Learning Dynamic Memory Network with Two Views

Chang-Uk Shin[1], *Jeong-Won Cha*[1]

[1]Changwon National University, Republic of Korea

papower1@changwon.ac.kr, jcha@changwon.ac.kr

Abstract

In this paper, we propose an end-to-end goal-oriented dialog system using dynamic memory networks with negative samples. A traditional artificial neural network learned the model using only correct answers. We improved the performance by learning the dialogue model using dynamic memory networks with negative samples for learning. After applying the negative sample, we evaluated the performance of DSTC6 track1 data up to 91.20% accuracy. On the other hand, we obtained performance decrease at track 2, 3 and 5 by applying a negative sample. We confirmed that most of this errors arise from the method of measuring sentence similarity method. Our novel learning mechanism can be easily applied to the problem of choosing the right answer among the various candidates to improve the performance.

Index Terms: dynamic memory network, negative sample

1. Introduction

Goal-oriented dialogue system is one of the dialogue systems which conducts dialogue with the user about the specified domains and goals. This identify the intent of the user during the conversation and achieve the user's desired goal. It should also be able to request appropriate information from the user if the given information is insufficient.

There have been attempts to solve natural language processing problems using end-to-end structures such as sequence-to-sequence and memory networks. For dialog management, there was an attempt to use a single structure for language understanding, dialog management and utterance generation. This end-to-end structure has the advantage of being able to learn only if there is data without knowledge of the domain of the conversation.

Sequence-to-sequence is a structure that acquires a distributed semantic representation corresponding to the entire utterance from the input utterance using the RNN(Recurrent Neural Network) encoder, and outputs the sequence by inputting the distributed semantic expression to the RNN decoder. Sequence-to-sequence is suitable for these natural language processing problems because these problems such as morphological analysis are attached to the label of each sentence element.

The dynamic memory network [1] is proposed to provide an appropriate response to the query. When we enter knowledge and queries separately into this network, the memory network computes the information to create the memory. Finally, it computes the memory and query to generate the correct answer.

Before Deep Learning, POMDP (Partially Observable Markov Decision Process) based dialogue system has been studied. The POMDP-based dialog system consists of a language understanding model for analyzing input utterances, a dialogue model, a policy model, a utterance generation model, and a reward function. And we study the dialogue model and the policy model in the direction that maximizes the reward generated by the reward function [2].

We applied a negative sample to the dynamic memory network to improve performance. We anticipate that improving models from information that is not the correct answer can easily be applied to many areas.

2. Related Research

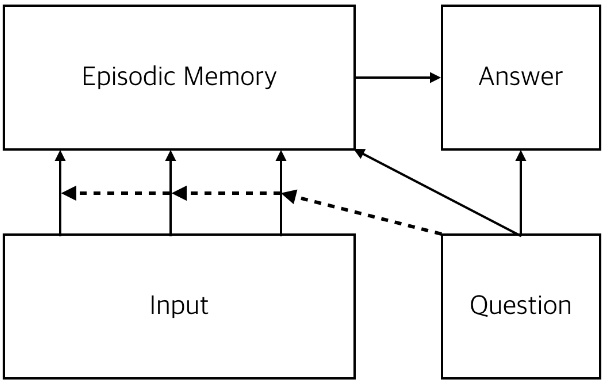
In order to coordinate the strengths and weaknesses of the traditional approaches such as POMDP and end-to-end approaches, there is a case that uses an end-to-end trainable network with a module responsible for linking the two approaches [3]. They used a database to grasp a clear attribute relationship and defined two modules, an intent network and a belief tracker, in order to grasp user intent. The policy network, which received information from the intent network, the belief tracker, and the database, combines the information and transmits it to the generation network, and the generation network generates the sentence.

Jason D. Williams et al. [4] proposed a new architecture, hybrid code networks (HCN), in which domain knowledge and action templates were applied to create a dialogue model. The HCN consists of an RNN module, a domain-specific software module, a domain-specific action template module, and a conventional entity extraction module. They reduced the learning complexity by inputting feature vectors extracted from the speech into the RNN, deducing action templates, and repopulating the inferred templates with entities. They also suggested a way to learn HCN in Reinforcement Learning.

In A. Bordes et al. (2017) [5], they defined the restaurant reservation task with four subtasks. And they learned by several modeling techniques such as memory network and measured their performance. The authors have used the match type features to solve the OOV(Out-Of-Vocabulary) problem.

In T. Mikolov et al. (2013) [6], they used negative sampling to improve word vectors. Word vector estimation before without negative sampling performed softmax operation. Since softmax must perform matrix multiplication for all words listed in the dictionary, the amount of computation is very high. To improve it, they proposed a negative sampling by modifying the Noise Constrastive Estimation (NCE) [7]. Using negative sampling, they obtained less computation in learning than softmax and NCE, and achieved high performance.

Figure 1: *Dynamic Memory Network*



3. Proposed Method

3.1. Dynamic Memory Network

We learned the dialog model using dynamic memory network structure. Dynamic memory network [1] has an input module and a question module. We update the memory of the episodic memory module using the representation created from both modules. Memory is a summary of the conversations and queries that have been conducted so far. When memory update to the last input statement is completed, the answer module generates the correct answer using the generated memory and question.

