Enhanced Question Understanding with Dynamic Memory Networks for Textual Question Answering

Chunyi Yue[a,b], Hanqiang Cao[a,*], Kun Xiong[c], Anqi Cui[c], Haocheng Qin[c], Ming Li[b]

[a] *School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, 430074, China*
[b] *David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, N2L 3G1, Canada*
[c] *RSVP Technologies Inc, Suite 19, 279 Weber St N, Waterloo, N2J 3H8, Canada*

Abstract

Memory networks show promising context understanding and reasoning capabilities in Textual Question Answering (Textual QA). We improve the previous dynamic memory networks to do Textual QA by processing inputs to simultaneously extract global and hierarchical salient features. We then use them to construct multiple feature sets at each reasoning step. Experiments were conducted on a public Textual Question Answering dataset (Facebook bAbI dataset) in two ways: with and without supervision from labels of supporting facts. Compared to previous works such as Dynamic Memory Networks, our models show better accuracy and stability.

Keywords: `dynamic memory networks`, `Attention based GRU`, `Textual Question Answering`

1. Introduction

Automated reasoning is a field of artificial intelligence (AI). It connects with mathematical logic and computer science. Given some facts, the machine needs to conduct inferences and then make judgements from these facts. In Natural

*Corresponding author. Phone:+86-027-87544820.
Email addresses: `d201377468@hust.edu.cn` (Chunyi Yue), `caohq@hust.edu.cn` (Hanqiang Cao), `kun@rsvptech.ca` (Kun Xiong), `caq@rsvptech.ca` (Anqi Cui), `qhc@rsvptech.ca` (Haocheng Qin), `mli@uwaterloo.ca` (Ming Li)

Language Processing (NLP), the task of Textual Question Answering (QA) can be seen as a type of reasoning tasks: Given a question, the machine provides an answer (judgement) based on a (miniature) knowledge base (facts) by analyzing the question, finding proper entities and attributes, and then retrieving the answer (inference steps). A sample of Textual QA is give in Figure 1. The well-known intelligent system IBM Watson builds its knowledge base from many different sources, from encyclopedia to the Internet, from structured infoboxes to unstructured texts (Fan et al., 2012). However, Textual QA has its difficulties: The facts are finite, simple sentences involving several objects (entities). To answer a question, the machine must infer from the single source of limited facts precisely and recognize the entities and relations accurately. Though challenging, AI researchers have built inference engines as components in expert systems (Jackson, 1998), to deduct new knowledge from existing knowledge bases. Typically the inference engines work with logics represented as IF-THEN rules, constructed from explicit variables, predicates and quantifiers. However, for natural language understanding, parsing the sentence may be difficult; the noises introduced may collapse the fragile logical system.

Facts	*supporting*	+	Question	→	Answer
Lily is a swan.					
Bernhard is a lion.	*Yes* 2				
Greg is a swan.					
Bernhard is white.	*Yes* 3				
Brian is a lion.	*Yes* 1	+	What color is Brian?	→	White
Lily is gray.					
Julius is a rhino.					
Julius is gray.					
Greg is gray.					

Figure 1: A sample from Facebook bAbI dataset.

The recent success of deep neural networks has brought a new solution to

this traditional task. Firstly proved by some image processing tasks, the neural networks have shown great potential of capturing connections between the observed elements, i.e., pixels or words (Vinyals et al., 2015; Xu et al., 2015; Yang et al., 2015; Antol et al., 2015). In NLP, the structures of convolutional neural networks (CNN) (Hu et al., 2014) and recurrent neural networks (RNN) (Sutskever et al., 2014) map the words to higher dimensions while keeping tracks of their contexts, hence are effective in many classification tasks (Kim, 2014; Lai et al., 2015) and sequential tasks (e.g.machine translation) (Kalchbrenner & Blunsom, 2013; Cho et al., 2014; Dzmitry Bahdanau, 2015; Meng et al., 2016). In addition, Memory Networks (Weston et al., 2015a,b) and Neural Turing Machines (Graves et al., 2014) introduce external memory units and flexible information storage mechanisms.

The paper is organized as follows. After reviewing related work in Section 2, we present our basic model in Section 3. Compared to other dynamic memory networks (DMNs) (Kumar et al., 2016; Xiong et al., 2016), our basic model has a subtly different internal structure and Attention based GRU (AttenGRU) (Kumar et al., 2016; Xiong et al., 2016) mechanism. In Section 4, we present our improved model - EnDMN. In Section 5, we show and analyze experimental results. In Section 6, we summarize the main contributions of this work and propose some future research.

2. Related work

Open-domain Question Answering (openQA) is a classical QA task, which requires an intelligent system to directly output a precise answer in natural language after receiving a question. For example, when a user inputs the question *"what is the largest inland lake in China?"*, the system is expected to output an answer *"Qinghai Lake"* rather than a list of ranked snippets and links. Recently, an increasing number of knowledge bases (e.g., Freebase, YAGO, and Google Knowledge) and corpus bases (e.g., blogs and forums) have become accessible. Combining with other techniques, some new progress in openQA has

been obtained (Sun et al., 2015; Bhati & Prasad, 2016).

