Paper 7 | Sequence to Sequence Learning with Neural Networks by Ilya et. al – published in September 2014

Abstract

1. The paper presents a general end-to-end approach to sequence learning that makes minimal assumptions on the sequence structure
2. The paper uses a multi-layered 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
3. The main result shows that on an English to French translation task, the LSTM achieve a BLEU score of 34.8 on the entire test set, where the LSTM's BLEU score was penalised on out-of-vocabulary words. The LSTM did not have difficulty on long sentences
4. The LSTM also learned sensible phrase and sentence representations that are sensitive to word order
5. The paper also found that reversing the order of the words in all source sentences improved the LSTM's performance because it introduced many short-term dependencies between the source and the target sentence, making the optimisation problem easier

Introduction

1. Deep neural networks (DNNs) are powerful because they can perform arbitrary parallel computation and large DNNs can be trained with supervised backpropagation whenever the labelled training set has enough information to specify the network's parameters
2. DNNs can only be applied to problems whose inputs and targets can be sensibly **encoded with vectors of fixed dimensionality** – a significant limitation. **Therefore, it is important to have a domain-independent method that learns to map sequences to sequences**
3. The main idea is to use one LSTM to read the input sequence, to obtain a large fixed-dimensional vector representation and then use another LSTM to extract the output sequence from that vector. The second LSTM is essentially a conditional recurrent neural network language model

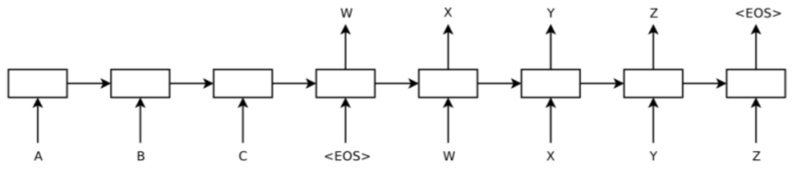


Figure 1: Our model reads an input sentence "ABC" and produces "WXYZ" as the output sentence. The model stops making predictions after outputting the end-of-sentence token. Note that the LSTM reads the input sentence in reverse, because doing so introduces many short term dependencies in the data that make the optimization problem much easier.

4. **The model was able to do well on long sentences because we reversed the order of words in the source sentence but not the target sentences in the training and test set**
5. The translation objective encourages LSTM to find sentence representations that capture their meaning and the result shows that the model is aware of word order and is fairly invariant to the active and passive voice

Models

1. RNN is a generalisation of feedforward neural networks to sequences. Given a sequence of inputs, RNN computes a sequence of outputs by iterating the following equation:

   $$\begin{aligned} h_t &= \operatorname{sigm}\left(W^{\mathrm{hx}} x_t + W^{\mathrm{hh}} h_{t-1}\right) \\ y_t &= W^{\mathrm{yh}} h_t \end{aligned}$$

2. It is not clear on how to apply an RNN to problems whose input and output sequences have different lengths. A simple 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. This would work but it would be difficult to train the RNNs due to long term dependencies, hence why we decided to use LSTM
3. Goal of LSTM is to estimate the conditional probability as below:

   $$p(y_1, \ldots, y_{T'} | x_1, \ldots, x_T) = \prod_{t=1}^{T'} p(y_t | v, y_1, \ldots, y_{t-1})$$

   Where x_1,….,x_T is the input sequence and y_1,….,$y_{T'}$ is its corresponding output sequence where length T' may differ from T. v is the fixed-dimensional representation of the input sequence. Each conditional probability distribution is represented with a softmax over all the words in the vocabulary
4. **The proposed model**
   - Used two different LSTMs, one for input and one for output
   - LSTM with four layers (as results shows that a deep LSTMS significantly outperformed shallow one)
   - Reserve the order of the words of the input sentence. This way, a is in close proximity to alpha, b is fairly close to beta and so on. This makes it easy for stochastic gradient descent to establish communication between the input and the output

Experiments and Results

1. Training objective and decoding
   - The objective is to maximise the log probability of a correct translation T, given the source sentence S

