Investigating Relational Recurrent Neural Networks with Variable Length Memory Pointer

Mahtab Ahmed and Robert E. Mercer
Department of Computer Science
University of Western Ontario
London, ON, Canada

Introduction

- Memory based Neural Networks can remember information longer while modelling temporal data.

- Encode a Relational Memory Core (RMC) as the cell state inside an LSTM cell.
  - Uses standard Multi-head Self Attention.
  - Uses variable length memory pointer.

- Evaluate on four different tasks.
  - State of the art on one of them; On par with the other three.

Standard LSTM

$$\boldsymbol{i}^t = \sigma(\mathbf{W}^{(i)}x^t + \mathbf{U}^{(i)}h^{t-1} + \mathbf{b}^{(i)})$$

$$\boldsymbol{f}^t = \sigma(\mathbf{W}^{(f)}x^t + \mathbf{U}^{(f)}h^{t-1} + \mathbf{b}^{(f)})$$

$$\tilde{\boldsymbol{c}}^t = \tanh(\mathbf{W}^{(c)}x^t + \mathbf{U}^{(c)}h^{t-1} + \mathbf{b}^{(c)})$$

$$\boldsymbol{c}^t = i^t \cdot \tilde{c}^t + f_t \cdot c^{t-1}$$

$$\boldsymbol{h}^t = \tanh(c^t)$$

The model: Fixed Length Memory Pointer

$$\widetilde{M}^t_{2\times b\times d} = [M^{t-1}_{1\times b\times d}; \mathbf{W}x^t_{1\times b\times d}]$$

Random Input at *t*

- Apply Multi-head Self Attention and create a weighted version, M

$$Q = \widetilde{M}\mathbf{W}^q \qquad K = \widetilde{M}\mathbf{W}^k \qquad V = \widetilde{M}\mathbf{W}^v$$

$$A = softmax(\frac{QK^T}{\sqrt{d_k}}) \qquad M = AV$$

- Add a residual connection

$$M = \widetilde{M} + M$$

- Apply Layer-Normalization block on top of M
  - Maintain separate version of mean and variance projection matrices.

The model: Fixed Length Memory Pointer (contd.)

- *n* non-linear projections of h^t are applied followed by a residual connection

$$X = \mathtt{f}(\mathbf{W}^{(1)}\mathtt{f}(h^t\mathbf{W}^{(2)})) + h^t$$

f = RELU and $h^t = M$

- Resultant tensor X (having shape 2 × b × d) is split on the cardinal dimension to extract the memory

