An Improved End-to-End Memory Network for QA Tasks

**Aziguli Wulamu[1, 2], Zhenqi Sun[1, 2], Yonghong Xie[1, 2, *], Cong Xu[1, 2],
and Alan Yang[3]**

Abstract: At present, End-to-End trainable Memory Networks (MemN2N) has proven to be promising in many deep learning fields, especially on simple natural language-based reasoning question and answer (QA) tasks. However, when solving some subtasks such as basic induction, path finding or time reasoning tasks, it remains challenging because of limited ability to learn useful information between memory and query. In this paper, we propose a novel gated linear units (GLU) and local-attention based end-to-end memory networks (MemN2N-GL) motivated by the success of attention mechanism theory in the field of neural machine translation, it shows an improved possibility to develop the ability of capturing complex memory-query relations and works better on some subtasks. It is an improved end-to-end memory network for QA tasks. We demonstrate the effectiveness of these approaches on the 20 bAbI dataset which includes 20 challenging tasks, without the use of any domain knowledge. Our project is open source on github[4].

Keywords: QA system, memory network, local attention, gated linear unit.

1 Introduction

The QA problem has been around for a long time. As early as 1950, the British mathematician A.M. Turing in his paper put forward a method to determine whether the machine can think-Turing Test, which is seen as the blueprint for the QA system [Turing (1950)]. The first generation of intelligent QA system converts simple natural language questions into pre-set single or multiple keywords and queries the information in a domain-specific database to obtain answers. Its earliest appearance can be traced back to the early 1950s and 1960s when the computer was born. The representative systems include two well-known QA systems, baseball [Green Jr, Wolf, Chomsky et al. (1961)] and lunar [Woods and Kaplan (1977)]. They have a database in the background that holds various data the system can provide. When the user asks a question, the system converts the user's question into a SQL query statement, and queries the data from the database to the user.

[1] School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, 100083, China.

[2] Beijing Key Laboratory of Knowledge Engineering for Materials Science, Beijing, 100083, China.

[3] Amphenol Assemble Tech, Houston, TX 77070, US.

[4] https://github.com/zhenqicool/An-Improved-End-To-End-Memory-Network-For-QA-Tasks-in-PyTorch.

[*] Corresponding Author: Yonghong Xie. Email: xieyh@ustb.edu.cn.

CMC.doi:10.32604/cmc.2019.07722

Shrdlu was a highly successful QA program developed by Terry Winograd in the late 60s and early 70s [Winograd (1972)], it simulated the operation of a robot in a toy world (the "blocks world"). The reason for its success was to choose a specific domain and the physical rules were easily written as programs. The travel information consultation system GUS developed by Bobrow et al. in 1977 is another successful QA system [Bobrow, Kaplan, Norman et al. (1977)]. In the 1990s, with the development of the Internet, a second generation question and answer system emerged. It extracts answers from large-scale text or web-based libraries based on information retrieval techniques and shallow NLP techniques [Srihari and Li (2000); Voorhees (1999); Zheng (2002)]. A representative example is Start (1993), which is the world's first web-based question answering system, was developed by the MIT Artificial Intelligence Lab [Katz (1997)]. In 1999, the TREC (Text REtrieval Conference) began the evaluation of the question and answer system. In October 2000, ACL(the Association for Computational Linguistics) used the open domain question and answer system as a topic, which promoted the rapid development of the question and answer system. With the rise of web 2.0 technology, the third generation question answering system has developed [Tapeh and Rahgozar (2008)]. It is characterized by high quality knowledge resources and deep NLP technology. Up to now, in addition to the "Cortana" of Microsoft [Young (2019)], the "Dumi" of Baidu [Zhu, Huang, Chen et al. (2018)] and the "Siri" of Apple [Hoy (2018)], many companies and research groups have also made breakthroughs in this field [Becker and Troendle (2018); Zhou, Gao, Li et al. (2018)].

