Towards Neural Network-based Reasoning

Baolin Peng[1]* Zhengdong Lu[2] Hang Li[2] Kam-Fai Wong[1]
[1]Dept. of Systems Engineering & Engineering Management,
The Chinese University of Hong Kong
{blpeng, kfwong}@se.cuhk.edu.hk
[2]Noah's Ark Lab, Huawei Technologies
{Lu.Zhengdong, HangLi.HL}@huawei.com

Abstract

We propose NEURAL REASONER , a framework for neural network-based reasoning over natural language sentences. Given a question, NEURAL REASONER can infer over multiple supporting facts and find an answer to the question in specific forms. NEURAL REASONER has 1) a specific interaction-pooling mechanism, allowing it to examine multiple facts, and 2) a deep architecture, allowing it to model the complicated logical relations in reasoning tasks. Assuming no particular structure exists in the question and facts, NEURAL REASONER is able to accommodate different types of reasoning and different forms of language expressions. Despite the model complexity, NEURAL REASONER can still be trained effectively in an end-to-end manner. Our empirical studies show that NEURAL REASONER can outperform existing neural reasoning systems with remarkable margins on two difficult artificial tasks (Positional Reasoning and Path Finding) proposed in [6]. For example, it improves the accuracy on Path Finding(10K) from 33.4% [4] to over 98%.

1 Introduction

Reasoning is essential to natural language processing tasks, most obviously in examples like document summarization, question-answering, and dialogue. Previous efforts in this direction are built on rule-based models, requiring first mapping natural languages to logic forms and then inference over them. The mapping (roughly corresponding to semantic parsing), and the inference, are by no means easy, given the variability and flexibility of natural language, the variety of the reasoning tasks, and the brittleness of a rule-based system.

Just recently, there is some new effort, mainly represented by Memory Network and its dynamic variants [7, 3], trying to build a purely neural network-based reasoning system with fully distributed semantics that can infer over multiple facts to answer simple questions, all in natural language.

In this paper we give a more systematic treatment of the problem and propose a flexible neural reasoning system, named NEURAL REASONER. It is purely neural network based and can be trained in an end-to-end way [4], using only supervision from the final answer. Our contributions are mainly two-folds

- we propose a novel neural reasoning system NEURAL REASONER that can infer over multiple facts in a way insensitive to 1) the number of supporting facts, 2)the form of language, and 3) the type of reasoning;
- we give a particular instantiation of NEURAL REASONER and a multi-task training method for effectively fitting the model with relatively small amount of data, yielding significantly better results than existing neural models on two artificial reasoning task;

*The work is done when the first author worked as intern at Noah's Ark Lab, Huawei Technologies.

2 Model

In a nutshell, question and facts, as symbol sequences, are first converted to vectorial representations in the encoding layer via recurrent neural networks (RNNs). The vectorial representations are then fed to the reasoning layers, where the question and the facts get updated through an nonlinear transformation jointly controlled by deep neural networks (DNNs)and pooling. Finally at the answering layer, the resulted question representation is used to generate the final answer to the question.

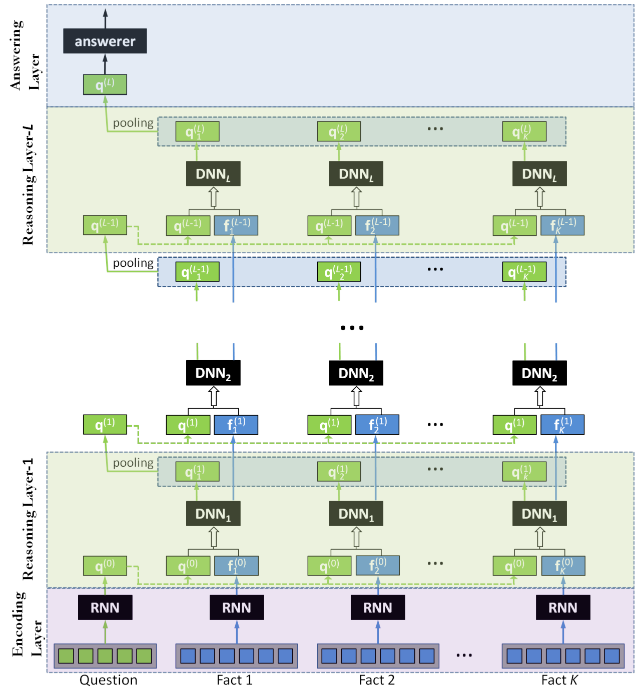

Figure 1: A diagram of our implementation of NEURAL REASONER with L reasoning layers, operating on one question and K facts.

2.1 Encoding Layer

The encoding layer is designed to find semantic representations of question and facts. Suppose that we are given a fact or a question as word sequence $\{x_1, \cdots, x_T\}$, the encoding module summarizes the word sequence with a vector with fixed length. We have different modeling choices for this purpose, e.g., CNN [2] and RNN [5], while in this paper we use GRU [1] as the encoding module.