The reasoning process is as follows. First, we encode the word sequence input from input module and question module defined as:

$$i_t = GRU(w_t^I, i_{t-1}), \tag{1}$$
$$q_t = GRU(w_t^Q, q_{t-1}). \tag{2}$$

We used GRU(Gated Recurrent Unit) [8] as a sentence encoder. Here, w_t is the words in the input and query, i_t and q_t are hidden states of each network at time t.

Then we update the memory module using the input representation created from the input module and the question representation generated from the question module. We use the attention mechanism to properly use the input and question representations as follows:

$$g_t^i = G(z(i, m, q)), \tag{3}$$

where, G is a scoring function that input a feature vector composed of input i, previous memory m, and query q. G is a simple multi-layer perceptron takes above feature vector and has one output node. If there is no previous memory, we use q instead of m.

We use the score when updating the memory from the input statement. Given the scalar score, equation 4 will be able to manage the input statements and the previous state together.

$$h_t^i = g_t^i GRU(i_t, h_{t-1}^i) + (1 - g_t^i)h_{t-1}^i \tag{4}$$

Memory can be stacked in multiple layers. In this case, the memory of each layer is generated by computing the memory state of the previous layer and the GRU state of this layer. Equation 5 describes the formula used for updating memory with previous layer and hidden state h. At this time, we use fully connected layer as function f.

$$m^i = f(m^{i-1}, h^i) \tag{5}$$

Finally, The answer module generates the prediction taking account the final memory and question vector as equation 6.

$$a = f(m^i, q^i), \tag{6}$$

where, in this paper, we used GRU decoder (as function *f*) to generate the response sentence. Figure 2 describes the proposed answer module.

Figure 2: *Answer Module of Proposed Method*

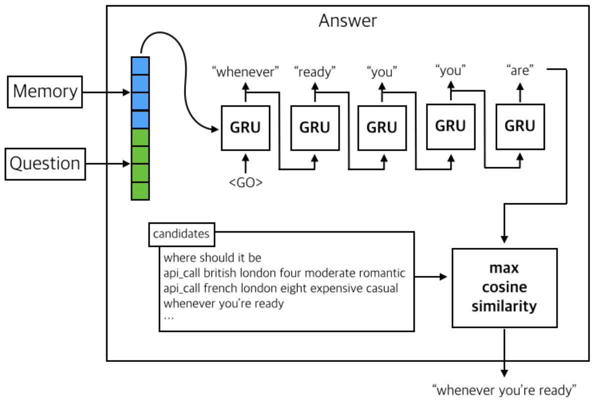
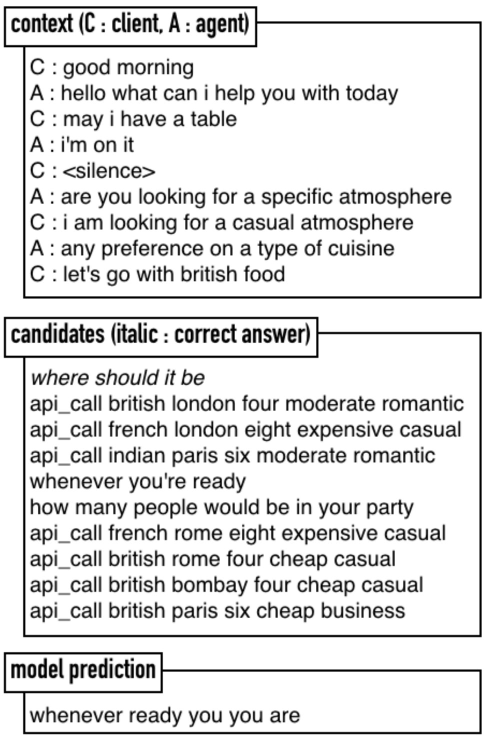


Figure 3: *DSTC6 track1 data sample*



3.2. Negative Sample

The DSTC6 track1 corpus we used for the training consisted of input, query, correct answer, and answer candidates. We regard the given candidates as negative sample and suggest a new loss function to use them. We assume that the new loss function has a performance advantage by handling more information than the loss function using only the correct answer. The proposed loss function measures the loss using the correct answer and the negative sample, respectively, and presents the difference as a loss.

$$e_t = e_p - e_n \tag{7}$$

$$e_p = \sum_t E(y_t, \hat{y}_t) \tag{8}$$

$$e_n = \sum_t E(y_t, \tilde{y}_t), \tag{9}$$

Where, e_t is the final loss created from the correct answer and the negative sample. e_p and e_n are the losses created from the correct and negative sample respectively, y_t is the system output word in time step t, $\hat{y}_t$ and $\tilde{y}_t$ are the t-th word of the correct answer and negative sample. Figure 3 shows an example of DSTC6 track1 corpus. From the above example, y_t can be set to 'whenever ready you you are', $\hat{y}_t$ to 'where should it be', and $\tilde{y}_t$ to be one of the remaining candidates except $\hat{y}_t$.

