Teaching Machine How to Think by Natural Language: A study on Machine Reading Comprehension

Anonymous authors
Paper under double-blind review

Abstract

Deep learning ends up as a *black box*, in which how it makes the decision cannot be directly understood by humans, let alone guide the reasoning process of deep network. In this work, we seek the possibility to guide the learning of network in reading comprehension task by *natural language*. Two approaches are proposed. In the first approach, the latent representation in the neural network is deciphered into text by a decoder; in the second approach, deep network uses text as latent representation. Human tutor provides ground truth for the output of the decoder or latent representation represented by text. On the bAbI QA tasks, we found that with the guidance on a few examples, the model can achieve the same performance with remarkably less training examples.

1 Introduction

People are now fascinated by the amazing power of deep neural networks in many fields. In reading comprehension, image recognition and speech recognition, the performance of the deep learning based models is almost comparable with human on some benchmark corpora. However, usually deep neural network ends up as a *black box*, in which how it makes the decision cannot be directly understood by humans. When machine makes wrong decision, usually people can only do some parameter engineering or feed the machine with more training data.

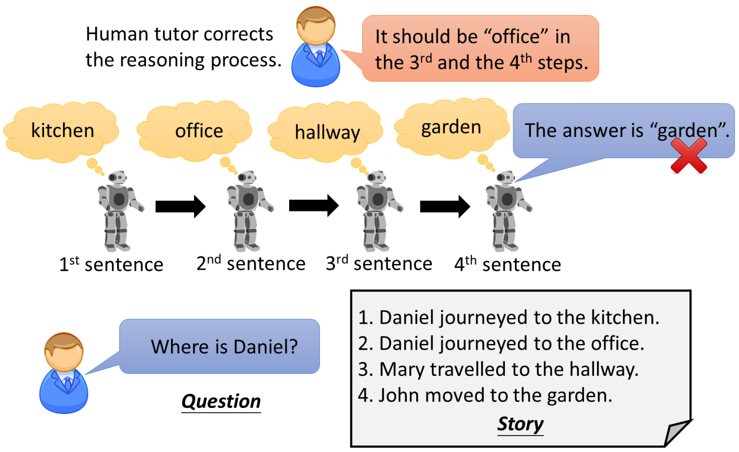
The latent representation in deep network can be analogized as what machine thinks in its mind when it solves a task. If we can guide the machine to not only get the correct answer but also *think* in the right way, the learning results can be more robust. People have already begun to elaborate on the meaning within the latent representation (Montavon et al., 2017; Doshi-Velez & Kim, 2017), but only the deep learning experts know how to utilize the results. In this work, we seek the possibility to guide the representation learning by natural language in reading comprehension.

In reading comprehension, the machine goes through several rounds of deductions, the latent representation in deep network implicitly carries knowledge that the machine learns or infers from the given story during the reasoning process. In this paper, we demonstrate how to *translating* the latent vector into natural language, so the reasoning process can be shown in natural language. Then we show a novel approach of guiding the reasoning process. Because with the proposed approaches, the tutor communicates with the machine directly by natural language, to be a machine tutor, one does not have to be a deep learning expert.

The idea of the proposed framework is shown in Figure 1. Given the question, machine reads the story sentence-by-sentence to find the answer. Each time when it reads a sentence, machine updates what it has inferred. For example, given the question "*Where is Daniel*", when machine reads the sentence "*Daniel journeyed to the kitchen*", its memory stores the location of *Daniel*, which is *kitchen*. The stored information is represented by latent representation in typical models, and it cannot be directed interpreted (we do not know machine truly store the location *kitchen* or not). In Figure 1, machine did not obtain the correct answer eventually. We can tune the hyperparameters until machine get the correct answer, but we do not know machine truly learns how to reasoning or just memorize the answer. With the proposed approaches, the machine can represent what it has inferred by natural language. Once we figure out what the machine is learning or inferring when it reads the story, we will have an opportunity to communicate with the machine intuitively, and guide it to become more intelligent. In the example, the tutor knows machine gets confused when it reads the 3rd and the 4th sentences, so the tutor directly use natural language to provide the correct reasoning. Then machine updates its model based on the correction.

In this paper, we propose two approaches to guide the latent representation learning by natural language. In the first approach, a reading comprehension model is learned as usual, and the latent representation in the neural network which is a vector with real numbers is deciphered into text by a decoder. Then the machine tutor provides reference output for the decoder to guide the model training. In the second approach, a deep network using text as latent representation is learned. In typical networks, different layers use real number vector to transmit information. Here the previous layer outputs a natural language sentence as the input of the next layer. Because the latent representation is text, human can directly correct the latent representation. On the bAbI QA tasks, we found with the guidance on a few examples (less than twenty), the model can achieve the same performance with remarkably less training examples.