Community Question Answering (CQA) is another hot issue in the field of QA. Many online CQA platforms (e.g., Quora and Stack Overflow) have become popular, where users can share knowledge in an interactive way. There is an obvious advantage to CQA - it allows users to obtain expected knowledge from other users in a variety of ways. Users can post their questions or answers, and comment or vote on contents posted by other users in the community. Namely, a user can be a questioner, an answerer, or a reviewer. Recently, many improved CQA systems have been proposed (Chang & Pal, 2013; Sahu et al., 2016a,b).

Question classification (QC) is a key part of traditional search engines and QA systems. QC can determine the entity of an answer and the pattern of an answer beforehand, which reduces the search scope and promotes search accuracy for the following information retrieval and answer selection. For instance, given questions *"who was the first man to win the Nobel Prize in Literature?"* and *"what is a violin?"*, the answer to the first question is supposed to be a name, and the answer to the second question should have a particular pattern like *"A violin is..."* or *"The violin is..."*. Some hybrid approaches to improve the performance of QC can be found in (Loni, 2011).

As described above, there is a significant difference between Textual QA and other QA tasks. In Textual QA, the question is always closely related to a miniature knowledge base (facts) about a particular scene. In this work, we focus on Textual QA. The main contributions of our work are threefold. First, we introduce global and hierarchical salient features of inputs (a question and a series of facts). Other models only use one type of features. Second, we propose using a modified network to extract the hierarchical salient features of a question to further improve the overall performance of our model. Third, we find a method to utilize these features to control the extraction of the information at each reasoning step. A main difference between our model and the closest related approaches is shown in Table 1.

Table 1: A main difference between our model and the closest related models.

EnDMN	DMNs
Global feature set **and** salient feature set	Global feature set **or** salient feature set

3. Basic framework and approach

As mentioned in Section 1, logical rules and the chaining mechanism are the traditional methods in building expert systems and solving logical problems, as well as the succeeding work of semantic networks with ontology. These strategies involve manual organization and labeling which are costly and time-consuming, hence are unsuitable to make use of the big amount of data.

The advance of deep learning revolution has presented new hope. With the development of memory networks and their attention mechanisms, some logical reasoning tasks have become popular and practicable recently. Researchers do not have to build the knowledge base (ontology) themselves; but instead can solve Textual QA tasks with end-to-end neural networks such as End-To-End Memory Network (E2E) (Sukhbaatar & Szlam, 2015), Dynamic Memory Networks (DMN) (Kumar et al., 2016), Dynamic Memory Networks for Visual and Textual Question Answering (DMN+) (Xiong et al., 2016), Neural Reasoner (NR) (Peng et al., 2015) and so on (Yu et al., 2015; Andreas et al., 2016).

3.1. DMN

A DMN is a type of end-to-end neural networks. It is usually composed of four modules: an input module, a question module, an episodic module, and an answer module. There are various networks to choose from for each module in a DMN. In this paper, our basic model is assembled as below :

Input module: The input module is seen as sentence readers to process facts. Different encoding methods, including long-short term memory (LSTM) (Hochreiter & Schmidhuber, 1997), gated recurrent units (GRUs) (Kumar et al., 2016; Xiong et al., 2016), and position encoding (PE) (Sukhbaatar & S-

zlam, 2015), are usually applied in a sentence reader. It contains two parts. The first part is a PE (Sukhbaatar & Szlam, 2015) layer, which is used to produce original representations of facts s_i by:

$$s_i = \sum_j (l_j \cdot Ax_{ij}) \quad (1)$$

Where A is a word embedding matrix, l_j is a column vector composed of the elements $l_{kj} = (1 - j/J) - (k/d)(1 - 2j/J)$, with the number of words in the sentence J and the dimension of the word embedding d. The second part is a bidirectional gated recurrent neural network (Schuster & Paliwal, 1997; Chung et al., 2014), which is used to produce final representations of facts $\overleftrightarrow{f_i}$. A same structure with different parameters is adopted to produce another final representations of facts $\overleftrightarrow{f_i^{(a)}}$, which are used to produce attention weights in the episodic memory module.

Question module: The question module of our basic model also is a sentence reader to process a question. It includes a Recurrent Neural Network (RNN). The final hidden state is seen as the representation of a question q.