2.2 Reasoning Layers

2.2.1 Question-Fact Interaction

On reasoning layer ℓ, the k^{th} interaction is between $\mathbf{q}^{(\ell-1)}$ and $\mathbf{f}_k^{(\ell-1)}$, resulting in updated representations $\mathbf{q}_k^{(\ell)}$ and $\mathbf{f}_k^{(\ell)}$

$$[\mathbf{q}_k^{(\ell)}, \mathbf{f}_k^{(\ell)}] \stackrel{\text{def}}{=} g_{\mathsf{DNN}_\ell}([(\mathbf{q}^{(\ell-1)})^\top, {\mathbf{f}_k^{(\ell-1)}}^\top]^\top; \Theta_\ell), \tag{1}$$

with Θ_ℓ being the parameters. In general, $\mathbf{q}_k^{(\ell)}$ and $\mathbf{f}_k^{(\ell)}$ can be of different dimensionality as those of the previous layers. In the simplest case with a single layer in DNN_ℓ, we have

$$\mathbf{q}_k^{(\ell)} \stackrel{\text{def}}{=} \sigma(\mathbf{W}_\ell^\top [(\mathbf{q}^{(\ell-1)})^\top, {\mathbf{f}_k^{(\ell-1)}}^\top] + \mathbf{b}_\ell), \tag{2}$$

Roughly speaking, $\mathbf{q}_k^{(\ell)}$ contains the update of the system's understanding on answering the question after its interaction with fact K, while $\mathbf{f}_k^{(\ell)}$ records the change of the K^{th} fact. Therefore, $\{(\mathbf{q}_k^{(\ell)}$,$\mathbf{f}_k^{(\ell)})\}$ constitute the "state" of the reasoning process. We choose to keep the representation unupdated on each layer when conducting experiments.

2.2.2 Pooling

Pooling aims to fuse the understanding of the question right after its interaction with all the facts to form the current status of the question, through which we can enable the comparison between different facts. There are several strategies for this pooling

Average/Max Pooling: To obtain the n^{th} element in $\mathbf{q}^{(\ell)}$, we can take the average or the maximum of the elements at the same location from $\{\mathbf{q}_1^{(\ell)}, \cdots \mathbf{q}_K^{(\ell)}\}$ At layer-L, the query representation $\mathbf{q}^{(L)}$ after the pooling will serve as the features for the final decision.

2.3 Answering Layer

For simplicity, we focus on the reasoning tasks which can be formulated as classification with predetermined classes. At reasoning Layer-L, it performs pooling over the intermediate results to select important information for further uses.

$$\mathbf{q} = \mathsf{pool}(\{\mathbf{q}_1^{(L)}, \mathbf{q}_2^{(L)}, \cdots, \mathbf{q}_K^{(L)}\}) \quad (3)$$

$$\mathbf{y} = \mathsf{softmax}(\mathbf{W}_{\mathsf{softmax}}^\top \mathbf{q}^{(L)}) \quad (4)$$

After reaching the last reasoning step, in this paper we take two steps, Q^2 is sent to a standard softmax layer to generate an answer which is formulated as a classification problem.

3 Auxiliary Training for Question/Fact Representation

We use auxiliary training to facilitate the learning of representations of question and facts. Basically, in addition to using the learned representations of question and facts in the reasoning process, we also use those representations to reconstruct the original questions.

In the auxiliary training, we intend to achieve the following two goals: 1) to compensate the lack of supervision in the learning task. 2)to introduce beneficial bias for the representation learning task.

We take the simplest way to fuse the auxiliary tasks (recovering) with the main task (reasoning) through linearly combining their costs with trade-off parameter α

$$E = \alpha E_{\mathsf{recovering}} + (1-\alpha) E_{\mathsf{reasoning}} \quad (5)$$

where$E_{\mathsf{reasoning}}$ is the cross entropy loss describing the discrepancy of model prediction from correct answer and $E_{\mathsf{recovering}}$ is the negative log-likelihood of the sequences (question or facts) to be recovered. The likelihood is estimated as in the encoder-decoder framework proposed in [1]. On top of the encoding layer (RNN), we add another decoding layer (RNN) which is trained to sequentially predict words in the original sentence.

4 Experiments

4.1 Setup

We select from bAbI the two most challenging tasks (among the 20 tasks in [6]) Positional Reasoning and Path Finding, to test the reasoning ability of NEURAL REASONER. bAbI is a synthetic question and answering dataset. It contains 20 tasks, and each of them is composed of a set of facts, a question and followed by an answer which is mostly a single word.

