Top-K Attention Mechanism for Complex Dialogue System

Chang-Uk Shin, **Jeong-Won Cha**
{papower1, **jcha**}@changwon.ac.kr

Adaptive **I**ntelligence **R**esearch Lab.,
Changwon National University, Republic of Korea

Abstract

- Dialogue modeling is usually attempted with a sequence or combination of Recurrent Neural Network and Attention Mechanism.
- Most of the operations in RNN cannot be parallelized because each operation needs the result of the previous timestep.
- In this paper, we use only CNN, Attention, and Pointer Network to model the 'NUC(Next Utterance Classification)' task.
- We proposed 'Top-K Attention'. The 'Top-K Attention' calculates the degree of mutual concentration for highly dependent K element which is selected by the operation itself.
- The experimental results show that our architecture achieves the better performance and the faster inference time.

- The dialogue system should be able to understand user's intent using the dialogue history processed and user's profile. So the understanding or extracting user's intent from the history of dialogue and profile of user is included in main goal of the dialogue manage system.

- As the natural language processing researches using the artificial neural network progress actively, the artificial neural network model is also applied to the dialogue modeling task. These models usually be composed of RNNs.

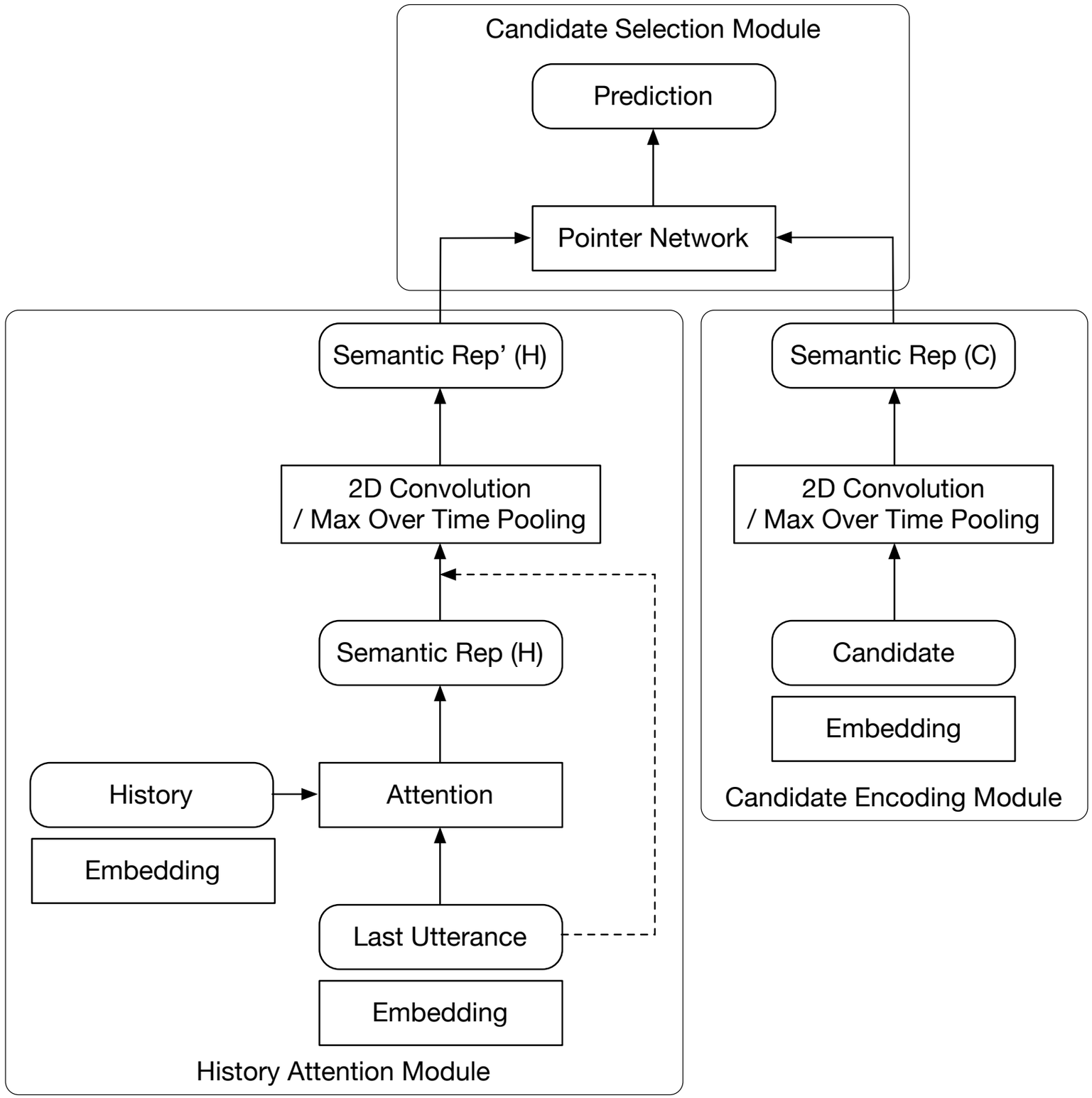
- RNN requires the hidden state of the previous timestep when performing every timestep operation. Therefore, a lot of operations cannot be performed in parallel, which degrades performance.