The remainder of this paper is organized as below. In Session 2, we review the related work. The reading comprehension model used in this study is briefly reviewed in Section 3. The proposed approaches are in Section 4. The experimental setting and results are in Section 5, and Section 6 is the concluding remarks.



1. Daniel journeyed to the kitchen.
2. Daniel journeyed to the office.
3. Mary travelled to the hallway.
4. John moved to the garden.

Figure 1: Machine describes its reasoning process in natural language. By monitoring the reasoning process, human tutors machine by natural language.

2 RELATED WORK

2.1 READING COMPREHENSION

Deep learning for reading comprehension has attracted much attention (Bordes et al., 2015; Kumar et al., 2015; Xiong et al., 2016; Hermann et al., 2015; Shih et al., 2015). Conventional reading comprehension systems are designed as a cascade of several components (syntactic parser, semantic parser, etc.). In end-to-end reading comprehension model, a neural network which takes a question and a story as input and an answer as output is directly learned from training data, making it possible to jointly learn components in conventional QA. The mechanisms like attention and hopping are widely used in reading comprehension model to model the deduction process (Weston et al.; Sukhbaatar et al., 2015). In a dynamic memory network (DMN) (Kumar et al., 2015), questions trigger an iterative attention process which allows the model to condition its attention on the inputs and result of previous iterations. Neural Reasoner (Peng et al., 2015) infers over multiple supporting facts to find an answer to a given question. The Recurrent Entity Network (EntNet) (Henaff et al., 2017) is the first method to solve all the tasks in the 10k training examples setting of bAbI. The Query-Reduction Network (QRN) (Seo et al., 2017) effectively handles both short-term and long-term sequential dependencies to reason over multiple facts. To achieve complex relational reasoning, new models are proposed (Bansal et al., 2017; Pavez et al., 2018; Santoro et al., 2017; Palm et al., 2018). Recent attention mechanisms and network architectures have been shown to be helpful for SQuAD (Xiong et al., 2017; Seo et al., 2016; Wang et al., 2017; Hu et al., 2017; Huang et al., 2017), and on the SQuAD leaderboard[1], deep learning-based models are competitive with human performance (Yu et al., 2018; Cui et al., 2017). Most of existing works in reading comprehension are dedicated to improving the performance of answer predictions, while leaving the explanation of answering unexploited.

2.2 Guiding Reasoning Process

In Visual Question Answering (VQA) task, to guide the reasoning process, the network architectures of the models are specially designed (Hudson & Manning, 2018; Andreas et al., 2016; Johnson et al., 2017; Cao et al., 2018). For example, neural modular networks build the compositional structure from the parsing results of the questions, which makes the reasoning process in the network easy to be interpreted (Andreas et al., 2016). Some reading comprehension models are also designed to guide the reasoning process. The analysis of EntNet shows that the model has indeed stored locations of all of the objects and characters in its memory slots which reflect the final state of the story (Henaff et al., 2017). The Interpretable Reasoning Network (IRN) (Zhou et al., 2018) makes reasoning on multi-relation questions with multiple triples in knowledge base, and the intermediate entities and relations predicted by the reasoning process construct traceable reasoning paths. QRN (Seo et al., 2017) considers the context sentences as a sequence of state-changing triggers to transform the original query to a more informed query. The authors of QRN claim that the hidden vectors in the model represent the informed queries, but only an example of the informed queries is shown in (Seo et al., 2017) without further analysis and utilization. In the previous work, the reasoning process is guided by carefully designing the network architecture, which can only be done by deep learning expert.

There are some attempt about producing simple text descriptions for AI interpretability. The idea has been applied on classification (Barratt, 2017), VQA (Wang et al., 2015; 2016; Aditya et al., 2018; Wu et al., 2017; Zhou et al., 2017) and diagnostic report generation (Wang et al., 2018; Gale et al., 2018). Existing models designed to produce interpretable traces of their decision-making process typically require hand-crafted rules or extra annotation about the traces for supervised learning at training time. The proposed approaches can also generate interpretable text descriptions, but here we focus on using the text descriptions to guide the model training.