$$M^t_{1\times b\times d} = X^1_{1\times b\times d}$$

- LSTM's candidate cell state gets changed to

$$\mathbf{c}^t = i^t \cdot M^t + f^t \cdot M^{t-1}$$

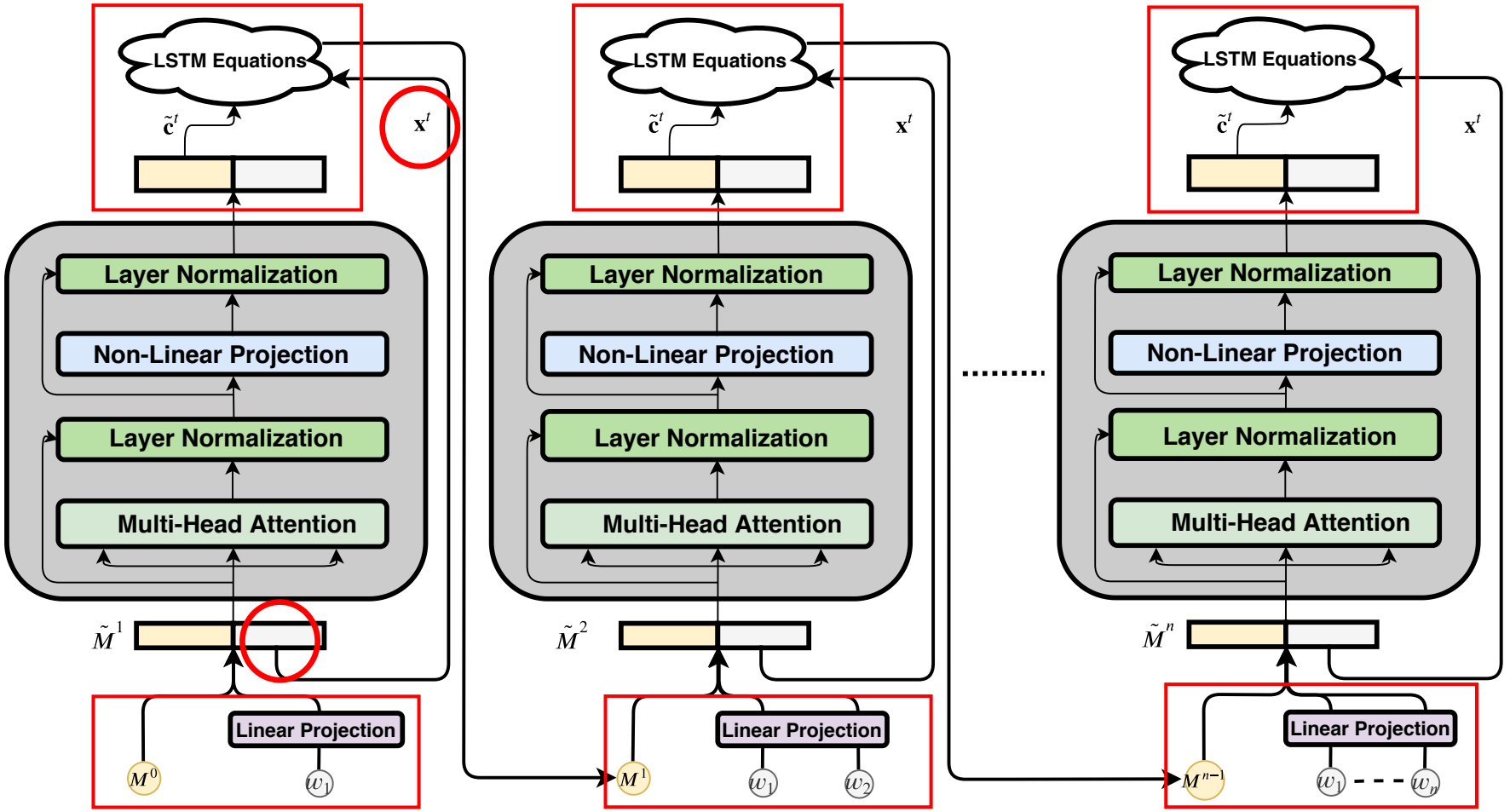
- x^t is replaced with the projected input (= Wx^t) in all LSTM equations.

