END-TO-END DYNAMIC QUERY MEMORY NETWORK FOR ENTITY-VALUE INDEPENDENT TASK-ORIENTED DIALOG

Chien-Sheng Wu, Andrea Madotto, Genta Indra Winata, Pascale Fung

Human Language Technology Center
Department of Electronic and Computer Engineering
The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong
[cwuak, eeandreamad, giwinata]@ust.hk, pascale@ece.ust.hk

ABSTRACT

In this paper, we propose an end-to-end Dynamic Query Memory Network (DQMemNN) with a delexicalization mechanism for task-oriented dialog systems. The added dynamic component enables memory networks to capture the dialog's sequential dependencies by using a context-based query. Besides, the delexicalization mechanism reduces learning complexity and it alleviates the out-of-vocabulary entity problems. Experiments show that DQMemNN outperforms original end-to-end memory network models on bAbI full-dialog task by 3.1% per-response and 39.3% per-dialog accuracy. In addition, the proposed framework achieves a promising average per-response accuracy of 99.7% and per-dialog accuracy of 97.8% without hand-crafted rules and features.

***Index Terms*—** Task-oriented Dialog Systems, Memory Network, Delexicalization, Recurrent Neural Network, Natural Language Processing

1. INTRODUCTION

Task-oriented dialog system has become an increasingly important research area which requires a machine to understand the human intent and generate proper answers to accomplish the assigned tasks via natural language. For instances, they are required to: understand user request, ask for clarification, properly issue API calls for querying knowledge base (KB) and interpret query results. Recently, a corpus called bAbI dialog [1] for training end-to-end task-oriented dialog has been released. The dialog tasks are well-defined in restaurant reservations domain.

Traditionally, these dialog systems have been built as a pipeline, with modules for language understanding, state tracking, action selection, and language generation [2, 3]. Even though those systems are known to be stable via combining domain-specific knowledge and slot-filling technique, they have limited ability to generalize into new domains and the dependencies between modules are quite complex. On the other hand, end-to-end approaches using recurrent neural networks (RNNs) are attractive solutions [4, 5], since they directly map the dialog history to the output responses. A key advantage is that the latent memory of RNN can be represented as a dialog state, obviating the need for hand-crafted state labels. However, these models may be inherently unstable over long time sequences because the memories are the RNN hidden states.

Promising results have been shown by using End-to-End Memory Network (MemNN) [1, 6, 7], which are neural networks with a recurrent attention model over an external memory. Besides, the multiple hop mechanism over the global memory is experimentally crucial for good performance on reasoning tasks. However, one major drawback of MemNN is that they are insensitive to represent temporal dependencies between memories. We found out two distinct problems of MemNN in dialog systems: 1) no temporal memory access and 2) weak entity identification. The former influences the conversational semantic since the utterance order is not taken into account. The latter bounds the memory's reasoning ability since the model cannot fully identify the dialog entities.

To solve these problems, we propose a novel entity-value independent framework based on end-to-end Dynamic Query Memory Networks and a recorded delexicalization mechanism. The former captures time step information in dialogs by utilizing RNNs between memory layers to represent latent dialog state and the dynamic query vector. The latter not only decreases the learning complexity but also makes our system scalable into new out-of-vocabulary (OOV) KB or new domains. We first outline the key methodologies used (Section 2), and we discuss the related models presented in literature (Section 3). Then the dataset, the model settings and the obtained results are shown (Section 4).

2. MODEL DESCRIPTION

The two main components of our proposed framework are Dynamic Query Memory Network and recorded delexicalization (RDL). Similarly to [8, 9], the intuition of RDL is to reduce the learning complexity by replacing entities in

