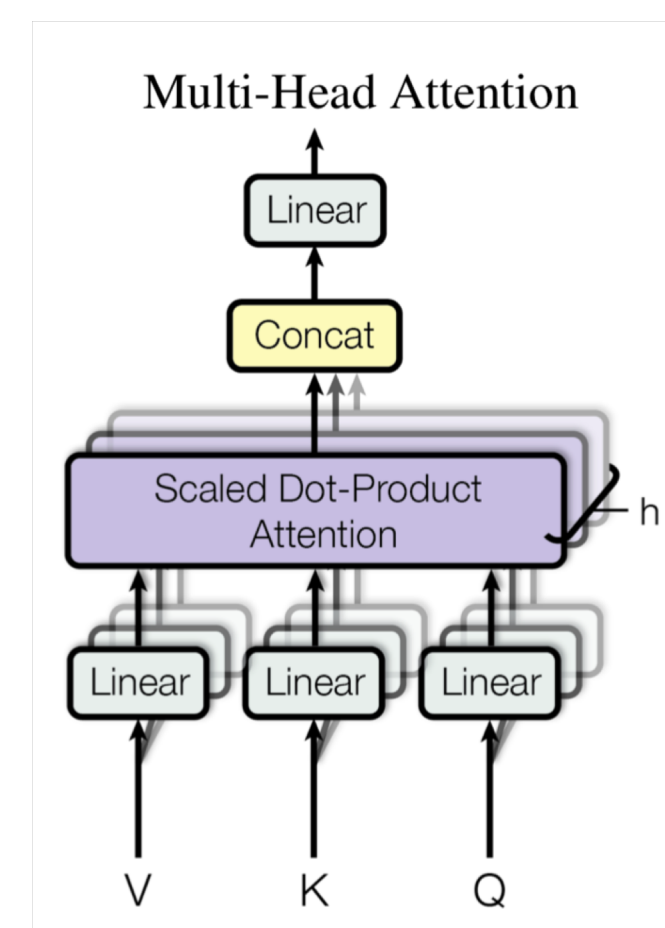


Figure 2. Multi-Head Attention

| Method | GEO | JOBS | ATIS | SPIDER |
|---------|-------|-------|-------|--------|
| Seq2Seq | 81.8% | 85.7% | 80.3% | 1.74% |
| DongL16 | 84.6% | 87.1% | 84.2% | N/A |
| JiaL16 | N/A | 84.6% | 76.3% | N/A |

Figure 3: Results for LSTM compared to STOA results for seq2seq

| Method | ATIS | SPIDER | JOBS | GEO |
|-------------|-------|--------|-------|-------|
| Transformer | 55.8% | 2.61% | 5.97% | 4.2% |
| JiaL16 | 76.3% | N/A | 84.6% | N/A |
| DongL16 | 84.2% | N/A | 87.1% | 84.6% |

Figure 4: Results for Transformer compared to STOA results for LSTM seq2seq

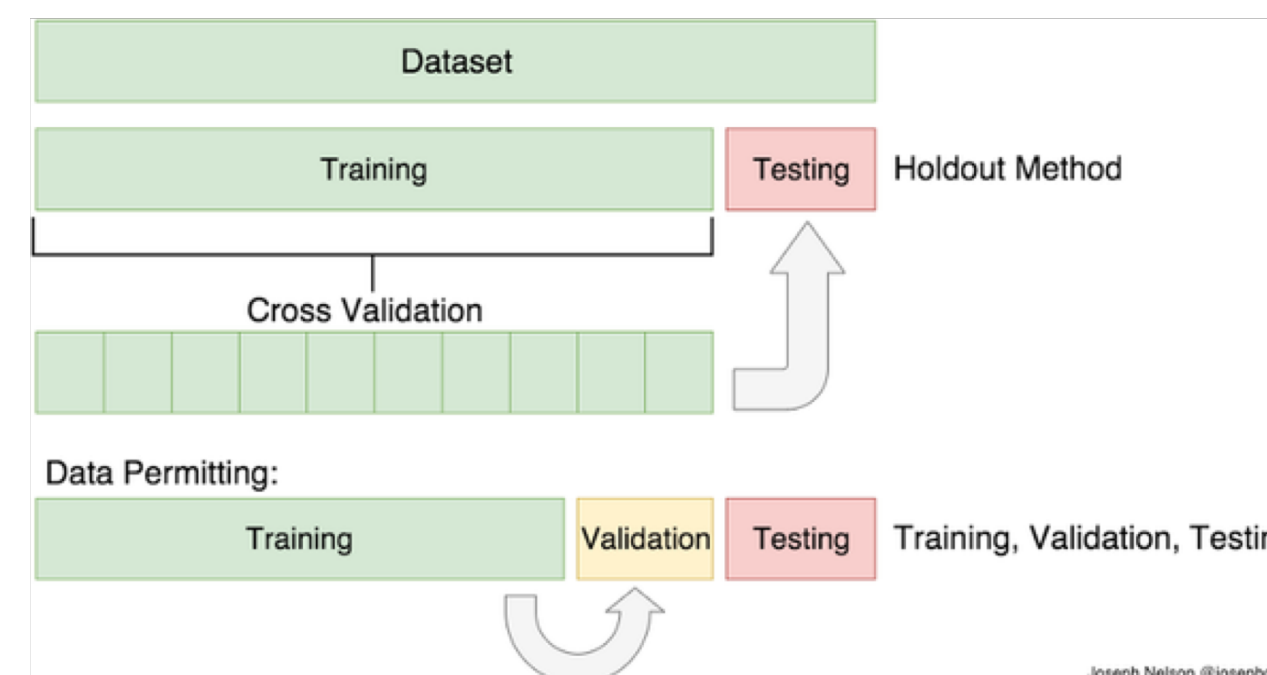


Figure 5: Cross Validation Model

Results

The results that we got for the LSTM are around the same as scores achieved through DongL16 paper with the Sequence-to-Sequence with attention model on Semantic Parsing tasks using the same data sets. I believe we were able to achieve state of the art results without much tuning because of the fast-ai network allowing us to utilize the Adam optimizer and achieve super-convergence on the learning rate of the model. Nonetheless, for the Transformer, we did not achieve the accuracy we wanted. The point of this experiment was to show that the Transformer would out-perform our LSTM model scores like the Transformer model did in the "Attention is All You Need" paper; however, our model did not out-perform the LSTM in any of the data-sets, except Spider. However, I believe this is because the data-sets we used were too small. The lowest differential between accuracy tests when comparing the Transformer and the LSTM on different data-sets was on the ATIS data set because that data set was the biggest out of the three.

Conclusion

I believe that this experiment should be continued because there is substantial potential in this sphere. If we were able to have access to a larger train set then I think that we would've gotten better results than the LSTM. Nonetheless, even though we didn't get the results we wanted, we still had a lot of fun utilizing fast-ai's framework and the fair-sequence-to-sequence models developed by Facebook, since it made our lives a lot easier. Furthermore, we might have achieved more significant results if we were able to gain a large corpus of Text-to-SQL data that doesn't use the type of structured learning that Spider encourages.

Acknowledgement

Thank you to Dragomir Radev and the entire LILY Lab for their help.