Episodic memory module: This module is a core part of a DMN, where the input module interacts with the question module. Typically, a DMN uses a recurrent attention structure to achieve the progressive information extraction or reasoning in the episodic memory module. A reasoning step is regarded as a hop. There are two mechanisms in the episodic memory module: an attention mechanism (Luong et al., 2015) and a memory mechanism. The attention mechanism decides how to extract information from facts at each hop, which generally is implemented by attention weights g_i^t:

$$z_i^t = [\overleftrightarrow{f_i^{(a)}} \circ m^{t-1}; \overleftrightarrow{f_i^{(a)}} \circ q; |\overleftrightarrow{f_i^{(a)}} - m^{t-1}|; |\overleftrightarrow{f_i^{(a)}} - q|] \quad (2)$$

$$Z_i^t = W^{(2)} tanh(W^{(1)} z_i^t + b^{(1)}) + b^{(2)} \quad (3)$$

$$g_i^t = \frac{exp(Z_i^t)}{\sum_{k=1}^{n} exp(Z_k^t)} \quad (4)$$

Where $\circ$ denotes element-wise multiplication. z_i^t is a feature set used to produce g_i^t, we notice that representations of a question q, facts $\overleftrightarrow{f_i^{(a)}}$ and previous memory

m_{t-1} are considered when the model extracts information from facts at each hop. After obtaining g_i^t, the DMN uses the AttenGRU mechanism rather than a soft attention mechanism to produce a contextual vector c^t.

Then, the memory mechanism is utilized to generate a new episodic memory m^t based on the previous episodic memory m^{t-1}, the current contextual vector c^t, and the representation of a question q.

$$m^t = ReLU(W[c^t; q; m^{t-1}] + b) \tag{5}$$

Where ReLU is a Rectified Linear Unit and is defined as $f(x) = max(0, x)$ where x is the input.

Answer module: The answer module receives the final output of the episodic memory module m^T to infer an answer by $softmax(W^{(a)}m^T)$. Then, the model is trained by minimizing cross-entropy error. In addition, whether an answer is composed of a single word or multiple words, we regard it as one word and do not use another RNN to produce the answer. For the English bAbI dataset, answers in several tasks such as QA8 and QA19 consist of multiple words and the model perform well on them when each answer is treated as one word. Thus, we simplify this part.

3.2. Attention based GRU

Because of the relatively concise structure and good properties of the GRU, an increasing number of studies use it to construct their models. The traditional GRU is implemented as follows:

$$u_i = \sigma(W^{(u)}x_i + U^{(u)}h_{i-1} + b^{(u)}) \tag{6}$$

$$r_i = \sigma(W^{(r)}x_i + U^{(r)}h_{i-1} + b^{(r)}) \tag{7}$$

$$\widetilde{h_i} = \tanh(W^{(h)}x_i + r_i \circ U^{(h)}h_{i-1} + b^{(h)}) \tag{8}$$

$$h_i = u_i \circ \widetilde{h_i} + (1 - u_i) \circ h_{i-1}, \tag{9}$$

Where u_i , r_i and h_i represent the update gate, the reset gate, and the hidden state of a GRU, respectively. Ideally, the update gate u_i controls how much of

the previous information is retained, and the reset gate r_i determines how to combine the previous information with a new input.

The introduction of the attention mechanism brings substantial improvement for RNNs, which makes a model pay more attention on more significant elements. In textual translation (Sutskever et al., 2014; Dzmitry Bahdanau, 2015), it significantly improves the quality of the translation. Attention based GRU (AttenGRU) is a new attention pattern (Dzmitry Bahdanau, 2015) for Textual QA, which combines the properties of a gated recurrent neural network with traditional attention mechanism. It can extract positional information of facts and significant information from facts. There are two types of AttenGRUs - $AttenGRU_1$ and $AttenGRU_2$. $AttenGRU_1$ uses attention weights g_i to directly modify the internal mechanics of a traditional GRU using Equation 10 instead of Equation 9.

$$h_i = g_i \circ \widetilde{h_i} + (1 - g_i) \circ h_{i-1} \tag{10}$$

$AttenGRU_2$ does not modify the traditional GRU, but adds Equation 11 behind Equation 9 to produce a new hidden state.

$$h_i^{new} = g_i \circ h_i + (1 - g_i) \circ h_{i-1} \tag{11}$$

$AttenGRU_1$ and $AttenGRU_2$ use attention weights g_i to control the updating of current information near the input port and the output port of a traditional GRU, respectively.

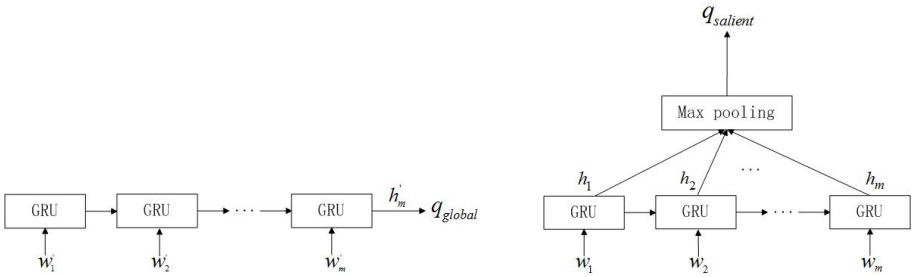
It is worth noting that we apply a slightly different AttenGRU by combining g_i produced by a softmax function with $AttenGRU_2$. Other works use it by combining g_i produced by a sigmoid function with $AttenGRU_2$, or combining g_i produced by a softmax function with $AttenGRU_1$, etc.

