Jan.26, 2017

Tensor Product Representation for Machine Comprehension: version 0.0

This note is a summary of what we have so far which will be the basis for implementation of primary baseline.

We have started with bi-directional attention flow model [BidirAtt16] proposed by AI2. This is the model that is one of the top ranked models in Stanford's SQUAD leader board [Squad16, Leaderboard17]. The model proposed by [Squad16] is shown in Fig. 1.1.

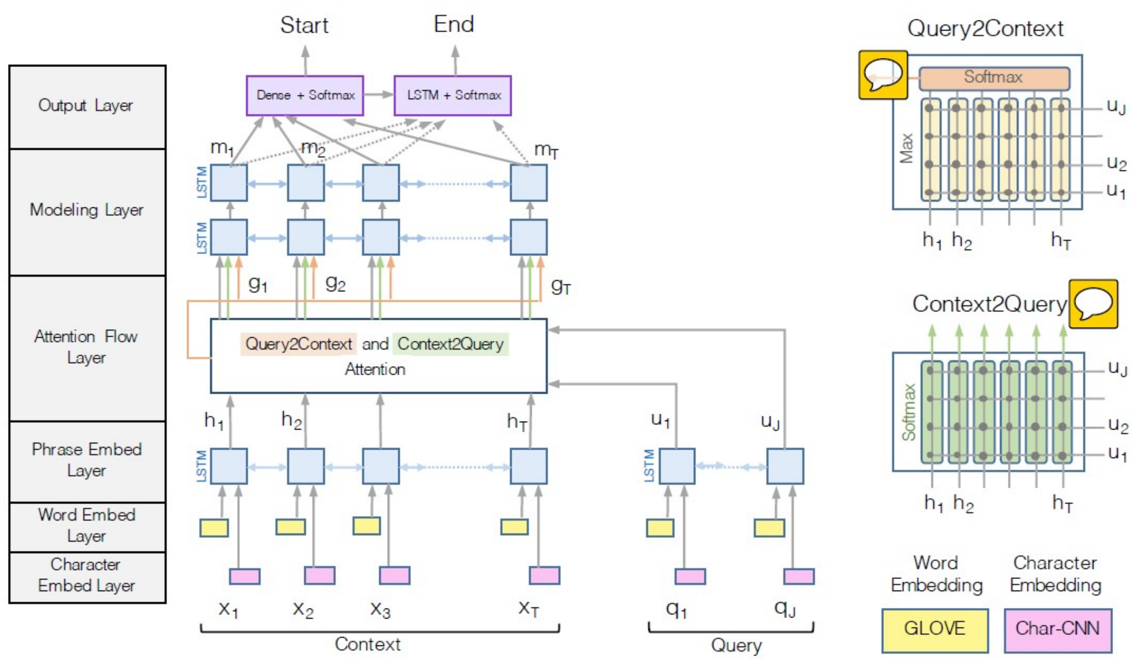

Figure 1.1: Model proposed by [Squad16] for machine comprehension. Figure from [Squad16].

The idea, proposed by Paul, is to have a TPR model along with the current model shown in Fig. 1.1. As a version 0.0, this TPR model is added before the Attention Flow layer. The idea is shown in Fig. 1.2 (from Paul's whiteboard).

The inputs to the TPR model at time "t" are the following:

1. $\mathbf{x}_t$: The output from Word Embed Layer for t-th word in Fig. 1.1.
2. $\mathbf{h}_{t-1}$: The output from Phrase Embed Layer for $t-1$-th word in Fig. 1.1.
3. $\mathbf{T}_{t-1}$: The TPR representation from previous word in the sequence, i.e., $t-1$-th word.

