Top-K Attention Mechanism for Complex Dialogue System

Chang-Uk Shin[a], Jeong-Won Cha[b*]

[a]Department of Eco-friendly Offshore Plant FEED Engineering, Changwon National University, Republic of Korea

[b]Department of Computer Engineering, Changwon National University, Republic of Korea

{papower1, jcha}@changwon.ac.kr

Abstract

Nowadays, natural language processing tasks such as dialogue modeling, question answering, sentence classification are usually attempted with a sequence or combination of Recurrent Neural Network(RNN), Convolutional Neural Network(CNN), and attention mechanism(Attention). RNN-based dialogue modeling architecture encodes given a history of the dialogue. Then the representation of the history is passed into the generator or classifier module to create or classify the system utterance. Most of the operation in RNN cannot be parallelized because each operation needs the result of the previous timestep. Thus, the total inference time is relatively longer than CNN and Attention-based models. In this paper, we use only CNN, Attention, and Pointer Network to model the 'Next Utterance Classification' task. We use Recall @ K, which measures a correct answer in the top K among the 100 candidates listed for the performance measure. The proposed system achieves R @ 1 20.92%, which exceeds the baseline performance. We show that it is possible to extract useful information from long utterance history without RNN and to solve the next utterance classification problem based on the information.

Introduction

The dialogue system should be able to understand user's utterance using the information of the given utterance history and user's profile, and consequently to be able to solve user's needs. The main purpose of the dialogue system research is to extract useful features from previous utterances and write the following utterances based on them.

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. At first, a chitchat model was studied which receives one utterance by using RNN encoder-decoder structure and responds appropriately. After that, research continues how to encode the conversation and how to write the corresponding response.

The difficulty of dialog system modeling varies according to given environment such as domains. The more domains you are trying to learn, the larger size of the dataset you need to learn the model. Also, learning an artificial neural network model from a large amount of dataset is sometimes very tricky. For example, processing relatively long sequence with a small frequency, processing words occurring at a low frequency, and time required for actual learning is increased in proportion to the size of the dataset. Also, in the case of RNN, it needs the hidden state of the previous timestep when performing every timestep operation. Therefore, many operations cannot be performed in parallel, which degrades performance.

In this paper, we point out the computational efficiency of RNN and design and exploit the next utterance classification model using only CNN, top-k attention and pointer network. We note the given conversation history by word, characterize it by convolution of the result and candidate, and finally, enter into the pointer network to select the final candidate.

All operations employ only CNN, attention, and pointer network so that there is no difference in reasoning time according to input length. Experiments on the ubuntu dialogue dataset of the DSTC7 track1 sentence selection with the proposed structure show that R @ 1 achieves 20.92%, which exceeds the performance of the dual encoder based model provided by the baseline, which is 8.32%.

Nonetheless, it took only 14% of the time to deduce the Dual Encoder. We demonstrate performance and reasoning time through experiments and verify the effectiveness of the proposed top-K attention.

Related Research

(Lowe 2016a, Lowe 2016b) pointed out the inadequacy of the evaluation of the generation-based dialogue model and proposed the classification-based task named NUC. Usually the performance of the generation-based dialog system was

* corresponding author.

measured based on the similarity of the generated system utterance to the pre-written gold-standard utterance. However, this has been found to have a very weak correlation with actual human evaluation.

NUC is a task that analyzes the given context and selects the utterances that follow from the given candidates. They asserted the following points as advantages of this evaluation: the ability to control the degree of difficulty by adjusting the number of candidates, the intuitiveness of the performance measure, the ability to use the actual dialog system as an extension, can be controlled to provide strong constraints on the output, and that the response can be guaranteed to be fluent.

(Lowe 2016a) also pointed out that a large amount of corpus is essential for successfully model this task and has released a large-scale dialogue dataset collected in ubuntu IRC logs. The dataset is a multi-turn conversation of 1 million and can be modeled by NUC. Since several speakers can speak at the same time in one chat room, they extracted only two-person conversations which exceeds three turns.