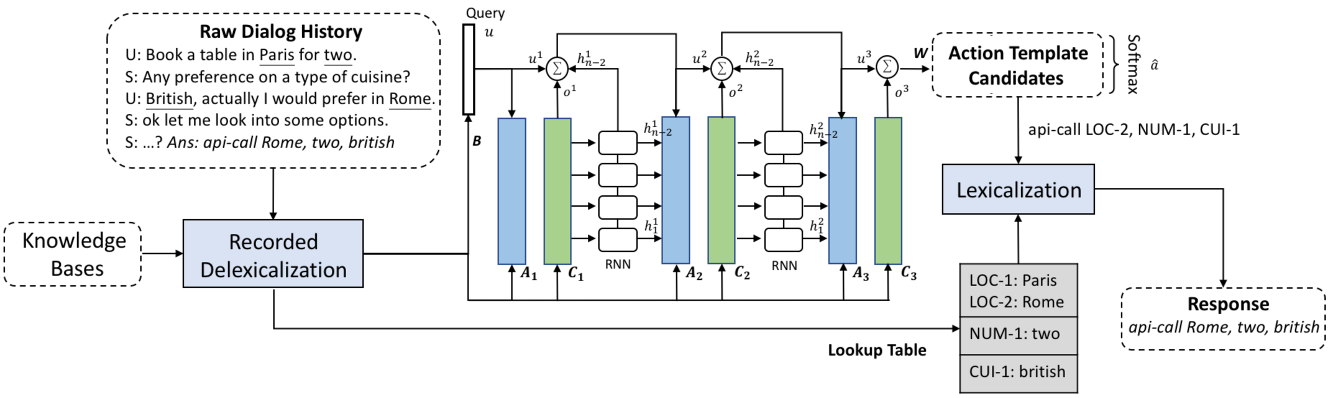

Fig. 1. Entity-value independent dynamic query memory networks for task-oriented dialog

the raw dialog history with a simplified format. In this way, DQMemNN can reason logically on the indexed entity types instead of the actual words based on the information provided by external KBs. Moreover, the number of system utterances can be reduced to the size of action template candidates. At last, DQMemNN outputs the action template is then converted to the final system response by lexicalization. The details are described below.

2.1. Recorded Delexicalization

We utilize the existing KB information to extract entities from both user and system utterances. For example, in the restaurant reservation domain, we extract seven entity types including *[NAME, LOCATION, CUISINE, PRICE, PHONE, ADDRESS, NUMBER]* using simple string matching, since we have the predefined KB information for all the entity values including OOV entities. However, we keep the real number of *[RATING]* in the utterances and leaves the tasks of sorting restaurant ranks to the DQMemNN model. Then we replace each real entity value with its entity type and the order appearance in the dialog, and we also build a lookup table to record the mapping. For example, the first user utterance in Figure 1, "*Book a table in Paris for two*", will be transformed into "*Book a table in [LOC-1] for [NUM-1]*". At the same time, *[LOC-1]* and *[NUM-1]* are stored in a lookup table as Paris and two, respectively. Furthermore, lexicalization is the reverse of RDL. Based on the lookup table, the delexicalized entity can always be reversed, replacing the type with the real world. At last, we build the action template candidates $cand$ by all the possible delexicalization system responses.

2.2. Dynamic Query Memory Networks

Our model takes a discrete set of RDL dialog utterances $s_i, i = 1, ..., N-1$ as input, and the output answer s_N is the one of the action templates in $cand$, where N is the total utterances number in the dialog. Specifically, input $s_1, ..., s_{N-2}$ are the utterances (stories) stored in memory, and s_{N-1} is the initial query (question).

2.2.1. Memory Network

We briefly introduce the original end-to-end memory network structure in [6]. MemNN is made of input memory cells $\{m_i\}$ and output memory cells $\{c_i\}$, where $i \in [1, ..., N-2]$. The memories are obtained by transforming input stories using two embedding matrices $\mathbf{A}$ and $\mathbf{C}$ that map each token into word embeddings. Similarly, the query s_{n-1} is embedded to obtain an internal state u via another embedding matrix $\mathbf{B}$. Then, each s_i is encoded into $\{m_i\}$ and $\{c_i\}$ by summing up the word embedding. The attention weights are computed to determine the relevance between u and each m_i.

$$p_i = Softmax(u^T m_i), \tag{1}$$

where $Softmax(z_i) = e^{z_i}/\Sigma_j e^{z_j}$. Then, the model read out memory o by the weighted sum

$$o = \sum_i p_i c_i. \tag{2}$$

The memory layers can be extended and stacked for k hop operations. In this way, the $k+1$ hop takes the output of k hop as the input query, that is, $u^{k+1} = u^k + o^k$. At last, the answer prediction is performed by

$$\hat{a} = Softmax(\mathbf{W}(o^K + u^K)) \tag{3}$$

where $\hat{a}$ is the prediction distribution, $\mathbf{W} \in \mathbb{R}^{|cand| \times d}$ is the final weight matrix, and K is the maximum hop predefined. Note that $\hat{a}$ shows the probability distribution over all action template candidates.