Table 1: Samples of three types of tasks

1 supporting fact	yes/no questions	3 supporting facts
Mary journeyed to the office.	Daniel went to the hallway.	Mary left the milk.
Daniel travelled to the office.	Mary journeyed to the office.	Mary left the milk.
Daniel moved to the garden.	John journeyed to the garden.	John travelled to the hallway.
Mary went to the kitchen.	John dropped the football.	Mary went back to the hallway.
John went to the bedroom.	John took the football there.	Mary went to the garden.
Daniel went back to the office.	John went back to the bathroom	Daniel picked up the football.
	Mary moved to the bedroom.	Daniel dropped the football.
	Mary moved to the bedroom.	Mary grabbed the milk.
	Sandra took the apple there	Mary put down the milk.
	John discarded the apple.	Mary picked up the milk.
	John got the apple there.	Daniel got the football.
Q: Where is Mary?	**Q: Is John in the garden?**	**Q: Where was the milk before the hallway?**
A: office.	**A: yes.**	**A: garden.**

In such a situation, end-to-end learning framework have shown promising performance because of their applicability in the real environment and efficiency in model updating [Shi and Yu (2018); Madotto, Wu and Fung (2018); Li, Wang, Sun et al. (2018); Liu, Tur, Hakkani-Tur et al. (2018)]. In end-to-end dialog systems, End-to-end memory network (MemN2N) and its variants have always been hot topics of research [Perez and Liu (2018); Ganhotra (2018)], in light of the powerful ability to describe long term dependencies [Huang, Qi, Huang et al. (2017)] and the flexibility in the implementation process.

Although MemN2N has achieved good performance on the dialog bAbI tasks, where the memory components effectively work as representation of the dialog context and play a good role in inference. There are still many tasks not very satisfactory in the bAbI [Shi and Yu (2018)]. In order to find out the reasons for this, we have made a careful comparison. We found that tasks achieved good performance on the dialog bAbI tasks (such as Task 3 supporting factsin Tab. 1) have in common that there are many more contextual sentences than those perform well (such as Task yes/no questions in Tab. 1). And when calculating the relevance of the memory and the query, MemN2N attend to all sentences on the memory side for query [Sukhbaatar, Szlam, Weston et al. (2015)], which is expensive and can potentially render it impractical. Inspired by the field of machine translation [Minh Thang and Hieu Pham (2015)], we introduce the local-attention mechanism when calculating the correlation between memory and query. We don't consider all the information of memory, but a subset of sentences which are more relevant to the query. At the same time, inspired by the gated convolutional network(GCN) [Dauphin, Fan, Auli et al. (2017)], we investigate the idea of the gated linear units (GLU) into MemN2N to update the intermediate state between layers. The purpose of these two improvements is the same, that is to appropriately reduce the complexity of the model, let the model pay more attention to useful information when training.

We have compared our two improved methods to MemN2N in the bAbI tasks, analyzed their number of successful tasks and error rates in different tasks. We also analyzed from the perspective of training speed and visual weight. Experimentally, we demonstrate that both of our approaches are effective.

In the following sections, we first introduce the application of MemN2N model and the innovation of our MemN2N-GL model in the second section; in the third section, we introduce the implementation methods of our model in detail, including local-attention matching and GLU mapping; then, we show our experimental results in the fourth section, and make a comparative analysis with baseline; finally, in the fifth section, we show our conclusion and planning for future work.

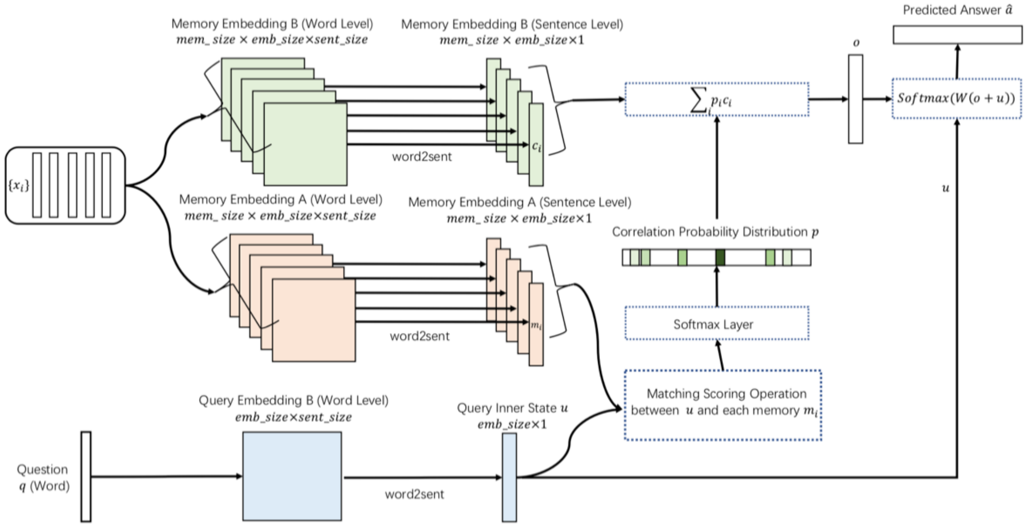
Memory Embedding B (Word Level)
$mem\_size \times emb\_size \times sent\_size$
Memory Embedding B (Sentence Level)
$mem\_size \times emb\_size \times 1$
Predicted Answer $\hat{a}$
o
$\sum_i p_i c_i$
$Softmax(W(o+u))$
c_i
word2sent
$\{x_i\}$
u
Memory Embedding A (Word Level)
$mem\_size \times emb\_size \times sent\_size$
Memory Embedding A (Sentence Level)
$mem\_size \times emb\_size \times 1$
Correlation Probability Distribution p
Softmax Layer
m_i
word2sent
Matching Scoring Operation between u and each memory m_i
Query Embedding B (Word Level)
$emb\_size \times sent\_size$
Query Inner State u
$emb\_size \times 1$
Question q (Word)
word2sent