Dataset

The dataset we used for the experiment is the ubuntu dialogue dataset provided in the sentence selection track of DSTC7(Koichiro Yoshino et al, 2018). It was quite preprocessed at the time of distribution, but we once again performed lemmatize using Stanford tagger (Toutanova and Manning 2003). We collect package lists that are registered in the apt repository, tokenize package names, and delexicalized the URLs and file paths to URL / PATH using regular expressions. Nevertheless, the size of the dictionary was huge, so there were many restrictions on learning. Therefore, we changed all to the unknown, except only top 10,000 words in frequency. The overall coverage of the top 10,000 words by frequency was 98.10%.

Proposed Model

The proposed structure consists of three modules. First is the history attention module that extracts features using the relationship between the utterance history and the last utterance, second is the candidate encoding module that encodes given candidates, and the last module is candidate selection module that selects the final answer based on the results of two modules.

Top-K History Attention Module

The History attention module uses the previous m utterances and the last utterance to extract the optimal features. We use m utterances removing sentence separations and concatenate them word by word. We tested m by changing it to 1, 3, 5, 7, 10 and got the highest performance at 10.

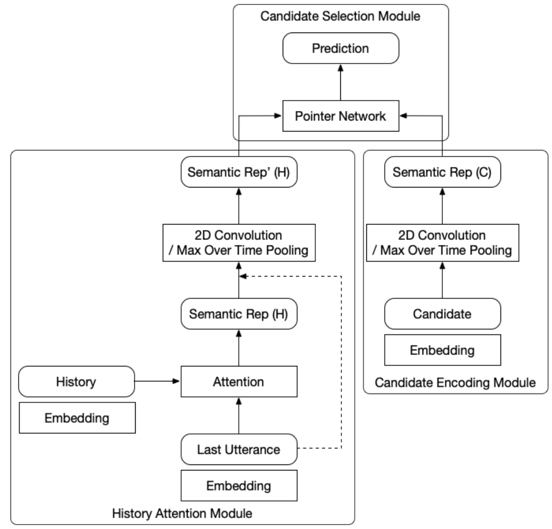

Figure 1 : The structure of the proposed methods

However, when we raised m over 10, the memory usage was too large to test.

For efficient computation, we adopted our novel top-k attention mechanism. like standard attention mechanism, the top-k attention calculates the degree of mutual concentration using one element and one sequence. The difference is that we do not perform weighted sum by multiplying all weights by all elements. Instead, we perform a weighted sum by extracting top k elements from the sequence that we believe are most closely related to the current element.

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

$$\mathrm{w_{ij}} = \frac{\exp(f_{ij})}{\sum_k \exp(f_{ik})} \quad (2)$$

$$t(x, X, k) = \begin{cases} x, & if x \geq \max(X, k) \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

$$a_j = \sum_{i=0}^{n} t(w_{ij}, w_i, k) h_i, \quad (4)$$

where, l_i is the i-th word in the last utterance, h_j is the j-th word in the history utterance. In Eq. (1), each word in the history is connected to each word in last utterance and input to the MLP that estimates the attention weight. The result, f, is taken by softmax and then multiplied by all words in the history to create an attended representation h (2, 4). In this case, we do not use all of the results of softmax, but we set it so that only the top k values can be left through (3). We call this operation top-k attention. In this paper, k is set to 10 for all experiments.

We use the top-k attention mechanism to reduce a number of computations considerably and to characterize only the words most closely related to each word in the last utterance.

In this way, we can cope with noise more robustly than existing attention mechanisms that receive attention from all words, so it is useful to extract only the information that our model needs.

In Experiment 5 to be described later, the performance is again improved by using the last utterance word embedding, again, with the semantic representation (H), which is the result of the history attention module(shown by a dotted line in Fig.). This is called skip-connection(Huang, Liu and Maaten, 2018, Srivastava, Greff and Schmidhuber, 2015, He et al, 2015). In this paper, it can be understood that the embedding of the original word and the information extracted from the history are used together as the variance representation of the last utterance.

We obtain a single utterance variance representation by inputting final attention and history attention results to the 2D CNN and the max-over-time-pooling layer. In all experiments, we set filter sizes to 2, 3 and 4, and the number of filters per each filter size to 32. Therefore, size of the variance expression after performing max-over-time pooling of the equation is 96.