3 Reading Comprehension Model

The reading comprehension task considered here is story-based question answering (QA). In story-based QA, the input is a story $\mathcal{X}$ and a question Q both in natural language. The story is as a sequence of sentences, $\mathcal{X} = \{X_1, X_2 \dots X_T\}$, where T is the number of sentences in the story, while X_t represents the t-th sentence. The output is the distribution y for predicted answer.

The reading comprehension model used in this paper is based on QRN (Seo et al., 2017), whose network architecture is shown in Figure 2. Each sentence X_t and question Q are first transformed into d-dimensional vectors $\mathbf{x}_t \in \mathcal{R}^n$ and $\mathbf{q} \in \mathcal{R}^n$ respectively by an encoder E, that is, $\mathbf{x}_t = E(X_t)$ and $\mathbf{q} = E(Q)$. Then L layers of QRN units use $\mathbf{x}_t$ and $\mathbf{q}$ to obtain the answer y. At each time step, the QRN layers take one sentence $\mathbf{x}_t$ as input. The QRN unit at the l-th layer accepts three inputs at the t-th time step: sentence vector $\mathbf{x}_t$, hidden vector $\mathbf{h}_{t-1}^l$ (from the last time step), and hidden vector $\mathbf{h}_t^{l-1}$ (from the previous layer). The output of the QRN unit is $\mathbf{h}_t^l$ (to next time step and to the next layer). $\mathbf{h}_t^l$, which is similar to the hidden state in RNN, carries the inference results in the reasoning process until step t. $\mathbf{h}_t^0$ is set to be $\mathbf{q}$. Two layers of QRN units ($L = 2$) are shown in Figure 2.

The output of the QRN unit in the last layer at the last time step, $\mathbf{h}_T^L$, is transformed into the final answer y by an output module as below.

$$y = softmax(W_o \mathbf{h}_T^L), \tag{1}$$

where the weight matrix $W_o \in \mathcal{R}^{V \times d}$, and y is a V-dimensional vector, and V is the size of the vocabulary including all possible answers. Each training example is a tuple, $(\mathcal{X}, Q, \hat{y})$, where $\hat{y}$

[1] https://rajpurkar.github.io/SQuAD-explorer/

denotes the true answer represented as a one-hot vector. The whole model is learned to minimize the following loss:

$$L_{QA} = \sum_{n=1}^{N} L_{ce}(\hat{y}^n, y^n), \tag{2}$$

where $L_{ce}(\cdot,\cdot)$ is the cross-entropy between two distributions. N is the number of training examples, and $\hat{y}^n$ and y^n are the ground truth and predicted answer respectively.

4 Methodology

We further propose two approaches to realize the idea of expressing the deductive process in natural language, and demonstrate how to guide the reasoning process. The two approaches are described in Sections 4.1 and 4.2 respectively. Based on these approaches, we translate $\mathbf{q}_t^l$ in the QRN model into text. We choose QRN here because it had achieved the state-of-the-art results on the bAbI dataset. The proposed approaches is general and can be applied on other reading comprehension models.

4.1 Learning External Decoder

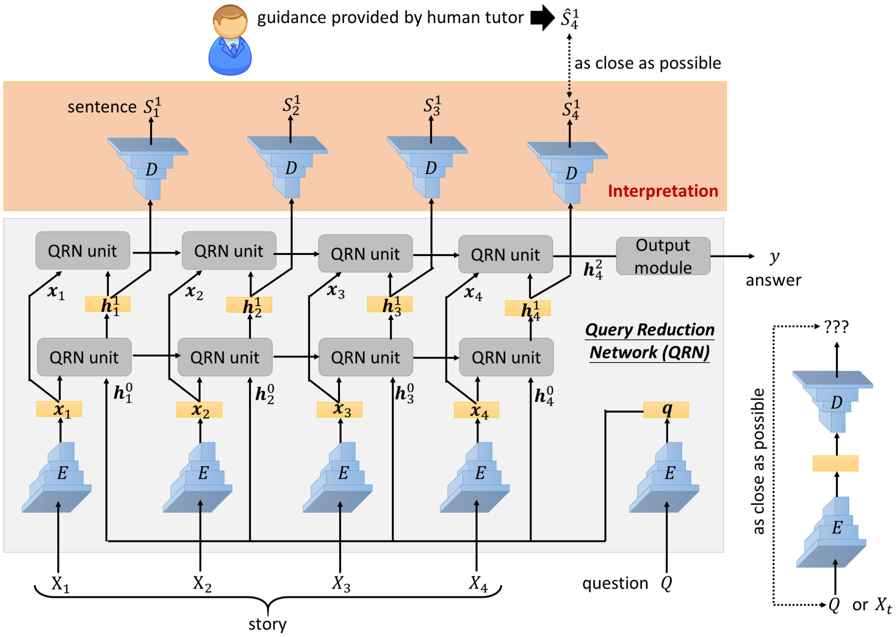


Figure 2: Query reduction network (QRN) and the decoder deciphering the latent representation into natural language. How to guide the machine is also shown in the figure.