4. Enhanced question understanding with dynamic memory networks (EnDMN)

4.1. Multiple representations

As previously mentioned, there are numerous effective models for Textual QA tasks. However, all of them only use one type of features from inputs at each hop. Actually, there are various different features in them. For example, when a human faces a question, he/she receives all kinds of information from the question, such as the common knowledge including the meaning or the type of the question, the logic relationship between different parts of the question, etc. Since, we produce a global representation q_{global} and a salient representation $q_{salient}$ of a question at each hop. q_{global} is expected to include common knowledge of the question, and $q_{salient}$ is expected to automatically extract salient features of the question required for each hop. Namely, q_{global} and $q_{salient}$ are used to represent a question simultaneously rather than a single representation in previous studies.

Compared to some other studies which focus on the facts, we focus more attention on the question. A gated recurrent neural network is usually used in the question module to produce the representation of a question, which is also used to produce q_{global} in our models. However, we build a finer network to produce $q_{salient}$. Because $q_{salient}$ is expected to extract required features about a question for each hop, how to focus more attention on the required information is significant. In particular, when the question is complex, such as a long sentence that includes multifaceted information, using a better strategy to extract salient features layer by layer is advantageous to reveal the relationship of multifaceted features of a question. Salient features can be produced by various attention methods such as hard-attention, local-attention and soft-attention. However, we add another question module which contains a gated recurrent neural network layer and a max-pooling layer to produce $q_{salient}$ due to its conciseness and



(a) q_{global} (b) $q_{salient}$

Figure 2: Internal structures of two question modules. $w^{'}, w$ are word embeddings of a question.

effectiveness for our reasoning tasks. They are implemented as follows:

$$q_{global}(l) = h_m^{'}(l) \tag{12}$$

$$q_{salient}(l) = max(h_1(l), h_2(l), ..., h_m(l)), \quad l = 1, 2, ..., D \tag{13}$$

Here, $h_m^{'}$ is the final hidden state of a gated recurrent neural network in a question module. $h_1, h_2, ...h_m$ are hidden states of a gated recurrent neural network in the other question module, and D is the dimension of hidden states. The maximum values of the elements at the same positions l of hidden states are chosen at the max-pooling layer. The internal structures of two question modules are shown in Figure 2.

4.2. More abundant feature set

Now, We utilize both multiple representations of inputs to produce more abundant feature sets $z^t_{i(global)}$ and $z^t_{i(salient)}$ by:

$$z^t_{i(global)} = [\overleftrightarrow{f_i^{(a')t}} \circ m^{t-1}; \overleftrightarrow{f_i^{(a)t}} \circ q^t_{global}; |\overleftrightarrow{f_i^{(a')t}} - m^{t-1}|; |\overleftrightarrow{f_i^{(a)t}} - q^t_{global}|] \tag{14}$$

and,

$$z^t_{i(salient)} = [\overleftrightarrow{f_i^{(a')t}} \circ m^{t-1}; \overleftrightarrow{f_i^{(a')t}} \circ q^t_{salient}; |\overleftrightarrow{f_i^{(a')t}} - m^{t-1}|; |\overleftrightarrow{f_i^{(a')t}} - q^t_{salient}|] \tag{15}$$

z^t_i reflects interaction among facts, a question and the previous memory. Obviously, multiple representations of inputs can lead to multiple feature sets -

$z^t_{i(global)}$ and $z^t_{i(salient)}$, which are used to form attention weights g^t_i at each step. In this part, we apply two types of networks to produce global and hierarchical salient representations of inputs. Static networks, which denotes networks with same parameters at each step, are used to produce global representations of facts and a question - $\overleftrightarrow{f^{(a)t}_i}$ and q^t_{global} in Equation 14. While dynamic networks, which denotes networks with different parameters at each step, are used to produce salient representations of facts and a question - $\overleftrightarrow{f^{(a')t}_i}$ and $q^t_{salient}$ in Equation 15 required for each step. In the episodic memory module, a dynamic network is applied. A combination of a static network and a dynamic network is beneficial to creating multiple feature sets - $z^t_{i(global)}$ and $z^t_{i(salient)}$. Then, we use them to produce $g^t_{i(global)}$ and $g^t_{i(salient)}$, which form the final attention weights g^t_i by:

$$g^t_i = g^t_{i(global)} + g^t_{i(salient)} \tag{16}$$

After obtaining g^t_i, we use $AttenGRU$ to extract required information of each hop. The process of forming a contextual vector c^t in EnDMN is illustrated in Figure 3.

In brief, we use the same basic framework as DMNs, which includes four main modules described in Section 3.1. We make modifications in the input module and the question module to obtain multiple representations of all inputs. We then infuse them in the episodic memory module. Finally, the output of the episodic memory module is delivered to the answer module.