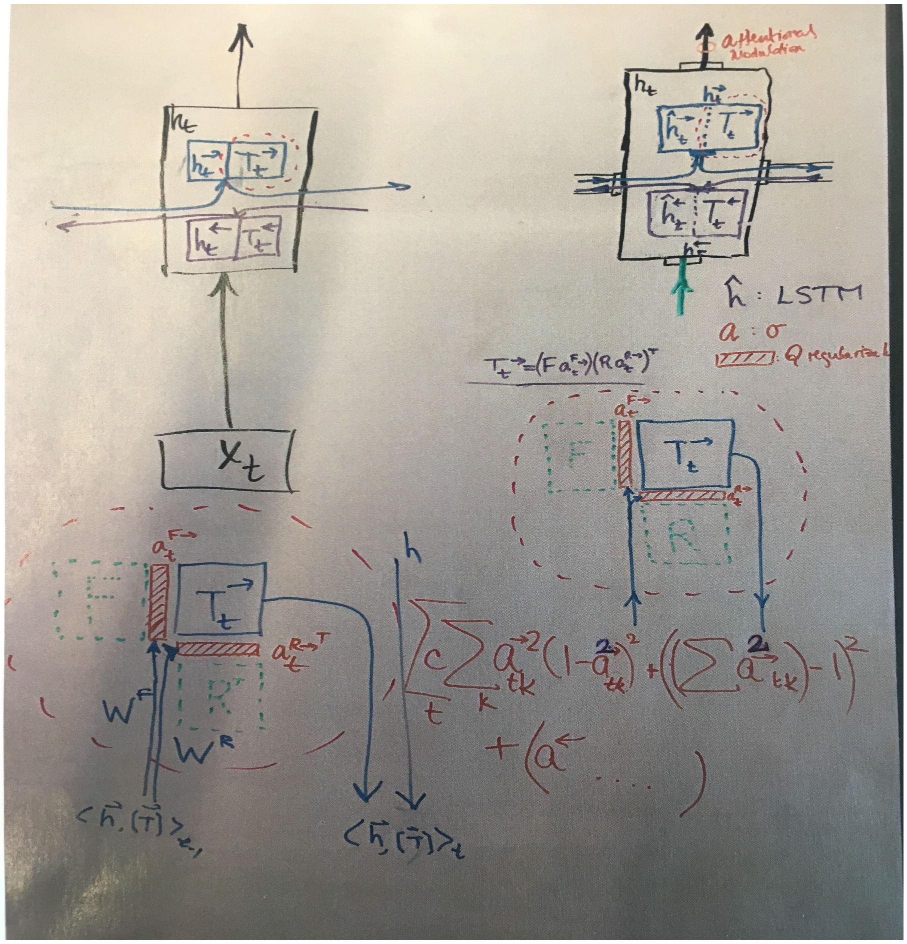

Figure 1.2: TPR model proposed by Paul. Figure from Paul's white board.

The inputs to the LSTM in phrase embed layer at time "t" are as before but there is also an extra input from $\mathbf{T}_{t-1}$.

A TPR forward pass will be as follows:

$$\mathbf{a}_t^{\mathbf{F}} = f(\mathbf{W}^{\mathbf{F}}\mathbf{x}_t + \mathbf{W}_{\mathbf{rec1}}^{\mathbf{F}}\mathbf{h}_{t-1} + \mathbf{W}_{\mathbf{rec2}}^{\mathbf{F}} vec(\mathbf{T}_{t-1})) \tag{1.1}$$

$$\mathbf{a}_t^{\mathbf{R}} = f(\mathbf{W}^{\mathbf{R}}\mathbf{x}_t + \mathbf{W}_{\mathbf{rec1}}^{\mathbf{R}}\mathbf{h}_{t-1} + \mathbf{W}_{\mathbf{rec2}}^{\mathbf{R}} vec(\mathbf{T}_{t-1})) \tag{1.2}$$

$$\mathbf{T}_t = \mathbf{F}\underbrace{\mathbf{a}_t^{\mathbf{F}}(\mathbf{a}_t^{\mathbf{R}})^T}_{\mathbf{B}_t}\mathbf{R}^T \tag{1.3}$$

where $\mathbf{F}$ and $\mathbf{R}$ refer to filler and roles matrices, $\mathbf{B}_t$ refers to the binding matrix and $vec(.)$ vectorizes the given matrix. The parameters that should be learned during training in above equations are $\{\mathbf{W}^{\mathbf{F}}, \mathbf{W}^{\mathbf{R}}, \mathbf{W}_{\mathbf{rec1}}^{\mathbf{F}}, \mathbf{W}_{\mathbf{rec1}}^{\mathbf{R}}, \mathbf{W}_{\mathbf{rec2}}^{\mathbf{F}}, \mathbf{W}_{\mathbf{rec2}}^{\mathbf{R}}, \mathbf{F}, \mathbf{R}\}$.

In a bidirectional version, the same set of equations with different parameters are used for left-to-right and right-to-left models.

To make the final binding matrix $\mathbf{B}_t$ as sparse as possible (ideally constructed by two one-hot vectors $\mathbf{a}_t^{\mathbf{F}}$ and

$\mathbf{a}_t^{\mathbf{R}}$), the following regularization term or a variant of it will be added to the cost function, i.e., to equation (5) of [BidirAtt16]:

$$C^F \sum_t [\sum_k (\mathbf{a}_{tk}^{\mathbf{F}})^2 (1 - \mathbf{a}_{tk}^{\mathbf{F}})^2 + ([\sum_k (\mathbf{a}_{tk}^{\mathbf{F}})^2] - 1)^2] +$$
$$C^R \sum_t [\sum_k (\mathbf{a}_{tk}^{\mathbf{R}})^2 (1 - \mathbf{a}_{tk}^{\mathbf{R}})^2 + ([\sum_k (\mathbf{a}_{tk}^{\mathbf{R}})^2] - 1)^2] \tag{1.4}$$

where C^F and C^R are used to adjust the effect of the regularization terms. This is to force the model to ideally assign one symbol to one role at each time step. In a bidirectional version, another two terms from right-to-left direction will also be added to above equations.

References

[BidirAtt16] M. Seo, et al, "Bidirectional Attention Flow for Machine Comprehension", *https://arxiv.org/abs/1611.01603*, November 2016.

[Squad16] P. Rajpurkar, et al, "SQuAD: 100,000+ Questions for Machine Comprehension of Text", *https://arxiv.org/abs/1606.05250*, November 2016.

[Leaderboard17] *https://rajpurkar.github.io/SQuAD-explorer/*, accessed on Jan. 26, 2017.