Figure 1: A single layer version of MemN2N

2 Model

The MemN2N architecture, introduced by Sukhbaatar et al. [Sukhbaatar, Szlam, Weston et al. (2015)], is a hot research method in the field of current QA system [Ganhotra and Polymenakos (2018); Perez and Liu (2018)]. It is a form of Memory Network (MemNN) [Weston, Chopra and Bordes (2014)], but unlike the model in that work, it is trained end-to-end that makes it easier to apply in practical situations [Sukhbaatar, Szlam, Weston et al. (2015)]. Compared with traditional MemNN, MemN2N gets less supervision during training, which means it will reduce some complexity and may not be able to fully capture the context information.

Because of the good characteristics of MemN2N, it has been used in a wide range of tasks in recent years. Boyuan Pan et al. introduced a novel neural network architecture called Multi-layer Embedding with Memory Network (MemNN) for machine reading task, where a memory network of full-orientation matching of the query and passage to catch more pivotal information [Pan, Li, Zhao et al. (2017)]. M Ghazvininejad et al. presented a novel, fully data-driven, and knowledge-grounded neural conversation model aimed at producing more contentful responses [Ghazvininejad, Brockett, Chang et al. (2018)]. In the field of computer vision, Wu et al. proposed a long-term feature bank, which extracts supportive information over the entire span of a video to augment state-of-the-art video models [Wu, Feichtenhofer, Fan et al. (2018)]. Although MemN2N has been widely used in many fields and has achieved good results, it may not scale well to the case where a larger memory is required [Sukhbaatar, Szlam, Weston et al. (2015)].

In this paper, inspired by the attention mechanism and its many deformations in the field of deep learning [Shen, Zhou, Long et al. (2018); Zhang, Goodfellow, Metaxas et al. (2018); Shen, He and Zhang (2018)], we propose an improvement point to introduce local attention mechanism into MemN2N to improve the model effect. Compared to the use of global matching between u and each memory m_i in MemN2N (Eq. (4)), local attention mechanism pays more attention to the local information related to the question state u in the memory. We also consider to optimize the updating of hidden state u between layers of MemN2N (Fig. 1). The original method uses a linear mapping H (Eq. (10)) while we draw on the experience of GLU (Gated Linear Unit) proposed by Dauphin et al. [Dauphin, Fan, Auli et al. (2017)]. We compare the improved model based on these two points with MemN2N in the same data sets. As a result, our model performs better in more complex QA tasks.

3 Methods

In this section, we introduce our proposed model MemN2N-GL. Our model aims to extract more useful interactions between memory and query to improve the accuracy of MemN2N. Similar to MemN2N, our MemN2N-GL consists of three main components: input memory representation, out memory representation and final answer prediction.

In the part of input memory representation, an input set $x_1,\ldots,xi$ are converted into memory vectors $\{m_i\}$ and $\{c_i\}$ of dimension d in a continuous space, using embedding matrixes A and C, both of them are $d \times V$, where d is embedding size, V is vocabulary size. Similarly, the query q is also embedded (by matrix B) to an internal state u. We use position encoding (PE) [Sukhbaatar, Szlam, Weston et al. (2015)] to convert word vectors as sentence vectors. This takes the form:

$$m_i = \sum_j l_j \cdot Ax_{ij} \tag{1}$$

where l_j is column vector with the structure:

$$l_{kj} = (1 - j/J) - (k/d)(1 - 2j/J) \tag{2}$$

J is the number of words in the sentence, and d is the dimension of the embedding. In this way, the position information of the words are taken into account when generating the sentence vector. Questions, memory inputs and memory outputs also use the same representation. In order to enable memory to have context temporal information, we also modify the memory vector by:

$$m_i = \sum_j l_j \cdot Ax_{ij} + T_A(i) \tag{3}$$

where $T_A(i)$ is the ith row of a special matrix T_A that encodes temporal information, and T_A is learned during training.