5. Experiments

5.1. Training details

In this study, we trained and tested our models on a public Textual QA dataset provided by the Facebook - 10k English bAbI dataset (10k is the sample size of each task). This dataset includes 20 types of logical reasoning tasks marked from QA1 to QA20, such as QA16: basic induction, QA17: positional reasoning, and QA19: path finding. An example from QA16 is illustrated in

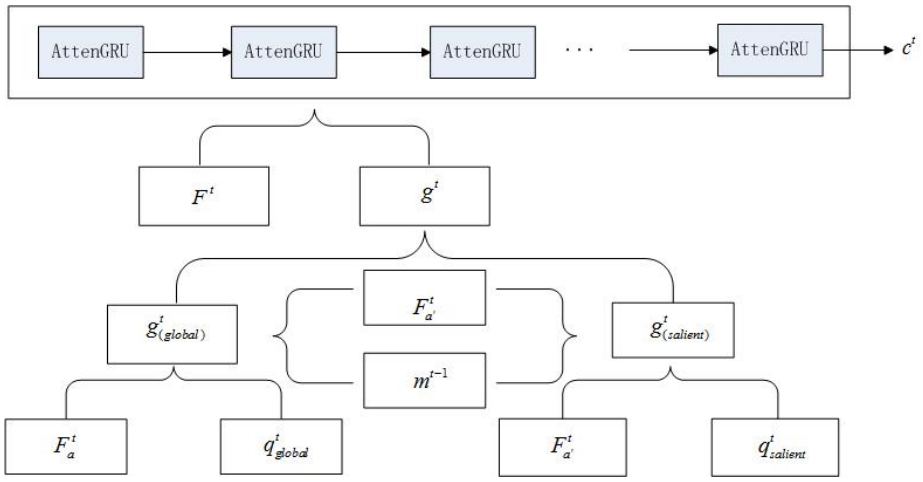


Figure 3: Process of forming a contextual vector c^t in EnDMN. F^t_a, $F^t_{a'}$, F^t are representations of facts. $q^t_{i(global)}$ and $q^t_{i(salient)}$ are representations of a question. $g^t_{i(global)}$, $g^t_{i(salient)}$ and g^t_i are attention weights. m^{t-1} is the previous memory. The initial values of m are $q^1_{i(global)}$ and $q^1_{i(salient)}$ when forming $g^1_{i(global)}$ and $g^1_{i(salient)}$.

Figure 1. Every sample contains facts, a question, an answer, and labels of supporting facts. We refer to a model that does not use labels of supporting facts as a single-supervision model. Otherwise we refer to it as a dual-supervision model.

To evaluate the performance of all the models, we split the original training data into two sets for every Textual QA task: 90% for the training set and 10% for the validation set. The Adam optimizer (Kingma & Ba, 2015) with a learning rate of 0.001 and batch size of 128 were applied for training. Training runs were conducted for up to 256 epochs with early stopping if the validation accuracy could not be improved within the last 20 epochs for all the QA tasks except QA3, for which we used the last 40 epochs. We used the last 70 sentences as inputs of the input module except QA3, for which we used the last 130 sentences. The dimension of the word embedding and hidden states was set to 80, and all the weights were initialized in $[-\sqrt{3},\sqrt{3}]$. We also applied dropout (Srivastava et al., 2014) in the input module and the answer module

Table 2: Settings of models. GRU_s and GRU_d represent a static gated recurrent neural network and a dynamic gated recurrent neural network, respectively.

models	Question module		Parameters	
	Representation	*Structure*	α	*hops*
$DMN1$	q_{global}	GRU_s	0.00	3
$DMN2$	$q_{salient}$	GRU_d	0.00	3
$DMN3$	$q_{global} + q_{salient}$	$GRU_s + GRU_d$	0.00	3
$EnDMN$	$q_{global} + q_{salient}$	$GRU_s + GRU_d$&$maxpooling$	0.00	3
$DMN1(gate)$	q_{global}	GRU_s	0.50	QA3,7,8: 5 others: 3
$DMN2(gate)$	$q_{salient}$	GRU_d	0.50	QA3,7,8: 5 others: 3
$DMN3(gate)$	$q_{global} + q_{salient}$	$GRU_s + GRU_d$	0.50	QA3,7,8: 5 others: 3
$EnDMN(gate)$	$q_{global} + q_{salient}$	$GRU_s + GRU_d$&$maxpooling$	0.50	QA3,7,8: 5 others: 3
Epoch: 256	batch size: 128	Early stop: QA3, 40; others, 20	β: 1.0	Training times: 10
Dimension of word embeddings and hidden states: 80		Length of facts: QA3, 130; others, 70		

(p=0.9) and l2 regularization (Ng, 2004) for all the weights. In order to avoid the oscillation problem caused by random initial values and the random order of training samples, we trained all the models 10 times.

We tested four kinds of models with two different supervision modes. The objective function of these models is $J = \alpha E(gates) + \beta E(answers)$, where E is the standard cross-entropy cost; $gates$ and $answers$ denote the supervision from labels of supporting facts and a real answer, respectively; and α and β are scalars to control the proportion of the cost. If $\alpha = 0$, it is a single-supervision model, otherwise it is a dual-supervision model. The main settings of all models are listed in Table 2.