- In this paper, we point out the computational efficiency of RNN and design the next utterance classification model using only CNN, top-K attention and pointer network.

Dataset Preprocessing

- The DSTC7 track1 dataset distributed by DSTC committee is already quite preprocessed. So we additionally processed preprocessing listed below.
  - Delexicalizing package names to special token “<PACKAGE>”.
  - Delexicalizing URLs to special token “<URL>” using regular expression.
  - Delexicalizing file paths to “<PATH>” using regular expression.
  - Restricting vocabulary to the top 10,000 most common words.

Proposed Architecture



- In this paper, we proposed the Top-K History Attention module to gather context information from dialogue history. Our novel Top-K Attention calculates the degree of mutual concentration like conventional attention mechanism. Then, It performs weighted sum by from extracted top K elements from the sequence those are most closely related to the current element.

$$f_{ij} = W^{(2)}(\tanh(W^{(1)}(l_i; h_j)))$$

$$w_{ij} = \frac{\exp(f_{ij})}{\sum_k \exp(f_{ik})}$$

$$t(x, X, k) = \begin{cases} x, \text{if } x \geq \max(X, k) \\ 0, \text{otherwise} \end{cases}$$

$$a_i = \sum_{j=0}^{n} t(w_{ij}, w_i, k) h_j$$

➤ The experimental results of the proposed architecture are summarized in table 1.

- **Official baseline** : Official Dual-Encoder baseline model. The performance was taken from the official GitHub repository. (* Inference time of the model was measured independently on the same environment that the other experiments done.)

- **Without history attention** : The history attention module does not take attention operation over utterance history but do the convolution operation over the last utterance embedding.

- **History attention** : The proposed architecture that performs the conventional attention operation over dialogue history with last utterance. Providing additional information to the model can improve the performance of the model. But the overall performance was not improved. We interpreted this result as not being able to extract adequate information to take into account the attention over all history utterance.

- **Top-K attention** : According to the assumption from the last experiment, we restrict the attention operation to find the relevant words. So we renew the standard attention mechanism to select most relevant element before summation of the information of sequence.

Experiment Settings - 2

- **Unsupervised embedding** : Above models adopted one unsupervised trained Glove embedding. In this experiment, we additionally compose two more word embedding matrix. One is trained on training dataset with Skip-Gram algorithm and the other one is random initialized word embedding.

- **Skip-connection** : Skip-connections are extra connections between not directly connected layers. It is broadly adopted from various neural network architectures to reduce the training costs and to achieve higher performance.

➤ The table 2 summarizes the experimental results of the parameter ‘history window’.

- **History window** : When we insert the whole history sequence as input, the performance of model was decreased by a large margin. So we conducted to find the optimal number of the history. (** When we increase the window size to 11 or more, the memory usage of the device was exceeded. The experiments listed in upper table was performed with history window size 10.)

experiments	R@1	R@2	R@5	R@10	R@50	inference time
official baseline	8.32%	13.36%	24.26%	35.98%	80.04%	2,898ms*
without history attention	10.76%	16.70%	27.80%	38.12%	80.10%	66ms
+history attention	5.84%	9.90%	17.86%	27.40%	71.62%	239ms
+top-K attention	14.56%	20.82%	31.96%	43.14%	83.38%	263ms
+unsupervised embedding	19.42%	27.92%	40.18%	51.38%	89.24%	374ms
+skip connection	**20.92%**	**30.10%**	**43.04%**	**54.76%**	**91.40%**	394ms

Table 1 : Trend of Experimental Results by the Architecture Modification

history window	R@1	R@2	R@5	R@10	R@50	inference time
3	12.82%	19.00%	29.44%	39.92%	81.44%	125ms
5	14.36%	**21.04%**	31.98%	42.16%	83.00%	177ms
7	14.30%	20.90%	**32.42%**	42.62%	**83.46%**	219ms
10**	**14.56%**	20.82%	31.96%	**43.14%**	83.38%	263ms
unlimitted	5.84%	9.90%	17.86%	27.40%	71.62%	239ms

Table 2 : Trend of Experimental Results by the Size of the History Window