BI-DIRECTIONAL ATTENTION FLOW FOR MACHINE COMPREHENSION

Published as a conference paper at ICLR 2017

Fengyue

2017.09.14

Task

- Machine comprehension (MC), answering a query about a given context paragraph.
- The answer is derived by predicting the start and the end indices of the phrase in the paragraph.

Motivation

- Typical methods use attention to focus on a small portion of the context and summarize it with a fixed-size vector, or often form a uni-directional attention.
- This paper introduce a multi-stage hierarchical process that represents the context at different levels of granularity and uses bi-directional attention flow mechanism to obtain a query-aware context representation without early summarization.

Model

1. **Character Embedding Layer** maps each word to a vector space using character-level CNNs.
2. **Word Embedding Layer** maps each word to a vector space using a pre-trained word embedding model.
3. **Contextual Embedding Layer** utilizes contextual cues from surrounding words to refine the embedding of the words. These first three layers are applied to both the query and context.
4. **Attention Flow Layer** couples the query and context vectors and produces a set of query-aware feature vectors for each word in the context.
5. **Modeling Layer** employs a Recurrent Neural Network to scan the context.
6. **Output Layer** provides an answer to the query.

Model

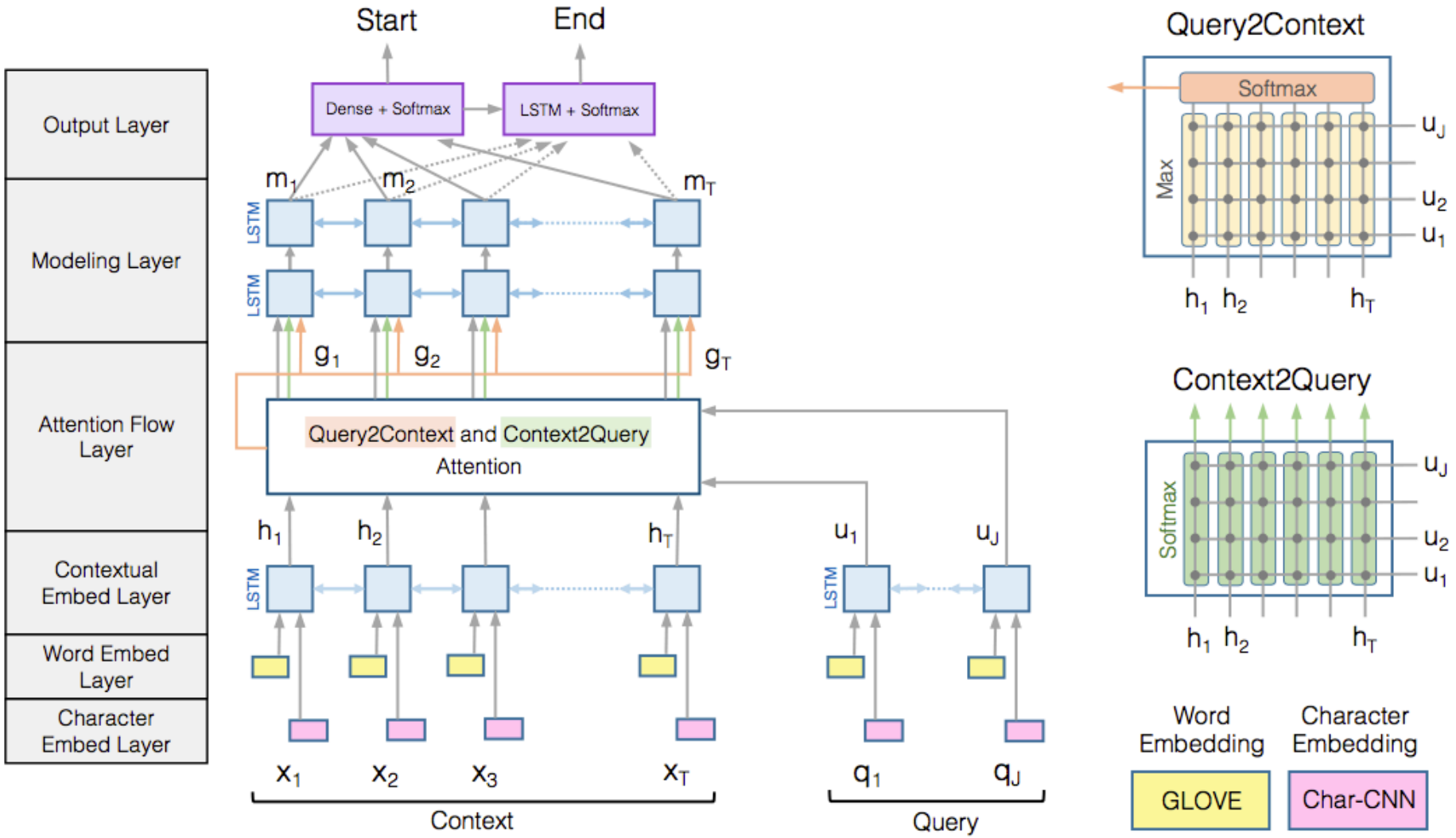
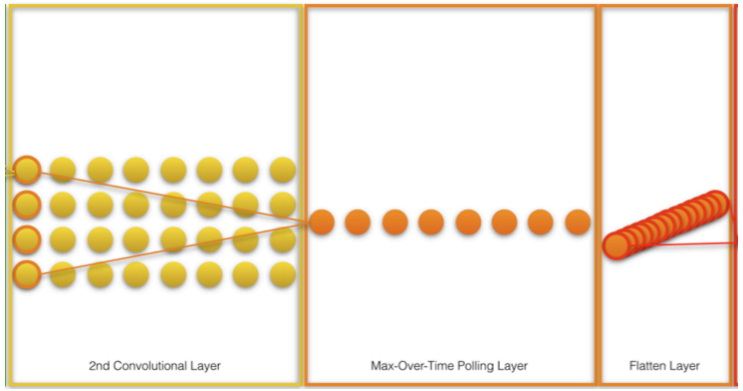