The first approach is also shown in Figure 2. In this approach, we aim to use a decoder to decipher the latent representation into human understandable language. Specifically, we target to learn a decoder to transform the vector representation $\mathbf{h}_t^l \in \mathcal{R}^d$ into a natural language sentence S_t^l. By browsing the sequence of sentences, S_1^l to S_T^l, one can have better understanding about the reasoning path of the model.

Because the text corresponding to the latent representation $\mathbf{h}_t^l$ is not available, at the first glance, the decoder cannot be learned without extra supervision. However, we realize that the query vector $\mathbf{q}$ and sentence vector $\mathbf{x}_t$ representing the original natural language question Q and sentence X_t. Therefore, we can learn a decoder D which can reconstruct Q and X_t given $\mathbf{q}$ and $\mathbf{x}_t$ respectively. Given a latent vector, the output of the decoder D is a sequence of word distributions by which the likelihood of generating a specific word sequence can be computed. We use $P_D(Q|\mathbf{q})$ (or $P_D(X_t|\mathbf{x}_t)$) to represent the likelihood that the decoder D generates Q (or X_t) from $\mathbf{q}$ (or $\mathbf{x}_t$). The decoder D learns to minimize the reconstruction loss $L_{reconstruct}$ (or maximize the log likelihood) for all the queries and sentences in the training data,

$$L_{reconstruct} = -\sum_{n=1}^{N} \{logP_D(Q^n|\mathbf{q}^n) + \sum_{t=1}^{T} logP_D(X_t^n|\mathbf{x}_t^n)\}, \tag{3}$$

where Q^n is the question for the n-th example, and x_t^n is the t-th sentence in the story of the n-th example.

Because the decoder D is only trained with $\mathbf{q}$ and $\mathbf{x}_t$, it is very possible that $S_t^l = D(\mathbf{h}_t^l)$ would not be readable without further constraint. Here we further force the distribution of the latent representation $\mathbf{h}_t^l$, $\mathbf{q}$ and $\mathbf{x}_t$ close to normal distribution. The KL divergence between normal distribution and the distribution of $\mathbf{h}_t^l$, $\mathbf{q}$ and $\mathbf{x}_t$ is denoted as L_{KL}. With L_{KL}, one can consider the encoder E and decoder D together form a variational autoencoder (VAE) (Kingma & Welling, 2014; Rezende et al., 2014). The decoder D and the whole QRN model are jointly trained to minimize the following loss function:

$$L = L_{QA} + L_{reconstruct} + L_{KL}, \tag{4}$$

where L_{QA} and $L_{reconstruct}$ have been shown in (2) and (3) respectively.

How to guide the reasoning process is also shown in Figure 2. The human tutor can provide a sentence $\hat{S}_t^l$ as the reference of S_t^l to guide the machine. Given $\hat{S}_t^l$, the network parameters are updated to maximize the log likelihood that the reference sentence $\hat{S}_t^l$ is generated from the decoder given latent representation $\mathbf{h}_t^l$.

4.2 Text as Latent Representation

The second approach is shown in Figure 3. In this approach, we still use the network architecture of QRN, but we directly use text as the latent representation in the network rather than learning an extra decoder. As shown in Figure 3 (only part of the QRN is shown), the model transforms $\mathbf{h}_t^l$ into a word. We pass $\mathbf{h}_t^l$ through a linear transformation and softmax activation function to obtain a vector π with vocabulary size V, which is a distribution over words. Then a word (which is *apple* in Figure 3) is sampled based on the distribution π. Based on the sampling result, the complete sentence S_t^l (*where is the apple*) is generated by template. The next layer uses the encoder E to encoder the sentence S_t^l as the input. The whole model is learn to minimize L_{QA} in (2).

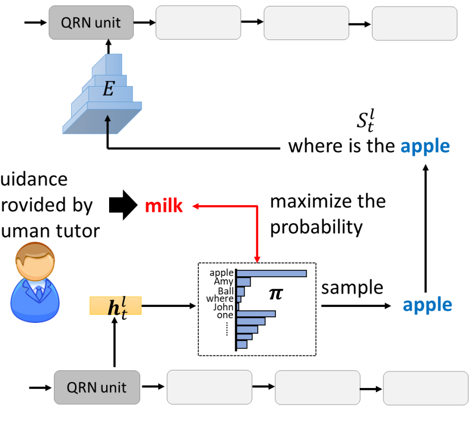


Figure 3: Using text as latent representation.