Then in the embedding space, MemN2N calculate the relevance score between u and each memory m_i by means of matching in dot form [Minh-Thang Luong (2015)] followed by a softmax:

$$p_i = Softmax(u^T m_i) \tag{4}$$

where

$$Softmax(i) = \frac{\exp(i)}{\sum_{j\in[1,n]} \exp(j)} \tag{5}$$

After applying softmax function, each component of the matrix $u^T m_i$ will be in the interval (0,1), and the components will add up to 1, so that they can be interpreted as probabilities. Furthermore, the larger input components will correspond to larger probabilities.

By contrast, we develop local-attention mechanism to calculate the correlation and filtering out irrelevant information between u and $\{m_i\}$, compared with the attention mechanism used in MemN2N, our model does not focus on the relevance of the global memory and query, but focuses on the local memory associated with the query.

3.1 Local-attention matching

As mentioned before, local-attention matching chooses to focus only on a small subset of the memory, which is more relevant to query q (Fig. 2). Concretely, the model first generates an aligned position p_u for the query q in the memory embedding A:

$$p_u = S \cdot \delta(v_p^T \tanh(W_a q)) \tag{6}$$

where v_p, W_a are the model parameters which will be learned to predict positions. S is the memory size, δ is activation function and $p_u \in [0, S]$.

Then the relevance score between u and each memory m_i is defined as:

$$p_i = p_i \cdot \exp(-\frac{(s-p_u)^2}{2\delta^2}) \tag{7}$$

where p_i is the original score (Eq. (4)), $\delta = \frac{D}{2}$ is the standard deviation and D is the window size of subset memory. Finally, we use the new relevance score p_i to calculate the out memory representation in Fig. 2. The response o from output memory is a sum of memory vectors $\{c_i\}$, weighted by the input probability vector:

$$o = \sum_i p_i c_i \tag{8}$$

Finally in final answer prediction, the predicted answer distribution $\hat{a}$ is produced by the sum of the output vector o and the input embedding u which then passed through a final weight matrix W (of size $V \times d$) and a softmax:

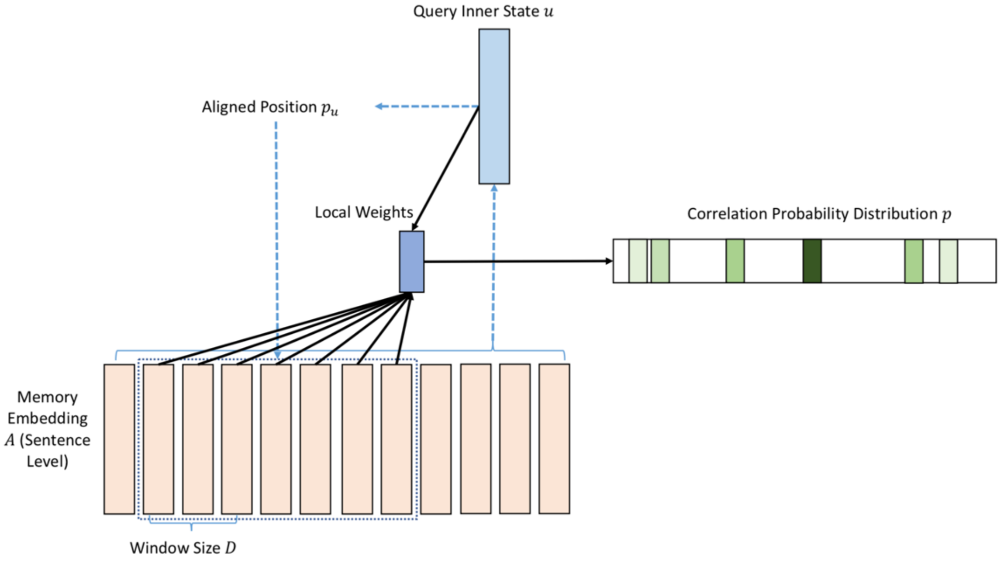
Query Inner State u
Aligned Position p_u
Local Weights
Correlation Probability Distribution p
Memory Embedding A (Sentence Level)
Window Size D

Figure 2: Local-Attention Matching

$$\hat{a} = Softmax(W(u + o)) \tag{9}$$

The above is for single layer structure (Fig. 2), for many different types of difficult tasks, the model can be extended to multi-layer memory structure (Fig. 1), where each memory layer is named a hop and in MemN2N, the $(K+1)^{th}$ hop's state u^{K+1} is calculated by :

$$u^{K+1} = Hu^K + o^K \tag{10}$$

By contrast, we utilize gated linear units (GLU) in our MemN2N-GL. Compared to linear mapping H or frequently used nonlinear mapping functions, GLU effectively reducing the gradient dispersion, but also retaining the ability of nonlinearity. Proved by experiment, it is better suit for MemN2N.