Figure 1: BiDirectional Attention Flow Model *(best viewed in color)*

Model

1. Character Embedding Layer

- Obtaining the character level embedding of each word using Convolutional Neural Networks (CNN).
- Characters are embedded into vectors, which can be considered as 1D inputs to the CNN, and whose size is the input channel size of the CNN. The outputs of the CNN are max-pooled over the entire width to obtain a fixed-size vector for each word.



Model

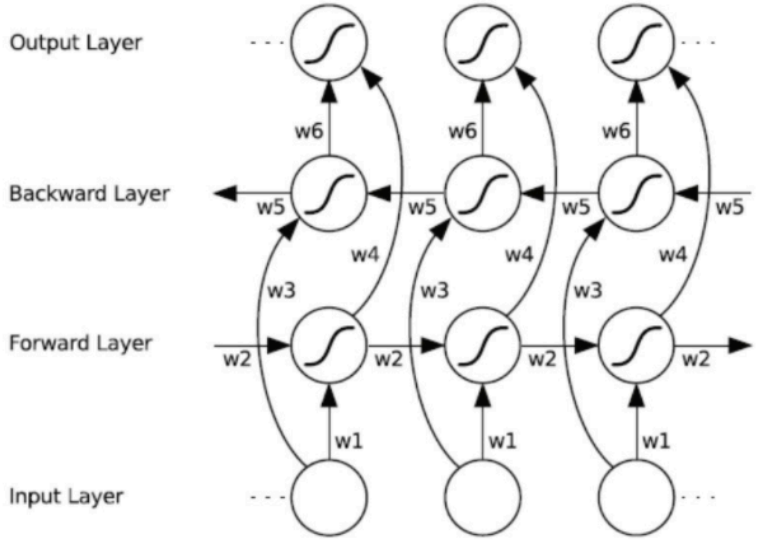
2. Word Embedding Layer

- Using pre-trained word vectors to obtain the fixed word embedding of each word.
- The concatenation of the character and word embedding vectors is passed to a two-layer Highway Network. The outputs of the Highway Network are two matrices: $X \in \mathbb{R}^{d*T}$ for the context and $Q \in \mathbb{R}^{d*T}$ for the query.

Model

3. Contextual Embedding Layer

- Using a LSTM on top of the embedding to model the temporal interactions between words.
- We place an LSTM in both directions, and concatenate the outputs of the two LSTMs. Hence we obtain $\mathbf{H} \in \mathbb{R}^{2d*T}$ from the context word vectors X, and $\mathbf{U} \in \mathbb{R}^{2d*J}$ from query word vectors Q.