In this approach, human tutor provides single word to guide the machine. As shown in Figure 3, if the guidance provided by the tutor is *milk*, the network parameters would be updated to maximize the probability corresponding to *milk* in π. After the update, the reasoning sentence S_t^l would become "*where is the milk*".

Because text is discrete, and thus the network in Figure 3 becomes non-differentiable. We address the non-differentiable issues by applying Gumbel-softmax (Jang et al., 2016), but the model is still difficult to train. This is why rather than output the whole sentence as in the first approach, here we only require the model to output a certain word among the vocabulary from latent representation, and then use template to generate a complete sentence.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Dataset The story-based QA dataset, bAbI Weston et al. (2015), is used in the experiments. BAbI is a synthetic dataset created to benchmark question answering models. It contains 20 types of question answer tasks, and each task is comprising a question, a set of statements, and a single-word answer. For each question, only some of the statements contain the relevant information. It contains 1,000 examples for each task in training and testing data. To answer questions in each task requires selecting a set of relevant sentences and applying different kinds of logical reasoning over them. We adopted bAbI rather than other datasets because reasoning process is necessary to solve the tasks. The extraction-based QA corpus like SQuAD may not be suitable because some of the questions can be correctly answered by simple attentive method, and multiple reasoning procedures are not necessary.

Network Architecture For QRN model, we use the hidden state size of 50, memory size of 50, hidden latent vector size of 50, and batch size of 32, same as the setting in (Seo et al., 2017). AdaGrad is utilized as optimizer. The initial learning rate is 0.5 in both models. The loss engaging in training process is split into 3 phases, L_{QA} is adpoted in the first training stage, after certain epoch L_{KL} is added in, lastly summing up L_{QA}, L_{KL}, $L_{reconstruct}$ in the remaining training process. The encoder used here is the same as the one use in (Seo et al., 2017). We obtained, $\mathbf{x}_t$, $\mathbf{q} \in \mathcal{R}^d$ with a trainable embedding matrix $\mathbf{A} \in \mathcal{R}^{d \times V}$ and a trainable encoding matrix $\mathbf{W} \in \mathcal{R}^{J \times d}$, where J represents max sentence length. Also, the symmetric structure is applied on the decoder.

5.2 EXPERIMENT RESULTS

Investigating the Impact of Human Guidance

Human tutor experiments are conducted on both model variants (learning external decoder in Section 4.1, text as latent representations in Section 4.2). Tables 1 and 2 are the illustrations of the guidance we provided to each sentence in the story. Here, we hope to teach the machine to react to the triggering fact it received. Due to the different natures of tasks in bAbI, the guidance is provided with different principles. In task 2, We provide the label indicating the supposed keyword of deductive question (Table 1). Whereas for some other tasks, such as task 1, the question should remain the same throughout the whole reasoning process. Therefore, the label we provided is the keyword in the story sentences that we deem it essential for the machine to recognize in its reasoning process (Table 2).

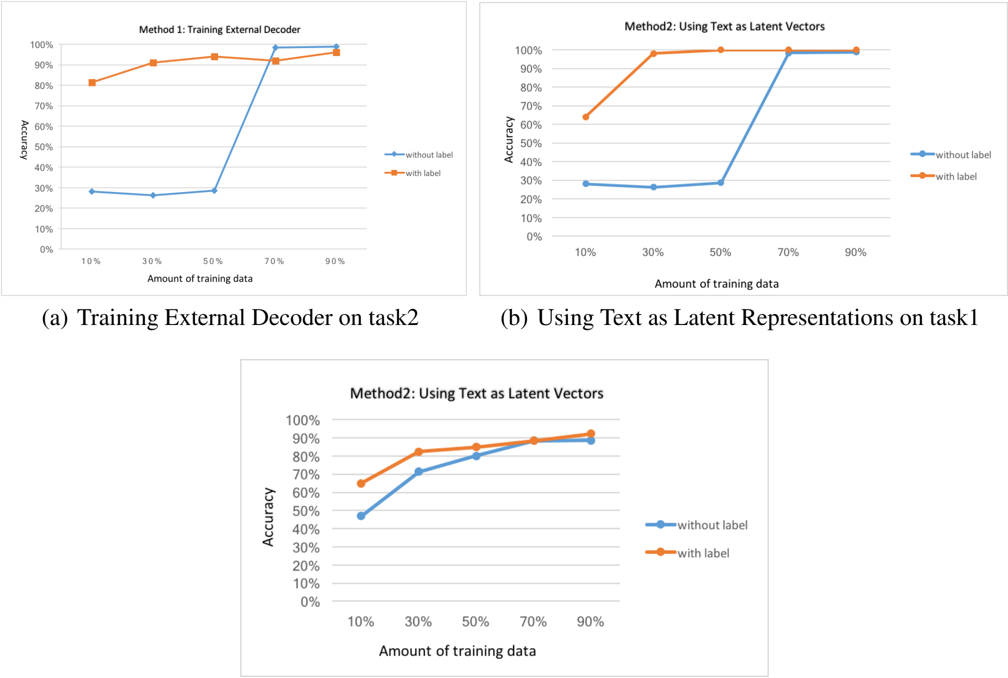
Table 1: Illustration of guidance for bAbI QA dataset, task2