2.2.2. Dynamic Query Components

To capture the sequential dependencies of dialog utterances, we adopt the idea from [10, 11], whose model can be seen as a bank of gated RNNs, whose hidden states correspond to latent concepts and attributes. Therefore, to obtain a similar behaviour, DQMemNN adds a recurrent architecture between hops. We use the output memory cells $\{c_i\}$ as the inputs of a Long Short Term Memory (LSTM) [12] based on the utterances order appearing in the dialog history. Firstly, the final hidden state of LSTM is added to the internal state u_k.

$$u^{k+1} = u^k + o^k + h^k_{N-2}, \tag{4}$$

where h^k_{N-2} is the last LSTM hidden state at the hop k. In this way, our model can capture the global attention over memory cells o^k, and also the internal latent representation of dialog state h^k_{N-2}.

Secondly, motivated by [13], we use each hidden state of the corresponding time step to query the next memory cells separately. That is, the next hop query vector is not generic over all the memory cells $\{m_i\}$. Each cell has its unique query vector

$$q_i^{k+1} = u^{k+1} + h_i^k, \tag{5}$$

which is then sent to attention weights computation in (1),

$$p_i^{k+1} = Softmax((q_i^{k+1})^T m_i). \tag{6}$$

DQMemNN considers the previous hop memory cells as a sequence of query-changing triggers, which trigger the LSTM to generate more dynamically informed queries. Therefore, our model can effectively alleviate temporal problems by the dynamic query components.

3. RELATED WORK

Generally, there are two approaches in applying machine learning for task-oriented dialog systems. The first one mainly implements a pipelined module for language understanding, dialog state tracking, action policy, and language generation [14, 15, 16, 5]. In this approach, the dialog policy has been implemented through a probabilistic decision process or feed-forward neural networks trained with supervised learning. However, the models depend on the state features summarized from state tracker at each time step, which requires designed and explicit labelling.

The second approach is to train model directly on text transcripts of dialogs in an end-to-end fashion, which learns a distributed vector representation of the dialog state automatically [4, 8, 9, 13, 17]. RNN model, like LSTM, plays an important role due to its ability to create a latent representation, avoiding the need for artificial state labels. In each of these architectures, the output is produced by generating a sequence of tokens, or by ranking all possible actions. Similarly, End-to-End Memory Networks [6, 7] can be seen as a variant RNN models. MemNN uses a global memory, with shared read-write functions and global attention mechanism over memory representation.

Table 1. Statistics of bAbI dialog dataset. All the reported number has been taken from the original article [1].

Task	**1**	**2**	**3**	**4**	**5**
Avg. User turns	4	6.5	6.4	3.5	12.9
Avg. Sys turns	6	9.5	9.9	3.5	18.4
Avg. KB results	0	0	24	7	23.7
Avg. Sys words	6.3	6.2	7.2	5.7	6.5
Vocabulary	3747				
Train dialogs	1000				
Val dialogs	1000				
Test dialogs	1000 + 1000 OOV				

DQMemNN is closely related to memory network but differs in the memory access. The additional RNN structure added is able to capture the sequential dependencies. Furthermore, the combination of DQMemNN and RDL makes our framework for task-oriented dialog system easier to extend to new domains, and also it also can alleviate the OOV entity issue.

4. EXPERIMENTS

4.1. Dataset

To evaluate the performance of our model, we use the bAbI dialog dataset [1]. It has become a standard benchmark since it evaluates all the desirable features of an end-to-end task-oriented dialog systems. The dataset is divided into 5 different tasks, each of which has its own training set. Task 1-4 are issuing API call, refining API call, recommending options and providing additional information, respectively. Task 5 (full dialogs) is a union of task 1-4 and includes more conversational turns. There are two test sets for each task, one follows the same distribution as the training set and the other has OOV words from a different KB. Dataset statistics are reported in Table 1, and a toy example of task 1 is shown in Figure 1.