Variable Length Memory Pointer

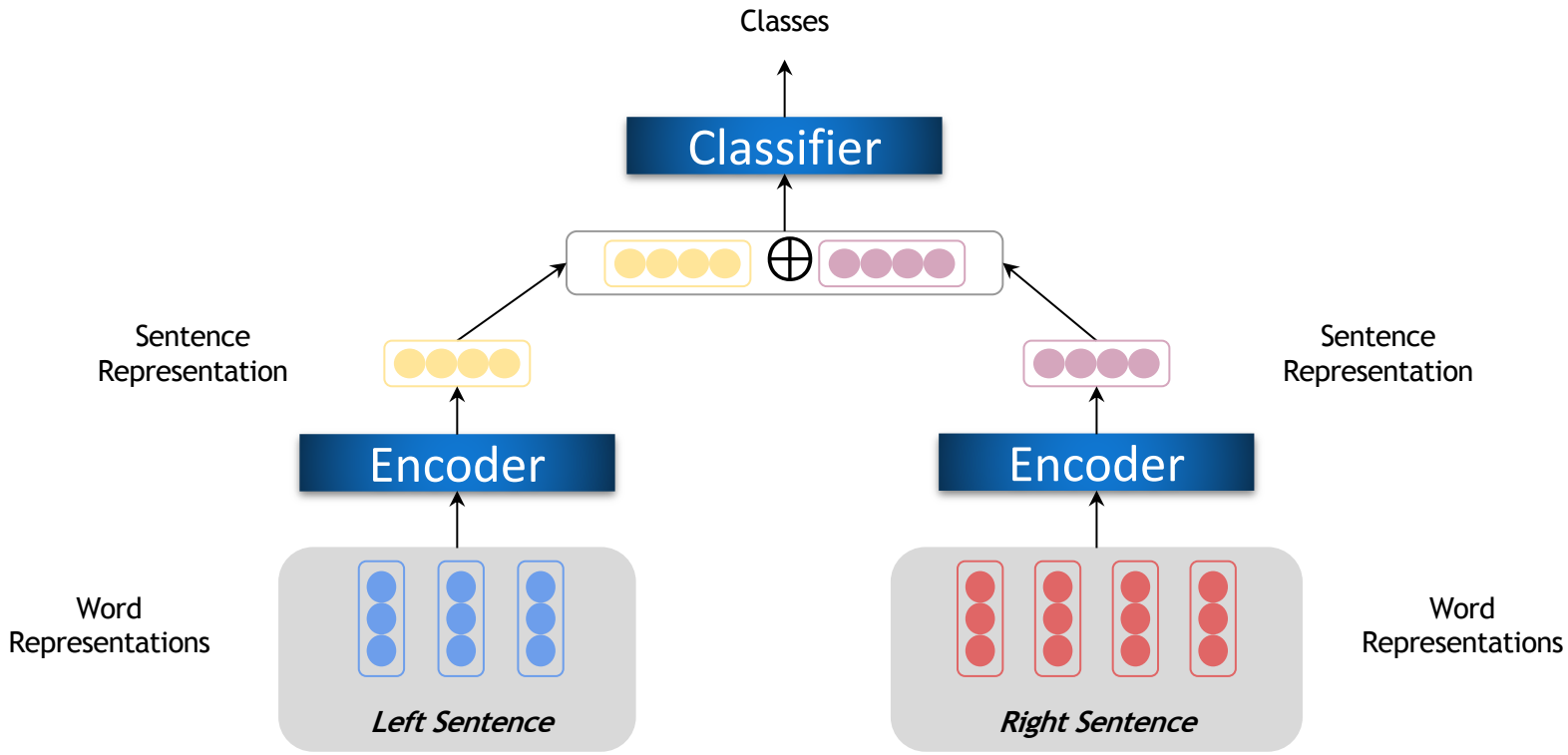
$$\widetilde{M}^{t}_{(t+1)\times b\times d} = [M^{t-1}_{1\times b\times d}; \mathbf{W}x^{t}_{1\times b\times d}; \mathbf{W}x^{t-1}_{1\times b\times d}; \cdots; \mathbf{W}x^{1}_{1\times b\times d}]$$

- Share ***W*** across all time steps.
- Apply all the steps as before.
- For Layer-Normalization, maintain just one version of mean and variance projection matrices.
- Memory is still at the cardinal dimension.
- Rather than looking at everything before
  - Track a fixed window of words (n-grams).
  - Mimic the behavior of convolution kernel.

Model Architecture



Sentence Pair Modelling



InferSent - https://arxiv.org/abs/1705.02364

Hyperparameters

Config	Value	Config	Value
Initial learning rate	**0.1** / 0.05 / 0.001	maxNorm	**5**
Batch size	10 / **16** / 25	Learning rate decay	**0.99**
No. of Attention layers	**1**/ 2 / 3	Dropout FC	**0.0375 - 0.5**
Hidden dimension	256, 512, **1024**	No. of Heads	**8**
Word embedding	Glove 300D	W^q, W^k, W^v dimension	**128**