Question	Story (facts)	Guidance	Answer
Where is the apple?	Mary went back to the garden	Where is the apple	Kitchen.
	Mary grabbed the milk there	Where is the apple	
	Sandra went to the hallway	Where is the apple	
	Mary got the football there	Where is the apple	
	John picked up the apple there	Where is John	
	Daniel went back to the kitchen	Where is John	
	John moved to the kitchen	Where is John	
	Mary left the milk	Where is John	

To show the importance of engaging human guidance on learning, guidance as Tables 1 and 2 is provided on 20 training examples. The systems are evaluated in terms of accuracy with different numbers of training examples. 10%, 30%, 50%, 70%, 90% of the training set are used. In Figure 4, the x-axis is the amount of the training data (10%, 30%, 50%, 70%, 90%); the y-axis is the machine-comprehension accuracy on the testing set. The blue curve and the orange curse are the results for the unguided and guided models.

Table 2: Illustration of guidance for bAbI QA dataset, task1

Question	**Story (facts)**	**Guidance**	**Answer**
Where is Daniel?	Daniel journeyed to the kitchen	kitchen	Office.
	Daniel journeyed to the office	office	
	Mary travelled to the hallway	office	
	John moved to the garden	office	



(a) Training External Decoder on task2

(b) Using Text as Latent Representations on task1

(c) Using Text as Latent Representations on task8

Figure 4: Comparisons of accuracy among different amount of training of examples with the guidance on 20 examples.

We found that human guidance can substantially improve the generalizability of learning, especially with small training data where original QRN model shows a very serious overfitting problem. Even with little human guidance (only 20 examples have guidance), machine is able to imitate human reasoning patterns to solve the certain tasks.

Figure 5 further demonstrates the the power of human guidance with the guidance on even less examples. 10% training data is used. With the guidance on only 10 examples, the performance can achieve nearly 60% accuracy.

Text Description Generated by Models

An example of S_1^1 to S_T^1 from the second approach is shown in Table 7. The decoded words are *office*,*garden*,*John*,*back*. Those are keys words related to the question. The according answer to the given question is *office*. *John* has also been to office, and Sandra was in the *garden* before she went to office. It may be confused that the appearance of those keywords is not in sequential order as the given facts. However, it can be explained by the model architecture of implementing bidirectional layer.

We found that the decoded sentences from the first method are unsatisfactory. In task 2, wherequestion of an object ”Where is the apple?”, should be transformed into ”Where is Amy?”, as Amy took up the apple in the following story. However, the deductive questions fail to change a single

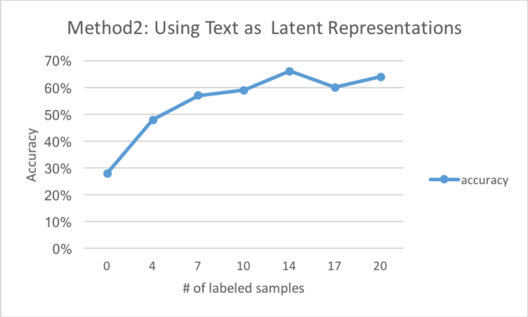


Figure 5: Comparison of accuracy with the guidance provided on 4, 7, 10, 14, 17 and 20 examples. 10% training examples are used here.

Table 3: An example of decoded latent result of machine reasoning.