4.2. Model Details

All experiments used a $K = 3$ hops model with the adjacent weight sharing scheme, that is, $\mathbf{C}_k = \mathbf{A}_{k+1}$ for $k = 1, 2$. We embed $cand$ via the last stage embedding matrix $\mathbf{C}_K$, and also share parameters between $\mathbf{A}_1$ and $\mathbf{B}$ to simplify our setting. For sentence representation, we use position encoding BoW as in [6] to capture words order. Our models were trained using a learning rate 0.01, with anneals half every 25 epochs until 100 epochs were reached. The weights were initialized randomly from a Gaussian distribution with zero mean and $\sigma = 0.1$. All training uses a batch size of 16 (but the cost is not averaged over a batch), and gradients with a $l2$ norm greater than 40 were clipped. During training, all embedding matrices are jointly learned by minimizing a standard

Table 2. Per-response accuracy and per-dialog accuracy (in parentheses) on bAbI dialog dataset. Results are compared on plain text without match type feature in [1]. For those results not reported in the original paper we use hyphen character.

Task	QRN*	MemNN	GMemNN	HCN	DQMemNN	DQMemNN+RDL
T1	99.4 (-)	99.9 (99.6)	100 (100)	-	100 (100)	100 (100)
T2	99.5 (-)	100 (100)	100 (100)	-	100 (100)	100 (100)
T3	74.8 (-)	74.9 (2.0)	74.9 (3.0)	-	74.9 (2.0)	98.7 (90.8)
T4	57.2 (-)	59.5 (3.0)	57.2 (0)	-	57.2 (0)	100 (100)
T5	99.6 (-)	96.1 (49.4)	96.3 (52.5)	-	99.2 (88.7)	99.9 (98.3)
Test Avg.	86.1 (-)	86.1 (50.8)	85.7 (50.5)	-	86.3 (58.1)	**99.7** (**97.8**)
T1-OOV	83.1 (-)	72.3 (0)	82.4 (0)	-	82.5 (0)	100 (100)
T2-OOV	78.9 (-)	78.9 (0)	78.9 (0)	-	78.9 (0)	100 (100)
T3-OOV	75.2 (-)	74.4 (0)	75.3 (0)	-	74.9 (0)	98.7 (90.4)
T4-OOV	56.9 (-)	57.6 (0)	57.0 (0)	-	57.0 (0)	100 (100)
T5-OOV	67.8 (-)	65.5 (0)	66.7 (0)	100 (100)	72.0 (0)	99.4(91.6)
Test-OOV Avg.	72.4 (-)	69.7 (0)	72.1 (0)	-	71.8 (0)	**99.6**(**96.4**)

cross-entropy loss between $\hat{a}$ and the true label a. Training is performed using stochastic gradient descent.

4.3. Results

In Table 2 we report the results obtained in the bAbI dialog test sets (including the OOV). We compare our proposed models DQMemNN with and without RDL to original End-to-End Memory Networks [6], Gated MemNN [7], Query Reduction Networks (QRN) [13], and Hybrid Code Networks (HCN) [8]. In this dataset, HCN represents the state-of-the-art in Task 5 OOV with a perfect accuracy, but this model is not fully end-to-end, which includes domain-specific rules provided by a software developer. Note that the results we listed in Table 2 for QRN are different from the original paper, since based on their released code [1] we found that the per-response accuracy was not correctly computed, thus we simply modified the evaluation part and reported the results.

As we can see, DQMemNN improves the average accuracy among the standard tasks compared to other memory network models. Specifically, in full dialog task 5, DQMemNN outperforms MemNN by 3.1% in the per-response accuracy and by 39.3% in the per-dialog accuracy. This result shows that the dynamic query components in DQMemNN make memory network learn more complex dialog policy. Task 5 includes long conversational turns, and it requires stronger dialog state tracking ability. Similarly, another RNN method QRN also performs better in task 5, which further confirmed our hypothesis. However, the difference between DQMemNN and the others is not that promising in the OOV test set.