All of the models were successfully trained via back propagation and did not require any preprocessing. First, we tested four single-supervision models - DMN1, DMN2, DMN3, EnDMN. Then, we compared EnDMN with DMN+ and NR, which have been tested in other works. Finally, we tested four dual-supervision models - DMN1(gate), DMN2(gate), DMN3(gate), EnDMN(gate). We can see progressive improvements of our models, from DMN1/DMN2 to EnDMN, from DMN1(gate)/DMN2(gate) to EnDMN(gate).

Table 3: Mean and minimum error rates (%) of single-supervision models. Other QA Tasks (No. 1, 4, 6, 8, 9, 10, 11, 12, 13, 15, 19, 20) achieved 0 minimum errors across all models.

(a) Mean and minimum error rates (%) of DMN1, DMN2,DMN3, EnDMN

Task	Mean error rates				Minimum error rates			
	DMN1	DMN2	DMN3	**EnDMN**	DMN1	DMN2	DMN3	**EnDMN**
*QA*2	7.7	3.0	**1.3**	1.4	0.1	0.5	0.2	**0.0**
*QA*3	23.2	21.6	**11.2**	**11.2**	6.4	7.8	9.0	**6.0**
*QA*5	0.9	0.7	**0.4**	0.6	0.3	0.5	**0.2**	0.4
*QA*7	2.5	**2.2**	**2.2**	2.3	0.8	1.0	1.2	**0.3**
*QA*14	1.7	**1.4**	**1.4**	1.5	0.5	**0.0**	0.5	**0.0**
*QA*16	48.9	49.0	47.7	**45.8**	43.7	45.3	43.0	**39.0**
*QA*17	6.9	11.1	5.4	**5.1**	1.5	4.5	1.7	**1.3**
*QA*18	3.9	2.9	3.0	**1.5**	0.4	1.0	0.4	**0.3**
Mean error	-	-	-	-	2.7	3.0	2.8	**2.4**

(b) Minimum error rates (%) of EnDMN, DMN+, NR

Task	**EnDMN**	**DMN+**	**NR**
QA2	0.0	0.3	–
QA3	6.0	1.1	–
QA5	0.4	0.5	–
QA7	0.3	2.4	–
QA14	0.0	0.2	–
QA16	39.0	45.3	–
QA17	1.3	4.2	0.9
QA18	0.3	2.1	–
QA19	0.0	0.0	1.6
Mean error	**2.4**	**2.8**	–

5.2. Results and Analysis

The results of the single-supervision models are listed in Table 3. We chose mean and minimum error rates of multiple measurements of these models as main reference values and marked the lowest error rates among these models by boldface. If the mean error rate and the minimum error rate are closer to each other for a model, we think the model is more stable or more resistant to

oscillation caused by random factors. We can find characteristics of four single-supervision models according to the results in Table 3(a). Compared to DMN1 and DMN2, DMN3 has better stability. EnDMN can not only keep stability but also obtain the lowest mean and minimum error rates in the most of tasks and better overall performance. It means EnDMN not only keep good characteristics of DMN1, DMN2 and DMN3, but improve ability of a model further. Then, the testing results of DMN+, NR and EnDMN are shown in Table 3(b), which chose minimum error rates as reference values. Compared to DMN+, NR, we find EnDMN has better or approximately equal performance on the most of tasks, especially on QA7: counting test, QA16: basic induction, QA17: positional reasoning, and QA18: size reasoning, and better overall performance.

Table 4: Mean and minimum error rates (%) of DMN1(gate), DMN2(gate), DMN3(gate), EnDMN(gate). Other Textual QA tasks (No. 1, 2, 4, 6, 8, 9, 10, 11, 12, 13, 14, 15, 18, 19, 20) and (No.16) achieved 0 minimum error and 0.1% minimum error across all models.

Task	Mean error rates				Minimum error rates			
	DMN1 (gate)	DMN2 (gate)	DMN3 (gate)	**EnDMN (gate)**	DMN1 (gate)	DMN2 (gate)	DMN3 (gate)	**EnDMN (gate)**
*QA*3	14.1	7.4	6.4	**6.3**	6.2	6.4	5.5	**3.8**
*QA*5	**0.9**	1.0	1.0	1.0	**0.7**	0.8	0.9	0.8
*QA*7	1.6	0.9	**0.3**	0.5	0.3	0.6	0.1	**0.0**
*QA*17	9.8	12.4	4.2	**4.0**	3.0	5.0	1.9	**1.0**
Mean error	-	-	-	-	0.5	0.6	0.4	**0.3**

Next, we showed the testing results of dual-supervision models in Table 4. More supervision provides more guidance to the training process. Since, there are more 0 test errors for dual-supervision models than single-supervision models. We applied a similar method to train models(gate) and mark the results. We can also see progressive improvements from DMN1(gate)/DMN2(gate) to EnDMN(gate). In brief, the performance of EnDMN(gate) is superior to that of the other models. It obtained the lowest or approximately equal mean and minimum error rates on almost all of tasks than other models.