Candidate Encoding Module

In the experiment, we must choose one utterance among the 100 candidate utterances. then, enter each candidate into the 2D CNN structure described above to acquire a distributed representation. We use the same structure of CNN, but we do not share the weight.

Candidate Selection Module

We encode utterance distributed representations and candidates, respectively, which have been supplemented by utterance history up to the present. Two distributed representations got 96 dimensions. A structure for selecting one of the given candidates using a pointer network has been applied to machine reading comprehension(Wang, 2017). In this paper, we adopt the structure, connect the utterance distributed representation as many as the number of candidates, and select a candidate using the pointer network.

$$f_i = W^{(2)}(\tanh(W^{(1)}(a; c_i))) \tag{5}$$

$$p_i = \frac{\exp(f_i)}{\sum_j \exp(f_j)} \tag{6}$$

$$\hat{p} = \operatorname{argmax}(p_0, \ldots, p_{100}), \tag{7}$$

where a denotes utterance distributed representation, and c is candidate. We perform attention between a and c using MLP and submit the candidate with the highest attention value as a prediction.

Experiments

Experimental Settings

In the experiment, we used the dataset of DSTC7 tack1. We randomly extracted 8/9 of the distributed train datasets and used them as the training dataset and the remaining 1/9 as the validation dataset. We used the distributed development dataset as the test dataset.

We used the performance of the dual encoder provided by DSTC7 as the baseline. Experiments were performed according to known experimental procedures. Dual Encoder is an artificial neural network structure proposed to perform NUC in (Rowe 2015). The Dual Encoder binds two RNNs with a RNN encoding the context and the other RNN encoding the candidate. The distributed representation resulting from both RNN is entered into the regression module to model the degree to which it is matched appropriately.

Then, the model submit the candidate with the highest value among the given candidates as the final correct answer. They randomly sampled one of the 99 negative samples to prevent bias during learning and used CommonCrawl word embedding as pertained embedding.

In our experiments, we set the hyper-parameters as follows. The number of 2D CNN filters was 32 per each filter size, and size of the filters was 2, 3, and 4. These are the same for history and candidate operations. We used the Adam as an optimizer and set the learning-rate to 0.001 and epsilon to 0.01. All embedding sizes were set to 25. Then, we loaded the pretrained glove twitter 25d embedding and fine-tuned it.

Experimental Results & Analysis

Table 1 shows experimental results of the baseline, the experimental results of the baseline we have tested and the experimental results of the proposed system. Experiment 1 is the experiment using the simplest structure that selects the given candidate using only last input utterance. In other words, the model encodes the last utterance with the word 2D CNN and choose one of the candidates by pointer network value. Since the model does not use history, does the smallest amount of computation in the experiments have presented, and nevertheless, achieve performance above the baseline.

Experiment 2 is the experiment in which history attention is added to Experiment 1. Providing additional information to the model can improve the performance of the model, and dialogue history is good features of conversation modeling. Experiment 2 performed over all of the words and the attention results with history. However, Experiment 1 consistently outperformed Experiment 2 from R @ 1 to R @ 50. We interpreted this result as meaningless to extract attention to all the words in the history and not to extract appropriate information because it considers the attention of unnecessary words. Therefore, we apply the top-K attention to extract only relevant information to the next experiment.

Experiment 3 is an experiment in which top-k attention is added. We try to verify the effectiveness of the proposed top-k attention in this experiment. Experimental results showed a significant improvement in the performance of R

@ 1 by 3.80%p increase. The top-K operation to find relevant words has been added, so the inference time is somewhat longer. Nevertheless, it is still 10 times shorter than the baseline.

Experiments 4 and 5 are tuning experiments performed to improve the performance of the proposed method further. Experiment 4 is an experiment with improved embedding. We used the glove word embedding with word embedding learned with skip-gram and randomly initialized word embedding. All dimension of 3 word embedding was set to 25. We have learned skip-gram word vectors by inputting all given training corpus. In this way, the glove word vector learned by twitter can represent general knowledge in a vast domain, and the skip-gram word vector can express meaning in a specific domain. We also added 25-dimensional random initialized word embedding to allow the model to capture the information that cannot be obtained in context. And in our experiments, the best performance was achieved when all 3 word-embeddings were set to be fine-tuned, and the difference was about 5% at R @ 1.