4. Experiments

4.1. Experiments Setting

We used the DSTC6 track1 corpus for the experiment. The corpus information is shown in table 1. Task1 ~ 4 are four subtasks to accomplish a given goal, and task5 is a combination of tasks1 ~ 4, which can result in any of the dialogs of task1 ~ 4. Task3 is given information such as location, price, and phone number of several restaurants as input, so that the number of input turns is relatively large compared to other tasks.

We used the same network hyper-parameters for all experiments, set the batch size to 100, the word embedding size to 100, the question, input, and memory module hidden size to 100. The memory layer is set to 3. We adapted early stopping, dropout, and l2 regularization.

Table 1: *DSTC6 training dataset.*

Task	average input turn	average word per turn
1	9.9	6.9
2	11.6	7.8
3	41.8	3.9
4	13.3	4.4
5	41.2	5.2

In order to verify the effectiveness of the proposed negative sample, we first train the network with the baseline and conduct experiment with the loss function in the same environment with the negative sample loss function.

The problem with DSTC6 track1 is to select a response from the candidates that is appropriate for the query. However, there is a possibility that the results generated in the dynamic memory network may not exist in the candidates. Therefore, we measure cosine similarity between each of the candidates and the generated result and select one with the highest similarity as the correct answer.

$$d_{cosine} = \frac{\sum_{k=1}^{n} w_{pk} \cdot w_{ck}}{\sqrt{\sum_{k=1}^{n} w_{pk}^2} \cdot \sqrt{\sum_{k=1}^{n} w_{ck}^2}} \tag{10}$$

Table 2: *experimental results*

Exp	Task	Test Acc
baseline	1	88.65%
	2	84.85%
	3	66.50%
	4	69.05%
	5	69.60%
with negative sample loss	1	91.20% (+2.55%)
	2	78.20% (-6.65%)
	3	66.40% (-0.10%)
	4	70.30% (+1.25%)
	5	69.50% (-0.10%)

Table 3: *Error example for DSTC6 task2. The words that matched the correct sentence were marked as bold.*

	sentence
baseline model prediction	api_call indian **bombay four cheap business**
baseline candidate selection	api_call spanish bombay four cheap business
proposed model prediction	api_call **spanish bombay** two **cheap business**
proposed model candidate selection	api_call spanish bombay two cheap business
gold standard	api_call spanish bombay four cheap business
baseline model prediction	api_call **french** rome **two cheap** casual
baseline candidate selection	api_call french paris two cheap romantic
proposed model prediction	api_call **french paris** six **cheap romantic**
proposed model candidate selection	api_call french paris six cheap romantic
gold standard	api_call french paris two cheap romantic

The experimental result is summarized in table 2. In our experiments, the negative sample showed a maximum performance improvement of 2.55%. task2 showed the lowest performance.

Table 3 summarizes the samples that failed when applying the proposed method, although they were correct on baseline. The words that matched the correct sentence were marked as bold. The first example in the table shows that the baseline and proposed approach matched the same number of words, but other candidates were selected in the process of selecting candidates.

The second example is an example where the proposed approach matches many words compared to the baseline, but the same candidate exists as the result of the model. Both examples suggest that a solution is needed to match the output sentence and candidates well.

5. Conclusions

Artificial neural networks researches are continuing in various problems of natural language processing. Artificial neural networks research can be divided into researches on the generation of new qualities and studies on designing network structures. In this study, we proposed a method to use negative sample as data which can be used with correct answer to learning.

We learned a goal-oriented dialogue model using a dynamic memory network. We also improved performance by using candidates that are not correct answers with learning data. Its performance was improved by up to 2.55% compared to the performance of the existing baseline 91.20%.

Now we use random generated negative samples. If a negative sample can be informatively generated or extracted from a given problem, we will improve the performance of the existing model.

6. Acknowledgements

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (2017R1D1A1B03033534).

7. References

1. Ankit Kumar, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus and Richard Socher, "Ask Me Anything: Dynamic Memory Networks for Natural Language Processing", *arxiv*, 2016.
2. Steve Young, Milica Gašić, Blaise Thomson and Jason D. Williams, "POMDP-based Statistical Spoken Dialogue Systems: a Review", *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1160–1179, 2013.
3. Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes and Steve Young, "A Network-based End-to-End Trainable Task-oriented Dialogue System" *arxiv*, 2017.
4. Jason D. Williams, Kavosh Asadi and Geoffrey Zweig, "Hybrid Code Networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning", arxiv, 2017.
5. Antoine Bordes, Y-Lan Boureau and Jason Weston, "Learning End-to-End Goal-Oriented Dialog", *arxiv*, 2017.
6. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado and Jeffrey Dean, "Distributed Representations of Words and Phrases and their Compositionality", *arxiv*, 2013.
7. Michael U. Gutmann and Aapo Hyvärinen, "Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics", *The Journal of Machine Learning Research,* 13:307-361, 2012.
8. Kyunghyun Cho, Bart van Merriënboer and Dzmitry Bahdanau, "On the Properties of Neural Machine Translation: Encoder-Decoder Approaches", *arxiv*, 2014.