The above results of experiments prove that our improvements play an active role. We speculate an appropriate combination of global features and hierarchi-

cal salient features from inputs is beneficial to improving the stability of a model and avoiding overfitting in some degree. Global features are required to govern the extraction of the information at each hop for some tasks, but hierarchical salient features for others. Considering both global and hierarchical salient features is a more flexible and comprehensive method to govern the process of inference. Since, EnDMN/DMN3 obtained lower mean error rates. In EnDMN, two distinguishing networks are used to produce global and hierarchical salient features of a question, which is expected to produce better features in different levels. The testing results of experiments prove the modification in the question module can further improve the overall performance. However, the training time of EnDMN is longer than those of other models. In our experiments, the training time of EnDMN/DMN3 is no longer than twice that of DMN1/DMN2. Nevertheless, the test time of every model is almost unaffected. Moreover, the good stability of EnDMN enables it to obtain better results than others in the same time interval. For example, we obtained 0 errors only at the eighth time when we trained QA19 by DMN2, but obtained 0 error in five of the first eight times (5/8) by EnDMN. Finally, we show an example from QA7 to reveal how EnDMN(gate) controls the extraction of the information in Figure 4.

6. Conclusions

In this work, we first summarized and analyzed the DMN and attention based GRU mechanism, and then introduced our improved model - EnDMN/EnDMN(gate), which can simultaneously consider both global and hierarchical salient features from inputs to control the extraction of the information at each hop. The results showed that EnDMN/EnDMN(gate) had better stability and accuracy than other models. Although EnDMN/EnDMN(gate) improves the overall performance, there is still room for improvement. In the future, we will consider how to produce a more effective salient representation of a question for each type of reasoning tasks or build a more flexible pattern to infuse the global and salient features from inputs.

Facts	$\sum_t g_{i(global)^t}$	$\sum_t g_{i(salient)^t}$	$\sum_t g_i^t$
Sandra went to the hallway.			
Sandra grabbed the apple there.	0.19	0.32	0.51
Daniel moved to the kitchen.			
Sandra got the milk there.	0.46	0.50	0.96
Mary got the football there.			
Sandra went back to the office.			
Sandra put down the apple.	0.35	0.16	0.51
Daniel journeyed to the hallway.			
Question :	*How many objects is Sandra carrying*?		

Figure 4: An example shows how a model pays attention on facts based on global features, hierarchical salient features, and both of them. We labeled supporting facts with boldface and showed the first three maximum value of the sum of $g^t_{i(global)}$, $g^t_{i(salient)}$ and g^t_i at each hop(Equation 16) with blue blocks, respectively. The deeper color means the larger value. Comparing the results of them, we find a model considering both global features and hierarchical salient features focuses intensely on every real supporting fact.

Acknowledgments: This work was partially supported by China Scholarship Council, NSERC OGP0046506, the Canada Research Chair program, and China's National Key Research and Development Program under grant 2016YFB1000902. Chunyi Yue proposed the initial idea. Hanqiang Cao, Kun Xiong and Ming Li provided ideas in the design of the models and helped writing the paper. Chunyi Yue designed and performed the experiments and analyzed the results. Kun Xiong and Anqi Cui helped with technical discussions and gave some suggestions for experiments. Haocheng Qin improved the code quality.

References

Andreas, J., Rohrbach, M., Darrell, T., & Klein, D. (2016). Learning to Compose Neural Networks for Question Answering. In *the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics*. San Diego, CA.

Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Lawrence Zitnick, C., & Parikh, D. (2015). Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision 2015* (pp. 2425–2433). Santiago, Chile. doi:10.1109/ICCV.2015.279.

Bhati, R., & Prasad, S. (2016). Open domain question answering system using cognitive computing. In *Cloud System and Big Data Engineering (Confluence), 2016 6th International Conference* (pp. 34–39). IEEE.

Chang, S., & Pal, A. (2013). Routing questions for collaborative answering in community question answering. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining* (pp. 494–501). ACM.

Cho, K., van Merrienboer, B., Bahdanau, D., & Bengio, Y. (2014). On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. In *Proceedings of SSST-8, The Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation* (pp. 103–111). Doha, Qatar.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. In *Advances in Neural Information Processing Systems 27* 47. Montreal, Canada.

Dzmitry Bahdanau, Y. B., Kyunghyun Cho (2015). Neural Machine Translation By Jointly Learning To Align and Translate. In *the 3rd International Conference on Learning Representations*. San Diego, CA.

Fan, J., Kalyanpur, A., Gondek, D. C., & Ferrucci, D. (2012). Automatic knowledge extraction from documents. *IBM Journal of Research and Development*, *56*, 5:1–5:10. doi:10.1147/JRD.2012.2186519.