Experiment 5 is an experiment in which the structure is tuned once again to the structure improved by experiment 4. In the last experiments, only the results of attention calculation of the last utterance were input to the next 2D CNN layer. In Experiment 5, we used the word embedding of the last utterance by skip-connection with the last utterance attention result. It's a simple extension, but we've got a 1.50%p improvement in R @ 1 and a 2.18%p improvement in R @ 2. This is similar to the module proposed by Highway Networks (Srivastava, Greff and Schmidhuber, 2015). Semantically, it can be expressed as considering both the word embedding of the last utterance itself and the history attended by the previous history.

The samples in training corpus have average 5.49 history utterances(first quartile 3, median 5, and third quartile 7). The higher performance can be obtained by providing more history as input. However, providing many features will cause overfitting problem so that the performance will also decrease. 'Ubuntu dialogue corpus' has up to 75 histories, so setting the upper limit may be a more appropriate modeling method. Therefore, we set the limit of the input history and did the experiments. The results are summarized in [Table 2]. From the experimental results, it can be seen that as the number of history utterances increases, the performance improves and the inference time increases. Also, providing the history over the upper limit makes the model overfit, which shows the performance decrease.

Conclusion

Attempts to model human dialogue are still being researched. Also, RNN-based natural language processing research is continuing. Although the RNN based architecture can extract information regardless of the length of the input, parallelization is limited because the operation must be performed after the operation result of the previous timestep is prepared. Therefore, the computation time is proportional to the length of the input.

In this paper, we tried to solve 'next utterance classification' problem with word level 2D CNN, attention, and pointer network. We use attention just for k words which are executed between history and last utterance and which have the highest attention value. And we limited the number of input history. Experimental results on the DSTC7 track1 dataset show that the proposed model achieves approximately 10 times faster inferencing time than the dual encoder, which is the base model.

The top-k attention mechanism proposed in this paper extracts k words those are highly related to conventional attention and performs attention. We predicted that this could extract more useful information for reasoning than conventional attention and could achieve better performance.

Using the top-K attention mechanism and applying limit of the length of the history proposed in this paper, we extract information within the threshold determined by the hyperparameter. If we can change it more adaptively, we could achieve better performance.

Acknowledgements

This work was supported by Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government. [18ZH1300, Development of programmable interactive media creation service platform based on open scenario] and Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2018-0-00247, Development of Data Self-proliferation and Validation Technology for Deep Learning Using GANs).

References

Ryan Lowe, Nissan Pow, Julian V. Serban and Joelle Pineau. 2016. The Ubuntu Dialogue Corpus: A Large Dataset for Research in Unstructured Multi-Turn Dialogue Systems. arXiv:1506.08909v3.

Ryan Lowe, Julian V. Serban, Mike Noseworthy, Laurent Charlin and Joelle Pineau. 2016. On the Evaluation of Dialogue Systems with Next Utterance Classification. In Proceedings of the SigDial 2016, 264-269.

Koichiro Yoshino, Chiori Hori, Julien Perez, Luis Fernando D'Haro, Lazaros Polymenakos, Chulaka Gunasekara, Walter S. Lasecki, Jonathan K. Kummerfeld, Michel Galley, Chris Brockett, Jianfeng Gao, Bill Dolan, Xiang Gao, Huda Alamari, Tim K. Marks, Devi Parikh and Dhruv Batra. 2018. The 7th dialogue system technology challenge. arXiv preprint.

Kristina Toutanova and Christopher D. Manning. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In Proceedings of HLT-NAACL, 252-259.

Gao Huang, Zhuang Liu and Laurens van der Maaten. 2018. Densely Connected Convolutional Networks. arXiv:1608.06993v5.

Rupesh Kumar Srivastava, Klaus Greff, Jürgen Schmidhuber. 2015. Highway Networks. arXiv: 1505.00387v2.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, 2015. Deep Residual Learning for Image Recognition. arXiv:1512.03385v1.

Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, Ming Zhou. 2017. Gated Self-Matching Networks for Reading Comprehension and Question Answering. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 189-198.