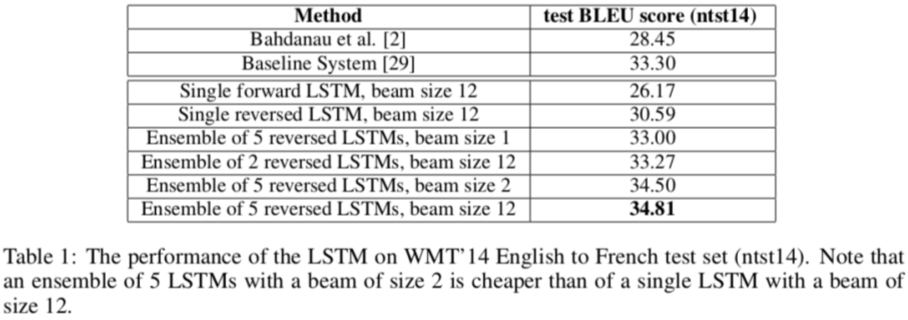
     $$1/|\mathcal{S}| \sum_{(T,S) \in \mathcal{S}} \log p(T|S)$$

     Where S is the training set. Once the training is complete, we produce translations by finding the most likely translation according to the LSTM:

     $$\hat{T} = \arg\max_{T} p(T|S)$$

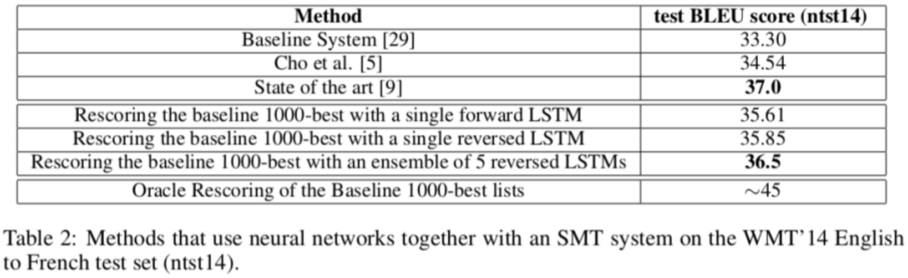
   - We use a simple left-to-right beam search decoder to identify the most likely translation, which maintains a small number B of partial hypotheses, where a partial hypothesis is a prefix of some translation
2. Reversing source sentences
   - By reversing the source sentences, the test BLEU score increased from 25.9 to 30.6
   - This improvement is possibly due to the introduction of many short-term dependencies to the dataset, whereby we reduced the "minimal time lag"

3. Experimental results



Method	test BLEU score (ntst14)
Bahdanau et al. [2]	28.45
Baseline System [29]	33.30
Single forward LSTM, beam size 12	26.17
Single reversed LSTM, beam size 12	30.59
Ensemble of 5 reversed LSTMs, beam size 1	33.00
Ensemble of 2 reversed LSTMs, beam size 12	33.27
Ensemble of 5 reversed LSTMs, beam size 2	34.50
Ensemble of 5 reversed LSTMs, beam size 12	**34.81**

Table 1: The performance of the LSTM on WMT'14 English to French test set (ntst14). Note that an ensemble of 5 LSTMs with a beam of size 2 is cheaper than of a single LSTM with a beam of size 12.



Method	test BLEU score (ntst14)
Baseline System [29]	33.30
Cho et al. [5]	34.54
State of the art [9]	**37.0**
Rescoring the baseline 1000-best with a single forward LSTM	35.61
Rescoring the baseline 1000-best with a single reversed LSTM	35.85
Rescoring the baseline 1000-best with an ensemble of 5 reversed LSTMs	**36.5**
Oracle Rescoring of the Baseline 1000-best lists	~45

Table 2: Methods that use neural networks together with an SMT system on the WMT'14 English to French test set (ntst14).

Conclusion

1. **Showed that a large deep LSTM with a limited vocabulary can outperform a standard SMT-based system whose vocabulary is unlimited. This suggests that it should also do well in many other sequence learning problems, provided there are enough training data**
2. Concluded that it is important to find a problem encoding that has the greatest number of short term dependencies
3. Surprised by LSTM ability to correctly translate very long sentences