Model

4. Attention Flow Layer

- Attention flow layer is responsible for linking and fusing information from the context and the query words.
- The similarity matrix is computed by

$$\mathbf{S}_{tj} = \alpha(\mathbf{H}_{:t}, \mathbf{U}_{:j}) \in \mathbb{R}$$

$$\boldsymbol{\alpha}(\boldsymbol{h}, \boldsymbol{u}) = \boldsymbol{w}^T[\boldsymbol{h}; \boldsymbol{u}; \boldsymbol{h} \circ \boldsymbol{u}]$$

- C2Q attention signifies which query words are most relevant to each context word.

$$\mathbf{a}_t = \mathrm{softmax}(\mathbf{S}_{t:})$$

$$\tilde{\mathbf{U}}_{:t} = \sum_j \mathbf{a}_{tj} \mathbf{U}_{:j}$$

Model

4. Attention Flow Layer

- Attention flow layer is responsible for linking and fusing information from the context and the query words.
- The similarity matrix is computed by

$$\mathbf{S}_{tj} = \alpha(\mathbf{H}_{:t}, \mathbf{U}_{:j}) \in \mathbb{R}$$

$$\boldsymbol{\alpha}(\boldsymbol{h}, \boldsymbol{u}) = \boldsymbol{w}^{T}[\boldsymbol{h}; \boldsymbol{u}; \boldsymbol{h} \circ \boldsymbol{u}]$$

- Q2C attention signifies which context words have the closest similarity to one of the query words and are hence critical for answering the query

$$\mathbf{b} = \mathrm{softmax}(\mathrm{max}_{col}(\mathbf{S}))$$

$$\tilde{\mathbf{h}} = \sum_t \mathbf{b}_t \mathbf{H}_{:t}$$

$\tilde{\mathbf{h}}$ is tiled T times across the column, thus giving $\tilde{\mathbf{H}} \in \mathbb{R}^{2d \times T}$

Model

4. Attention Flow Layer

- Attention flow layer is responsible for linking and fusing information from the context and the query words.
- The similarity matrix is computed by

$$\mathbf{S}_{tj} = \alpha(\mathbf{H}_{:t}, \mathbf{U}_{:j}) \in \mathbb{R}$$

$$\boldsymbol{\alpha}(\boldsymbol{h}, \boldsymbol{u}) = \boldsymbol{w}^{T}[\boldsymbol{h}; \boldsymbol{u}; \boldsymbol{h} \circ \boldsymbol{u}]$$

- Finally, the contextual embedding and the attention vectors are combined together to yield G.

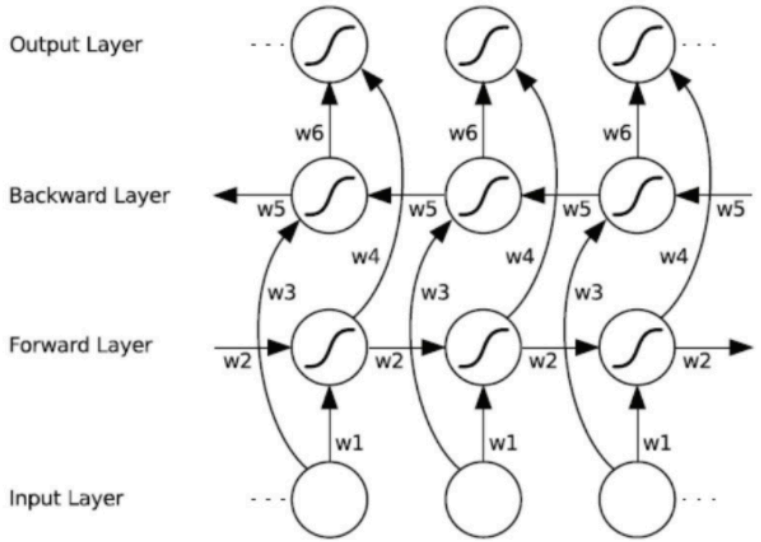
$$\mathbf{G}_{:t} = \boldsymbol{\beta}(\mathbf{H}_{:t}, \tilde{\mathbf{U}}_{:t}, \tilde{\mathbf{H}}_{:t})$$

$$\boldsymbol{\beta}(\boldsymbol{h}, \tilde{\boldsymbol{u}}, \tilde{\boldsymbol{h}}) = [\boldsymbol{h}; \tilde{\boldsymbol{u}}; \boldsymbol{h} \circ \tilde{\boldsymbol{u}}, \boldsymbol{h} \circ \tilde{\boldsymbol{h}}]$$

Model

5. Modeling Layer

- The modeling layer captures the interaction among the context words conditioned on the query.
- Using two layers of bi-directional LSTM.
- The output is passed onto the output layer to predict the answer.



Model

6. Output Layer

- The phrase is derived by predicting the start and the end indices of the phrase in the paragraph.

$$\mathbf{p}^1 = \text{softmax}(\mathbf{w}_{(\mathbf{p}^1)}^\top [\mathbf{G}; \mathbf{M}])$$

$$\mathbf{p}^2 = \text{softmax}(\mathbf{w}_{(\mathbf{p}^2)}^\top [\mathbf{G}; \mathbf{M}^2])$$

Thanks