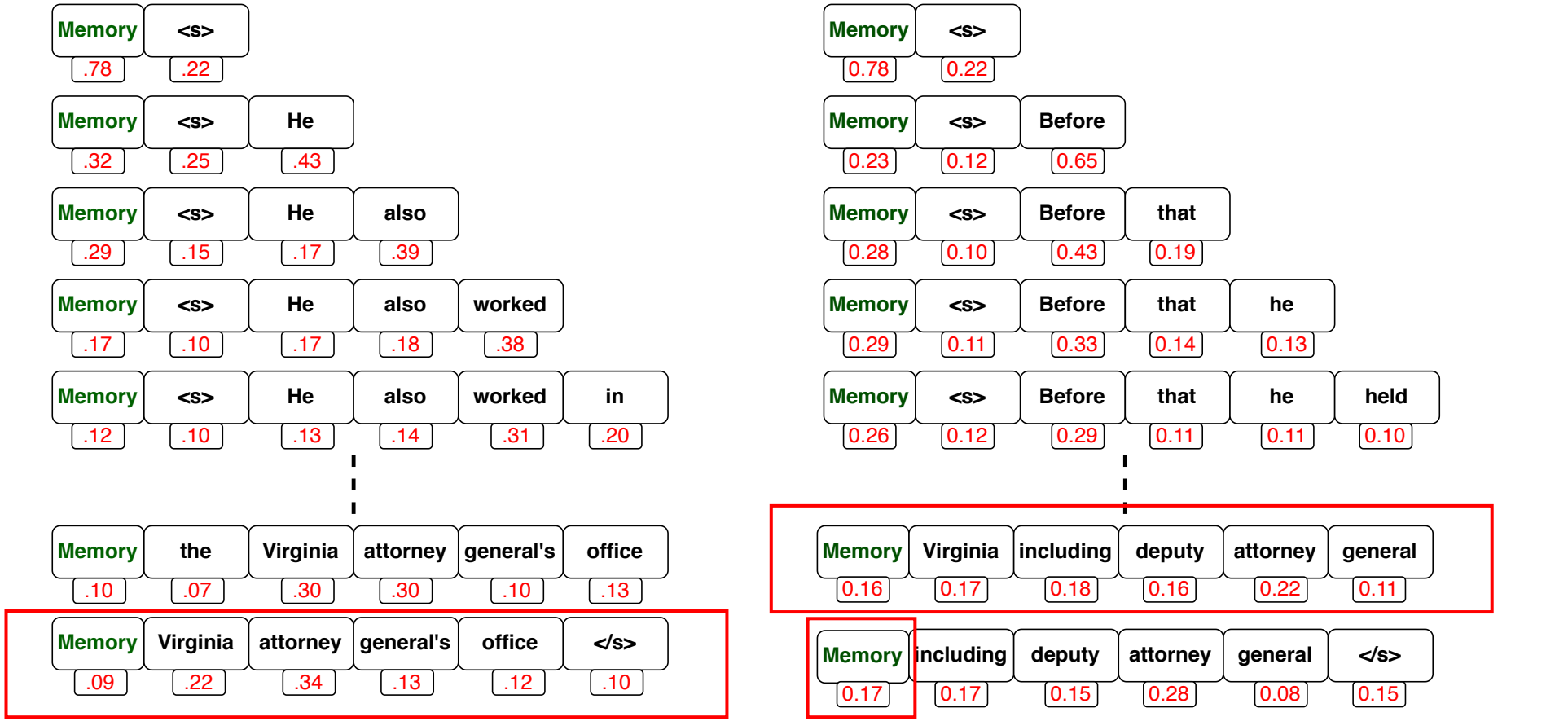
- We tried a range of values for each hyperparameter. The ones that worked for us are bold-faced.

Experimental Results

Model	MSRP	AI2-8grade	SICK-E	SICK-R
	Acc.	Acc.	Acc.	r/MSE
LSTM_{RMC} + FLMP	74.67	74.72	85.38	0.8107/0.3452
LSTM_{RMC} + VLMP (window size = 5)	75.89	74.72	84.28	0.8440/0.2925
InferSent [4] †	74.46	74.10	84.62	0.8563/0.2732
LSTM [4] †	70.74	74.24	76.80	0.8291/0.3244
BiLSTM Projection Layer [4] †	74.24	75.15	85.20	0.8037/0.3667
Inner Attention [9] †	69.74	74.32	72.01	0.7863/0.3944
ConvNet Encoder [19] †	73.96	75.15	83.82	0.8520/0.2806
Transformer Encoder [3]	74.96	-	81.15	-/0.5241
Seq-LSTMs [20]	71.70	63.30	-	0.8528/0.2831
Tree LSTM [20]	73.50	69.10	-	0.8664/0.2610
Tree LSTM + Attn. [20]	75.80	72.50	-	0.8730/0.2426
Tree GRU [20]	73.96	70.60	-	0.8672/0.2573
Tree GRU + Attn. [20]	74.80	72.10	-	0.8701/0.2524
RAE [16]	76.80	-	-	-

- Models marked with † are the ones that we implemented

Attention Visualization



Conclusion

- Extend the classical RMC with variable length memory pointer.
  - Uses a non-local context to compute an enhanced memory.

- Design a sentence pair modelling architecture.
  - Evaluate on four different tasks.
  - On par performance on most of the tasks and best performance on one of them.

- Interprets the attention shifting very well.

- Memory pointer length does not follow a uniform pattern across all datasets.

Thank you