On the other hand, one can observe that DQMemNN with RDL outperforms all the other models by far. It improves the average test accuracy by 13.6.%, 13.6% and 14.0% in standard test set compared to QRN, MemNN and GMemNN, respectively; and an improvement of 27.0%, 29.9% and 27.5% in OOV test set. Note that our framework can achieve almost perfect per-response accuracy in Task5-OOV, which also confirm our initial assumption that using RDL strongly decrease the learning complexity. This strategy leads to an overall accuracy improvement, which is particularly useful when the network needs to learn how to work with abstract OOV entities.

[1] https://github.com/uwnlp/qrn

5. CONCLUSION

This paper has introduced an end-to-end framework for task-oriented dialog systems based on Dynamic Query Memory Network and a recorded delexicalization mechanism. DQMemNN is designed to overcome the major drawback of MemNN, namely no temporal dependencies during memory access. In addition, RDL is able to reduce the learning complexity and also alleviate OOV entity problems. The results show that DQMemNN outperforms other memory network models, especially in the task with longer dialog turns. In addition, the proposed framework achieves average 99.6 % per-response accuracy without hand-crafted rules and features in OOV test set. In future work, we would like to extend our model to a more domain-general setting and design a more efficient way to capture sequential dependencies.

6. REFERENCES

[1] Antoine Bordes and Jason Weston, "Learning end-to-end goal-oriented dialog," *ICLR*, vol. abs/1605.07683, 2017.

[2] Oliver Lemon, Kallirroi Georgila, James Henderson, and Matthew Stuttle, "An isu dialogue system exhibiting reinforcement learning of dialogue policies: generic slot-filling in the talk in-car system," in *Proceedings of the Eleventh Conference of the European Chapter of*

the Association for Computational Linguistics: Posters & Demonstrations. Association for Computational Linguistics, 2006, pp. 119–122.

[3] Zhuoran Wang and Oliver Lemon, "A simple and generic belief tracking mechanism for the dialog state tracking challenge: On the believability of observed information.," in *SIGDIAL Conference*, 2013, pp. 423–432.

[4] Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C Courville, and Joelle Pineau, "Building end-to-end dialogue systems using generative hierarchical neural network models.," in *AAAI*, 2016, pp. 3776–3784.

[5] Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Lina Maria Rojas-Barahona, Pei hao Su, Stefan Ultes, David Vandyke, and Steve J. Young, "A network-based end-to-end trainable task-oriented dialogue system," in *EACL*, 2017.

[6] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al., "End-to-end memory networks," in *Advances in neural information processing systems*, 2015, pp. 2440–2448.

[7] Julien Perez and Fei Liu, "Gated end-to-end memory networks," in *EACL*, 2017.

[8] Jason D. Williams, Kavosh Asadi, and Geoffrey Zweig, "Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning," in *ACL*, 2017.

[9] Tiancheng Zhao, Allen Lu, Kyusong Lee, and Maxine Eskenazi, "Generative encoder-decoder models for task-oriented spoken dialog systems with chatting capability," in *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*. August 2017, pp. 27–36, Association for Computational Linguistics.

[10] Mikael Henaff, Jason Weston, Arthur Szlam, Antoine Bordes, and Yann LeCun, "Tracking the world state with recurrent entity networks," *ICLR*, vol. abs/1612.03969, 2017.

[11] Andrea Madotto and Giuseppe Attardi, "Question dependent recurrent entity network for question answering," *arXiv preprint arXiv:1707.07922 (Accepted at NL4AI)*, 2017.

[12] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[13] Minjoon Seo, Sewon Min, Ali Farhadi, and Hannaneh Hajishirzi, "Query-reduction networks for question answering," *ICLR*, 2017.

[14] Jason D Williams and Steve Young, "Partially observable markov decision processes for spoken dialog systems," *Computer Speech & Language*, vol. 21, no. 2, pp. 393–422, 2007.

[15] Chiori Hori, Kiyonori Ohtake, Teruhisa Misu, Hideki Kashioka, and Satoshi Nakamura, "Statistical dialog management applied to wfst-based dialog systems," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*. IEEE, 2009, pp. 4793–4796.

[16] Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams, "Pomdp-based statistical spoken dialog systems: A review," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1160–1179, 2013.

[17] Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C Courville, and Yoshua Bengio, "A hierarchical latent variable encoder-decoder model for generating dialogues.," in *AAAI*, 2017, pp. 3295–3301.