4.2 NEURAL REASONER vs. Competitor Models

We compare NEURAL REASONER with the following three neural reasoning models: 1)Memory Network, including the one with step-by-step supervision [7](denoted as MEMORY NET-STEP) and the end-to-end version [4] (denoted as MEMORY NET-N2N), and 2) DYNAMIC MEMORY NETWORK, proposed in [3], also with step-by-step supervision.

In Table 1, we report the performance of a particular case of NEURAL REASONER with 1) two reasoning layers, 2) 2-layer DNNs as the interaction modules in each reasoning layer, and 3) auxiliary task of recovering the original question and facts.

The results are compared against three neural competitors. We have the observations 1) the proposed NEURAL REASONER performs significantly better than Memory Net-N2N, especially with more training data. 2) although not a fair comparison to our model, NEURAL REASONER is actually better than MEMORY NET-N2N, DYNAMIC MEMORY NET on Positional Reasoning (1K) & (10K) as well as Path Finding (10K), with about 20% margin on both tasks with 10K training instances.

Further study of architectural variants of NEURAL REASONERis conducted form the following aspect : 1)the number of reasoning layers, 2) the depth of the interaction DNN, and 3) the auxiliary tasks, with results summarized by Table 2.

	Po.Rea. (1K)	Po. Rea. (10K)	Pa. Fin. (1K)	Pa. Fin. (10K)
Step-by-step Supervision				
MEMORY NET-STEP	65.0%	75.4%	36.0%	68.1 %
DYNAMIC MEMORY NET	59.6%	-	34.5%	-
End-to-End				
MEMORY NET-N2N	59.6%	60.3%	17.2%	33.4%
NEURAL REASONER	66.4%	97.9%	17.3%	87.0%

Table 1: Results on two reasoning tasks. The results of MEMORY NET-STEP, MEMORY NET-N2N, and DYNAMIC MEMORY NET are taken respectively from [7],[4] and [3].

	Po.Rea. (1K)	Po. Rea. (10K)	Pa. Fin. (1K)	Pa. Fin. (10K)
No auxiliary task				
2-layer reasoning, 1-layer DNN	60.2%	72.1%	13.6%	52.2%
2-layer reasoning, 2-layer DNN	59.6%	69.3%	12.3%	54.2%
3-layer reasoning, 3-layer DNN	58.7%	59.7%	13.1%	51.7%
Auxiliary task: Original				
2-layer reasoning, 1-layer DNN	63.1%	93.8%	14.1%	57.0%
2-layer reasoning, 2-layer DNN	66.4%	97.9%	17.3%	87%
3-layer reasoning, 3-layer DNN	69.4%	99.1%	14%	98.4%

Table 2: Results on two reasoning tasks yielded by NEURAL REASONER with different architectural variations.

Auxiliary tasks are essential to the efficacy of NEURAL REASONER, without which the performances drop dramatically. The reason, as we conjecture, is that the reasoning task alone cannot give enough supervision for learning accurate word vectors and parameters of the RNN encoder. We note that NEURAL REASONER can still outperform Memory Net (N2N) with 10K data on both tasks.

When larger training datasets are available, NEURAL REASONER appears to prefer relatively deeper architectures. More importantly, although both tasks require two reasoning steps, the performance does not deteriorate with three reasoning layers. On both tasks, with 10K training instances, NEURAL REASONER with three reasoning layers and 3-layer DNN can achieve over 98% accuracy.

5 Conclusion and Future Work

We have proposed NEURAL REASONER, a framework for neural network-based reasoning over natural language sentences. NEURAL REASONER is flexible, powerful, and language indepedent. Our empirical studies show that NEURAL REASONER can dramatically improve upon existing neural reasoning systems on two difficult artificial tasks proposed in [7]. For future work, we will explore 1) tasks with higher difficulty and reasoning depth, e.g., tasks which require a large number of supporting facts and facts with complex intrinsic structures, 2) the common structure in different but similar reasoning tasks (e.g., multiple tasks all with general questions), and 3) automatic selection of the reasoning architecture, for example, determining when to stop the reasoning based on the data.

References

[1] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proceedings of EMNLP*, pages 1724–1734, 2014.

[2] B. Hu, Z. Lu, H. Li, and Q. Chen. Convolutional neural network architectures for matching natural language sentences. In *Advances in Neural Information Processing Systems 27*, pages 2042–2050, 2014.

[3] 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.

[4] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus. End-to-end memory networks. *CoRR*, abs/1503.08895, 2015.

[5] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112, 2014.

[6] J. Weston, A. Bordes, S. Chopra, and T. Mikolov. Towards ai-complete question answering: A set of prerequisite toy tasks. *CoRR*, abs/1502.05698, 2015.

[7] J. Weston, S. Chopra, and A. Bordes. Memory networks. *CoRR*, abs/1410.3916, 2014.

[8] M. D. Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.