3.2 Gated linear units mapping

In our MemN2N-GL, we use the layer-wise [Sukhbaatar, Szlam, Weston et al. (2015)] form (where the memory embeddings are the same across different layers, i.e., $A_1 \ldots = A_K$ and $C_1 \ldots = C_K$.) to expand the model from a single-layer to a multi-layer structure. In this case, we utilize GLU mapping to the update of u between layers:

$$u^{K+1} = (Wu^K + b) \otimes \delta(Vu^K + c) + o^K \tag{11}$$

where $W, V \in \mathbb{R}^{m \times m}$, m is embedding size, $b, c \in \mathbb{R}^{m \times 1}$, W, V, b, c are learned parameters, $o^K \in \mathbb{R}^{m \times 1}$ is the output of the Kth layer (Fig. 1).

Then, the predicted answer distribution $\hat{a}$ is the combination of the input and the output of the top memory layer:

$$\begin{aligned} \hat{a} &= Softmax(Wu^{K+1}) \\ &= Softmax(W(u^K + o^K)) \end{aligned} \tag{12}$$

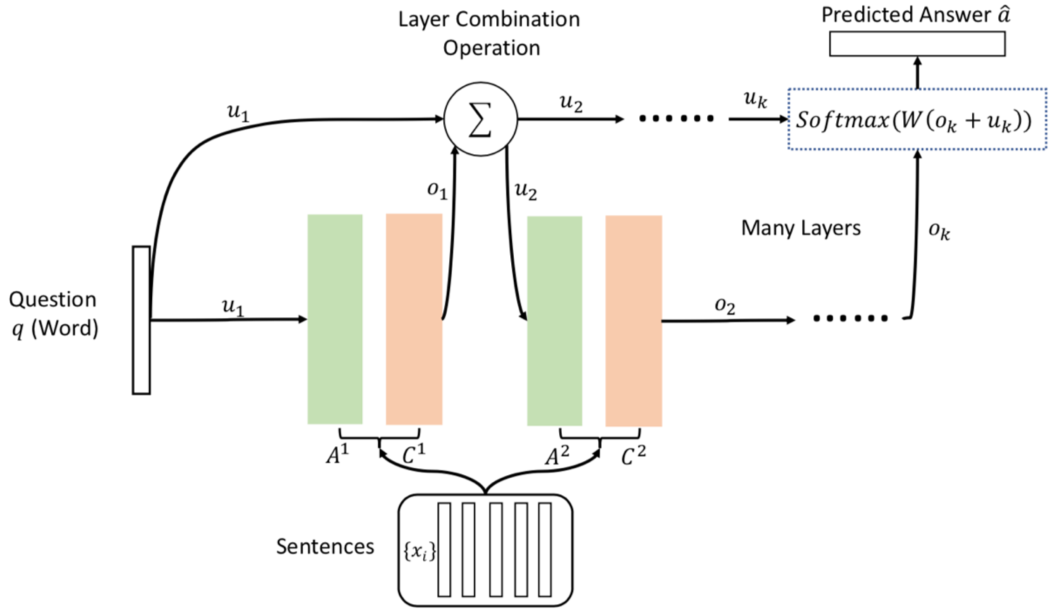

Figure 3: A k layers version of MemN2N

Finally, at the top of the network, we adopt the original approach combining the input u_k and the output o_k of the top memory layer:

$$\hat{a} = Softmax(W(o_k + u_k)) \tag{13}$$

where W is the parameter learned during training.

Compared with MemN2N, our model performs better in complex QA problems, which can be confirmed in the experimental results in the next section. We believe that the local attention mechanism removes redundant memory when calculating the correlation between memory and query, so the weight vector obtained is "purer" and contains more useful information. Besides, compared with linear mapping, GLU mapping has the ability of nonlinearity, which makes the model has stronger learning ability in the update of u between layers.