Question	**Story (facts)**	**Decoded Latent Representation**	**Answer**
Where is Sandra?	1. John journeyed to the **office**	**office**	**office.**
	2. Sandra travelled to the *garden*	*garden*	
	3. Sandra moved to the **office**	back	
	4. Daniel travelled to the hallway	John	
	5. John travelled to the bathroom	**office**	
	6. Daniel journeyed to the garden	John	

word in most of the cases whereas the decoder can learn to perfectly reconstruct original question. It is assumed that the variation of the latent codes, $\mathbf{h}_1^1$ to $\mathbf{h}_T^1$, is too small for the decoder to recognize.

6 Conclusion

In this work, we seek the possibility to guide the learning of network in reading comprehension task by natural language. In the first approach, the latent representation in the neural network is deciphered into text by a decoder; in the second approach, deep network uses text as latent representation. On the bAbI QA tasks, we found that with the guidance on less than twenty examples, the model can achieve the same performance with remarkably less training examples.

For the future work, we will test the proposed approaches on larger and more realistic data sets with different reading comprehension models. We are trying to directly generate complete sentences without templates, and we will use adversarial learning to make the generated text description more readable. We will recruit the people without computer science background to provide the guidance to know whether non-expert can also guide the network training,

References

Somak Aditya, Yezhou Yang, and Chitta Baral. Explicit reasoning over end-to-end neural architectures for visual question answering. In *AAAI*, 2018.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *CVPR*, 2016.

Trapit Bansal, Arvind Neelakantan, and Andrew McCallum. Relnet: End-to-end modeling of entities and relations. In *AKBC*, 2017.

Shane Barratt. InterpNET: Neural introspection for interpretable deep learning. In *NIPS Symposium on Interpretable Machine Learning*, 2017.

A. Bordes, N. Usunier, S. Chopra, and J. Weston. Large-scale simple question answering with memory networks. *CoRR*, abs/1506.02075, 2015.

Qingxing Cao, Xiaodan Liang, Bailin Li, and Liang Lin. Interpretable visual question answering by reasoning on dependency trees. In *arXiv*, 2018.

Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. Attention-over-attention neural networks for reading comprehension. In *ACL*, 2017.

Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. In *arXiv*, 2017.

William Gale, Luke Oakden-Rayner, Gustavo Carneiro, Andrew P Bradley, and Lyle J Palmer. Producing radiologist-quality reports for interpretable artificial intelligence. In *arXiv*, 2018.

Mikael Henaff, Jason Weston, Arthur Szlam, Antoine Bordes, and Yann LeCun. Tracking the world state with recurrent entity networks. In *ICLR*, 2017.

K. M. Hermann, T. Kociský, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom. Teaching machines to read and comprehend. *CoRR*, abs/1506.03340, 2015.

Minghao Hu, Yuxing Peng, and Xipeng Qiu. Reinforced mnemonic reader for machine comprehension. *CoRR, abs/1705.02798*, 2017.

Hsin-Yuan Huang, Chenguang Zhu, Yelong Shen, and Weizhu Chen. Fusionnet: Fusing via fully-aware attention with application to machine comprehension. *arXiv preprint arXiv:1711.07341*, 2017.

Drew A. Hudson and Christopher D. Manning. Compositional attention networks for machine reasoning. In *ICLR*, 2018.

E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. In *arXiv*, 2016.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Judy Hoffman, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. Inferring and executing programs for visual reasoning. In *ICCV*, 2017.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

A. Kumar, O. Irsoy, J. Su, J. Bradbury, R. English, B. Pierce, P. Ondruska, I. Gulrajani, and R. Socher. Ask me anything: Dynamic memory networks for natural language processing. *CoRR*, abs/1506.07285, 2015.

Gregoire Montavon, Wojciech Samek, and Klaus-Robert Muller. Methods for interpreting and understanding deep neural networks. In *arXiv*, 2017.

Rasmus Berg Palm, Ulrich Paquet, and Ole Winther. Recurrent relational networks. In *submitted to NIPS*, 2018.

Juan Pavez, Hector Allende, and Hector Allende-Cid. Working memory networks: Augmenting memory networks with a relational reasoning module. In *ACL*, 2018.

Baolin Peng, Zhengdong Lu, Hang Li, and Kam-Fai Wong. Towards neural network-based reasoning. *CoRR*, abs/1508.05508, 2015.

Danilo J. Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, 2014.

Adam Santoro, David Raposo, David G.T. Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. In *NIPS*, 2017.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*, 2016.

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

Kevin J. Shih, Saurabh Singh, and Derek Hoiem. Where to look: Focus regions for visual question answering. *CoRR*, abs/1511.07394, 2015.