Graves, A., Wayne, G., & Danihelka, I. (2014). Neural turing machines. arXiv preprint. arXiv:1410.5401.

Hochreiter, S., & Schmidhuber, J. J. (1997). Long short-term memory. *Neural Computation*, *9*, 1735–1780. doi:10.1162/neco.1997.9.8.1735.

Hu, B., Lu, Z., Li, H., & Chen, Q. (2014). Convolutional Neural Network Architectures for Matching Natural Language Sentences. In *Advances in Neural Information Processing Systems 2014* (pp. 2042–2050). Montreal, Canada.

Jackson, P. (1998). Introduction to expert systems (3rd edition). In *Introduction To Expert Systems (3rd Edition)* chapter 1. Boston, Massachusetts: Addison-Wesley.

Kalchbrenner, N., & Blunsom, P. (2013). Recurrent Continuous Translation Models. In *the 2013 Conference on Empirical Methods on Natural Language Processing* (pp. 1700–1709). Seattle, USA.

Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing* (pp. 1746–1751). Doha, Qatar.

Kingma, D., & Ba, J. (2015). Adam: A method for stochastic optimization. In *the 3rd International Conference for Learning Representations*. San Diego, CA.

Kumar, A., Irsoy, O., Ondruska, P., Iyyer, M., Bradbury, J., Gulrajani, I., Zhong, V., Paulus, R., & Socher, R. (2016). Ask Me Anything: Dynamic Memory Networks for Natural Language Processing. In *the 33rd International Conference on Machine Learning*. New York.

Lai, S., Xu, L., Liu, K., & Zhao, J. (2015). Recurrent Convolutional Neural Networks for Text Classification. In *the 29th AAAI Conference on Artificial Intelligence* (pp. 2267–2273). Austin Texas, USA.

Loni, B. (2011). A survey of state-of-the-art methods on question classification. *Electrical Engineering Mathematics & Computer Science*, (pp. 1–40).

Luong, M.-T., Pham, H., & Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (pp. 1412–1421). Lisbon, Portugal.

Meng, F., Lu, Z., Tu, Z., Li, H., & Liu, Q. (2016). A Deep Memory-based Architecture for Sequence-to-Sequence Learning. In *the 4th International Conference on Learning Representations*. Puerto Rico.

Ng, A. (2004). Feature selection, L 1 vs. L 2 regularization, and rotational invariance. In *Proceedings of the 21th international conference on Machine learning*. Banff, Canada. doi:10.1145/1015330.1015435.

Peng, B., Lu, Z., Li, H., & Wong, K.-F. (2015). Towards Neural Network-based Reasoning. arXiv preprint. arXiv:1508.05508.

Sahu, T., Nagwani, N., & Verma, S. (2016a). Multivariate beta mixture model for automatic identification of topical authoritative users in community question answering sites. *IEEE Access*, .

Sahu, T. P., Nagwani, N. K., & Verma, S. (2016b). Selecting best answer: An empirical analysis on community question answering sites. *IEEE Access*, *4*, 4797–4808.

Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, *45*, 2673–2681. doi:10.1109/78.650093.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*, 1929–1958.

Sukhbaatar, S., & Szlam, A. (2015). End-To-End Memory Networks. In *Advances in Neural Information Processing Systems 28* (pp. 2431–2439). Montreal, Canada.

Sun, H., Ma, H., Yih, W.-t., Tsai, C.-T., Liu, J., & Chang, M.-W. (2015). Open domain question answering via semantic enrichment. In *Proceedings of the 24th International Conference on World Wide Web* (pp. 1045–1055). ACM.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27* (pp. 3104–3112). Montreal, Canada.

Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). Show and Tell: A Neural Image Caption Generator. In *Conference on Computer Vision and Pattern Recognition* (pp. 3156–3164). Boston, Massachusetts.

Weston, J., Bordes, A., Chopra, S., Rush, A. M., van Merriënboer, B., Joulin, A., & Mikolov, T. (2015a). Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks. arXiv preprint. `arXiv:1502.05698`.

Weston, J., Chopra, S., & Bordes, A. (2015b). Memory networks. In *the 3rd International Conference on Learning Representations*. San Diego, CA.

Xiong, C., Merity, S., & Socher, R. (2016). Dynamic Memory Networks for Visual and Textual Question Answering. In *the 33rd International Conference on Machine Learning*. New York.

Xu, K., Courville, A., Zemel, R. S., & Bengio, Y. (2015). Show , Attend and Tell : Neural Image Caption Generation with Visual Attention. In *the 32nd International Conference on Machine Learning*. Lille, France.

Yang, Z., He, X., Gao, J., Deng, L., & Smola, A. (2015). Stacked Attention Networks for Image Question Answering. In *IEEE Conference on Computer Vision and Pattern Recognition* 1 (pp. 21–29). Boston, Massachusetts.

Yu, Y., Zhang, W., Hang, C.-W., & Zhou, B. (2015). Empirical Study on Deep Learning Models for QA. arXiv preprint. `arXiv:1510.07526`.