4 Experiments and results

We perform experiments on goal-oriented dialog datasets Dialog bAbI [Weston, Bordes, Chopra et al. (2015)], which contains 20 subtasks. Each of subtasks consists of three parts: the context statements of the problem, the question, and correct answer. There are samples of three of the tasks in Tab. 1. For each question, only certain subsets of the statement contain the information needed for the answer, while other statements are basically unrelated interferers. And the difficulty of various subtasks is different, which is reflected in the increase of interference statements. During training, we choose to use 10K dataset, our goal is to improve the ability of the model to answer questions correctly based on context.

4.1 Training details

We perform our experiments with the following hyper-parameter values: embedding dimension $embed\_size = 128$, learning rate $\lambda = 0.01$, size of each batch $batch\_size = 32$, number of layers $K = 3$, capacity of memory $memory\_size = 50$ and max gradient norm to clip $max\_clip = 40.0$. We also used some skills during the training, for example, the learning rate of our model automatically adjusts with the change of loss. If the loss value does not decrease but increases between adjacent training epochs, the learning rate will be reduced to $2/3$ of the current value. The condition of training termination is that the loss value is less than a certain threshold (the experiment was 0.001), or the number of training epochs reaches the upper limit. In the course of training, the training time varies with the difficulty of different subtasks, but all of them are within one day.

Table 2: Test error rates (%) on the 20 QA tasks

QA tasks	Base Line		My Model	
	MemN2N	MemN2N (Local-Attention)	MemN2N (GLU)	MemN2N-GL
1: 1 supporting fact	**0.0**	**0.0**	**0.0**	**0.0**
2: 2 supporting facts	78.6	69.9	75.8	**66.2**
3: 3 supporting facts	71.7	74.8	76.6	**71.5**
4:2 argument relations	**0.0**	**0.0**	**0.0**	**0.0**

5: 3 argument relation	9.5	11.2	3.2	**0.9**
6: yes/no questions	50.0	50.0	**25.2**	48.2
7: counting	50.3	11.5	16.1	**10.9**
8: lists/sets	8.7	7.3	6.0	**5.6**
9: simple negation	12.3	35.2	13.1	**4.4**
10:indefinite knowledge	11.3	**2.6**	3.9	13.9
11: basic coreference	15.9	18.0	40.9	**9.0**
12: conjunction	**0.0**	**0.0**	2.8	**0.0**
13: compound coreference	46.1	21.3	18.8	**1.4**
14: time reasoning	6.9	5.8	**4.4**	10.3
15: basic deduction	43.7	75.8	2.4	**0.0**
16: basic induction	53.3	**52.5**	57.8	53.1
17: positional reasoning	46.4	50.8	48.6	**46.2**
18: size reasoning	9.7	**7.4**	13.6	12.9
19: path finding	89.1	89.3	**14.5**	24.7
20: agent's motivation	0.6	**0.0**	**0.0**	1.7
Mean error (%)	30.2	29.2	21.2	**19.0**
Successful tasks (err < 5%)	4	5	**8**	**8**

Table 3: The visualization weights of layers

Compound coreference task	MemN2N			MemN2N-GL		
	layer1	layer2	layer3	layer1	layer2	layer3
Fred gave the apple to Bill	0.0000	4.6e-43	0.0000	0.3679	0.1353	0.3679
Fred grabbed the apple there	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Fred moved to the garden	0.0000	1.7e-26	0.0000	0.0000	0.0000	0.0000
Fred journeyed to the hallway	6.3e-37	1.7e-19	5.9e-39	0.0000	0.0000	0.0000
Fred journeyed to the garden	9.8e-40	4.5e-24	2.7e-37	0.0000	0.0000	0.0000
Mary journeyed to the bedroom	5.7e-40	2.1e-32	2.0e-27	0.0000	0.0000	0.0000
Jeff took the football there	0.0000	0.0000	2.5e-39	0.0000	0.0000	0.0000
Bill moved to the garden	0.0000	3.9e-39	2.8e-45	0.0000	0.0000	0.0000
Q: Who did Fred give the apple to?	wrong answer: Jeff			correct answer: **Bill**		

4.2 Results and analysis

Our baseline is MemN2N. We try three different combinations of improvements and compare their performance in different subtasks (as shown in Tab. 2). MemN2N (Local-Attention) and MemN2N (GLU) indicate that the model only adds the local attention mechanism and the GLU mechanism respectively, MemN2N-GL means that both improvements exist simultaneously. The number in the table represents the error rates of each sub-task, and the bold number represents the result of the model that performs best in the same subtask. In the last two rows of the table, we counted the mean error rates of all models and the number of successful tasks (subtasks with error rates less than 5).