S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus. Weakly supervised memory networks. *CoRR*, abs/1503.08895, 2015.

Peng Wang, Qi Wu, Chunhua Shen, Anton van den Hengel, and Anthony R. Dick. Explicit knowledge-based reasoning for visual question answering. *CoRR*, abs/1511.02570, 2015.

Peng Wang, Qi Wu, Chunhua Shen, and Anton van den Hengel. The VQA-machine: Learning how to use existing vision algorithms to answer new questions. *CoRR*, abs/1612.05386, 2016.

Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pp. 189–198, 2017.

Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, and Ronald M. Summers. TieNet: Text-image embedding network for common thorax disease classification and reporting in chest X-rays. In *CVPR*, 2018.

J. Weston, S. Chopra, and A. Bordes. Memory networks. *CoRR*, abs/1410.3916.

Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. Towards AI-complete question answering: A set of prerequisite toy tasks. *CoRR*, abs/1502.05698, 2015.

Qi Wu, Peng Wang, Chunhua Shen, Ian Reid, and Anton van den Hengel. Are you talking to me? reasoned visual dialog generation through adversarial learning. In *arXiv*, 2017.

C. Xiong, S. Merity, and R. Socher. Dynamic memory networks for visual and textual question answering. *CoRR*, abs/1603.01417, 2016.

Caiming Xiong, Victor Zhong, and Richard Socher. Dynamic coattention networks for question answering. In *ICLR*, 2017.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. QANet: Combining local convolution with global self-attention for reading comprehension. In *ICLR*, 2018.

Mantong Zhou, Minlie Huang, and Xiaoyan Zhu. An interpretable reasoning network for multi-relation question answering. In *COLING*, 2018.

Yiyi Zhou, Rongrong Ji, Jinsong Su, Yongjian Wu, and Yunsheng Wu. More than an answer: Neural pivot network for visual qestion answering. In *Proceedings of the 2017 ACM on Multimedia Conference*, MM '17, 2017.

A AVERAGE PERFORMANCE

In our experiment, the average performance of model variant 1 *(Using external decoders)* on task 1, task 2 is 98% in accuracy ;while the average performance of model variant 2 *(Using text as latent representation)* on task 1, task 2 is 100%. Both the results are similar with original QRN model(99.4% in accuracy). It is thus proved that the two model invariants still remain same machine comprehension ability with original QRN model .

B LABLE

Table 4: Illustration of lables for bAbI QA dataset, task16

Question	**Story (facts)**	**Supervised Labels**	**Answer**
What color is Lily?	Bernhard is a lion	What color is Lily	yellow
	Brian is a frog	What color is Lily	
	Bernhard is gray	What color is Lily	
	Julius is a frog	What color is Lily	
	Greg is a swan	What color is Greg	
	Brian is gray	What color is Lily	
	Greg is yellow	What color is Greg	
	Julius is gray	What color is Lily	
	Lily is a swan	What color is Lily	

Table 5: Illustration of lables for bAbI QA dataset, task13

Question	**Story (facts)**	**Supervised Labels**	**Answer**
Where is Mary?	Mary and Daniel went to the kitchen	kitchen	bathroom
	Afterwards they went back to the hallway	hallway	
	Daniel and Sandra went back to the bathroom	hallway	
	Then they moved to the kitchen	hallway	
	John and Sandra went back to the office	hallway	
	Following that they moved to the garden	hallway	
	Mary and John travelled to the kitchen	kitchen	
	Afterwards they went to the bathroom	bathroom	

Table 6: Illustration of lables for bAbI QA dataset, task8

Question	**Story (facts)**	**Supervised Labels**	**Answer**
What is Mary carring?	John got the milk there	nothing	milk
	John left the milk	nothing	
	Daniel went to the garden	nothing	
	John got the milk there	nothing	
	Daniel travelled to the bedroom	nothing	
	John discarded the milk	nothing	
	Mary took the milk there	milk	
	Daniel went to the hallway	milk	

C OTHER EXAMPLES

Table 7: An example of decoded latent result of machine reasoning

Question	Story (facts)	Decoded Latent Representation	Answer
Where is Sandra?	1. Sandra travelled to the bedroom	*moved*	**office.**
	2. Sandra journeyed to the garden	back	
	3. Sandra *moved* to the hallway	in	
	4. Sandra travelled to the bathroom	to	
	5. Mary went to the bathroom	**office**	
	6. <u>John</u> *moved* to the **office**	**office**	
	7. Sandra went back to the **office**	Where	
	8. Mary travelled to the hallway	<u>John</u>	