In terms of results, MemN2N-GL achieves the best results both in mean error rates and in the number of successful tasks. Compared with MemN2N, the mean error rates is reduced by 37.09%, and the number of successful tasks doubled from four to eight. MemN2N (Local-Attention) and MemN2N (GLU) have their own advantages and disadvantages, but both of their effect are better than MemN2N.

4.3 Related tasks

In addition to comparing the results of different tasks with each model in Tab. 2, we use a specific example to quantitatively analyze the result by the visualization weights of layers. As shown in Tab. 3, the most relevant memory sentence to the query “Who did Fred give the apple to ?” is the first memory sentence: “Fred gave the apple to Bill”. After training, MemN2N does not focus on the memory sentences of greater relevance, which led to the wrong answer as a result. While our model pays close attention to contextual information that is highly relevant to query, which can be reflected in the size of the correlation weight at each layers. The darker the color, the greater the weight.

5 Conclusion and future work

In this paper we proposed two improvements based on MemN2N model for QA problem and perform empirical evaluation on dialog datasets bAbI. The experimental results show that our improved model has a greater performance than the original model, which strongly confirms our conjecture that the model should pay more attention to the useful information when training. In the future, we are prepared to further improve the ability of the model to handle complex tasks. At the same time, we are going to test our model on more datasets. We also intend to combine our model with recent research results Bert (Bidirectional Encoder Representations from Transformers) and use our model as a downstream part to see if it will achieve better results.

Acknowledgement: This work is supported by the National Key Research, Development Program of China under Grant 2017YFB1002304 and the National Natural Science Foundation of China (No. 61672178)

References

Becker, T.; Troendle, T. (2018): Alexa, can you get me an insurance? a structured approach to the hyped technology of voice-based assistants. *The InsurTech Book: The*

Insurance Technology Handbook for Investors, Entrepreneurs and FinTech Visionaries, vol. 57, no. 10, pp. 250-253.

Bobrow, D. G.; Kaplan, R. M.; Norman, D. A.; Kay, M.; Thompson, H. et al. (1977): Gus, a frame-driven dialog system. *Artificial Intelligence*, vol. 8, no. 2, pp. 155-173.

Dauphin, Y. N.; Fan, A.; Auli, M.; Grangier, D. (2017): Language modeling with gated convolutional networks. *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, pp. 933-941.

Fan, A.; Lewis, M.; Dauphin, Y. (2018): Hierarchical neural story generation. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 889-898.

Ganhotra, J.; Polymenakos, L. (2018): Knowledge-based end-to-end memory networks. https://www.researchgate.net/publication/324717719_Knowledge-based_end-to-end_memory_networks.

Ghazvininejad, M.; Brockett, C.; Chang, M. W.; Dolan, B.; Gao, J. et al. (2018): A knowledge-grounded neural conversation model. *Thirty-Second AAAI Conference on Artificial Intelligence*, pp. 5110-5117.

Green Jr, B. F.; Wolf, A. K.; Chomsky, C.; Laughery, K. (1961): Baseball: an automatic question-answerer. *IRE-AIEE-ACM Computer Conference*, pp. 219-224.

Hou, Y.; Kong, Q.; Wang, J.; Li, S. (2018): Polyphonic audio tagging with sequentially labelled data using CRNN with learnable gated linear units. arXiv:1811.07072v1.

Hoy, M. B. (2018): Alexa, siri, cortana, and more: an introduction to voice assistants. *Medical Reference Services Quarterly*, vol. 37, no. 1, pp. 81-88.

Huang, H.; Qi, Z.; Huang, X.; Huang, H.; Qi, Z. et al. (2017): Mention recommendation for twitter with end-to-end memory network. *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pp. 1872-1878.

Katz, B. (1997): From sentence processing to information access on the world wide web. *AAAI Spring Symposium on Natural Language Processing for the World Wide Web*, vol. 1, pp. 997.

Li, X.; Wang, Y.; Sun, S.; Panda, S.; Liu, J. et al. (2018): Microsoft dialogue challenge: Building end-to-end task-completion dialogue systems. arXiv:1807.11125v2.

Liu, B.; Tur, G.; Hakkani Tur, D.; Shah, P.; Heck, L. (2018): Dialogue learning with human teaching and feedback in end-to-end trainable task-oriented dialogue systems. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, pp. 2060-2069.

Madotto, A.; Wu, C. S.; Fung, P. (2018): Mem2seq: effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 1468-1478.

Minh Thang, L.; Hieu Pham, C. D. M. (2015): Effective approaches to attention-based neural machine translation. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, vol. 1, pp. 1412-1421.

Mirsamadi, S.; Barsoum, E.; Zhang, C. (2017): Automatic speech emotion recognition using recurrent neural networks with local attention. *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2227-2231.

Pan, B.; Li, H.; Zhao, Z.; Cao, B.; Cai, D. et al. (2017): Memen: multi-layer embedding with memory networks for machine comprehension. arXiv:1707.09098v1.

Perez, J.; Liu, F. (2017): Gated end-to-end memory networks. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, vol. 1, pp. 1-10.

Seo, S.; Huang, J.; Yang, H.; Liu, Y. (2017): Interpretable convolutional neural networks with dual local and global attention for review rating prediction. *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pp. 297-305.

Shen, T.; Zhou, T.; Long, G.; Jiang, J.; Pan, S. et al. (2018): Disan: directional self-attention network for RNN/CNN-free language understanding. *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pp. 5446-5455.

Shen, Y. H.; He, K. X.; Zhang, W. Q. (2018): Sam-GCNN: a gated convolutional neural network with segment-level attention mechanism for home activity monitoring. *IEEE International Symposium on Signal Processing and Information Technology*, pp. 679-684.

Shi, W.; Yu, Z. (2018): Sentiment adaptive end-to-end dialog systems. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 1509-1519.

Srihari, R.; Li, W. (2000): A question answering system supported by information extraction. *6th Applied Natural Language Processing Conference*, pp. 166-172.

Sukhbaatar, S.; Szlam, A.; Weston, J.; Fergus, R. (2015): End-to-end memory networks. *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems*, pp. 2440-2448.

Tapeh, A. G.; Rahgozar, M. (2008): A knowledge-based question answering system for b2c ecommerce. *Knowledge-Based Systems*, vol. 21, no. 8, pp. 946-950.

Tjandra, A.; Sakti, S.; Nakamura, S. (2017): Local monotonic attention mechanism for end-to-end speech recognition. arXiv:1705.08091v2.

Turing, A. M. (1950): Computing machinery and intelligence. *Mind*, vol. 59, pp. 433-64.

Voorhees, E. M. (1999): The trec-8 question answering track report. *Proceedings of the Eighth Text REtrieval Conference*, vol. 99, pp. 77-82.

Weston, J.; Bordes, A.; Chopra, S.; Rush, A. M.; van Merriënboer, B. et al. (2015): Towards ai-complete question answering: a set of prerequisite toy tasks. arXiv:1502.05698v10.

Weston, J.; Chopra, S.; Bordes, A. (2014): Memory networks. arXiv:1410.3916v11.

Winograd, T. (1972): SHRDLU: a system for dialog. *Ill and Diagrams Includes Bibliography*, vol. 2, pp. 20-48.

Woods, W. A.; Kaplan, R. (1977): Lunar rocks in natural English: explorations in natural language question answering. *Linguistic Structures Processing*, vol. 5, pp. 521-569.

Wu, C. Y.; Feichtenhofer, C.; Fan, H.; He, K.; Krähenbühl, P. et al. (2018): Long-term feature banks for detailed video understanding. arXiv:1812.05038v2.

Xu, Y.; Kong, Q.; Wang, W.; Plumbley, M. D. (2018): Large-scale weakly supervised audio classification using gated convolutional neural network. *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 121-125.

Young, L. (2019): 'I'm a cloud of infinitesimal data computation' when machines talk back: an interview with deborah harrison, one of the personality designers of Microsoft's cortana AI. *Architectural Design*, vol. 89, no. 1, pp. 112-117.

Zhang, H.; Goodfellow, I.; Metaxas, D.; Odena, A. (2018): Self-attention generative adversarial networks. arXiv:1805.08318.

Zheng, Z. (2002): Answerbus question answering system. *Proceedings of the Second International Conference on Human Language Technology Research*, pp. 399-404.

Zhou, L.; Gao, J.; Li, D.; Shum, H. Y. (2018): The design and implementation of XiaoIce, an empathetic social chatbot. arXiv:1812.08989.

Zhu, J.; Huang, T.; Chen, W.; Gao, W. (2018): The future of artificial intelligence in china. *Communications of the ACM*, vol. 61, no. 11, pp. 44-45.