ATLANTIS PRESS

Advances in Engineering Research, volume 119

2nd International Conference on Automatic Control and Information Engineering (ICACIE 2017)

Fast Associative Attentive Memory Network

Xiaomin Wang
Tongji University
Shanghai, China
1531845@tongji.edu.cn

Samuel Cheng
Tongji University & University of Oklahoma
Shanghai, China &Tulsa, OK, USA

***Abstract*—To solve the Cloze-style reading comprehension task, a challenging task to test the understanding and reasoning abilities of model, we propose a general and novel model called Fast Associative with Attention Memory Network in this paper. Unlike regular language model, we use fast weights to store associative memory for the recent past instead of activity hidden units which pay attention to the recent past. Our preliminary experiments indicate that our model outperforms regular RNN and LSTM.**

Keywords—Cloze Style, Fast Weights, Attention, Memory Network

I. INTRODUCTION

Natural language processing (NLP) is a synthetic subject synchronizing computer science, mathematics and linguistics. It's also an important research direction in the field of artificial intelligence. Reading Comprehension (RC) is a very challenging research topic in NLP area, which needs to answer questions correctly based on a given document. A good reading comprehension system can not only get useful information from given document, but also make sophisticated inference.

Recently, Cloze-style queries promote the development of deep learning techniques in machine comprehension. Cloze-style question is a problem that needs to infer the missing word in a question according to the given document. Hill et al. (2015) [1] released the Children's Book Test (CBT), where the first 20 sentences form documents and the 21st sentence form queries with a word removed. There are four types of the removed word need to be predicted, including named entities, common nouns, prepositions and verbs.

In order to solve the cloze style task, a lot of neural machine comprehension models are investigated (Hill et al., 2015; Weston et al., 2014; Hermann et al., 2015; Kadlec et al., 2016; Chen et al., 2016) [1,2,3,4,5]. Considering the varying length of the text sequences in reading comprehension tasks, Recurrent Neural Networks (RNNs) are more suitable than traditional statistics model. RNNs regard text sequences as time series, update them constantly and finally get the representation of the whole sequence. But simple recurrent neural network has propagating dependencies over long distances problem, it cannot make use of the long history information effectively. Existing works often use two improved models: the Long and Short Memory Neural Network (LSTM) [6] and the Gated Recurrent Unit (GRU) [7]. LSTM and GRU store recent information in hidden vector. If a hidden state has m units, the capacity of short-term memory for has only O(m) which forms a bottleneck for information flow.

In this paper, we propose a novel neural network, called fast associative with attention memory network (FAA). Motivated by the work of Ba et al. (2016) [8], we utilize weight matrix to store short-term information instead of activity hidden units which can pay attention to the recent past.

II. RELATED WORK

Hermann (2015) [3] released the CNN/Daily mail dataset, a large scale cloze-style reading comprehension dataset, where the questions are constructed by bullet point summaries. To handle this task, they proposed attentive and impatient readers model which apply LSTM to compute the representations of document and question in both forward and backward directions. We adopt the similar idea to present our document and question, but use different method to select correct answer.

Hill et al. (2015) [1] released another cloze-style dataset, The Children's Book Test (CBT), generated from raw story rather than summaries. Compared to CNN and Daily mail, model can make use of context information or prior knowledge to pay more attention to do understanding and inference task without any anonymous symbols in text in CBT. In order to test the ability of reading comprehension and reasoning of model, we evaluate our network on CBT task.

The attention mechanism is also widely developed in natural language processing task. Kadlec et al. (2016) [9] proposed the Attention-Sum (AS) Reader, which use two bidirectional GRU networks to encode document and question. Unlike the work of Hermann [3], AS Reader picked answer from the document in a simpler and more effective way, by which merely computing the sum of the probabilities of the same word appeared in the document. AoA Reader proposed by Yiming(2016) [10] used another new attention mechanism on the basis of AS Reader for the same cloze-style task and outperforms the state-of-the-art models.

Memory is also an important issue in NLP which provides data storage for long time. Weston [2] exploited external memory to store information. AS Reader and AoA reader all make experiments on GRU which store short memory in cells. Unlike LSTM and GRU, Ba et al. (2016) [8] used "fast weights" to store temporary memories of recent past and attend to the past instead of hidden units. It has been proved very helpful in sequence to sequence model, but the study has never been conducted on NLP task.

Our work is mainly inspired by Kadlec et al. (2016) [9] and Ba et al. (2016) [8]. We use fast associative memory model as

encoder to get the representation of text, then compute the probabilities of candidate answers via AS Reader.

III. FAST ASSOCIATIVE WITH ATTENTION MEMORY

A. *Task and dataset*

The task is to answer the cloze style question whose answer depends on the understanding of the relevant sentences and reasoning ability. We evaluated the effectiveness of our model on CBT. In Each CBT Sample, document is composed of 20 consecutive sentences and question is formed by the next sentence. There are four question types by removing four types of words: named entities (NE), common nouns (CN), prepositions (P) and verbs (V). In each question, the model is asked to select one correct answer given ten candidates.

The reading comprehension task can be treated as a quadruple <D, Q, A, a>. Where D represents the document, Q represents a question based on the content of document, A is a set of all candidate answers and the correct answer is a. In general, artificial cloze-style tasks are designed to satisfy the condition of $A \subset D$, ensuring all candidate answers appear in the document D.

Document:
1 At first, she wouldn't go without a great deal of coaxing, but after a while he didn't have to coax at all.
2 She seemed to delight to be with him as much as he did to be with her.
3 So Johnny Chuck grew happier and happier.
4 He was happier than he had ever been in all his life before.
5 You see Johnny Chuck had found the greatest thing in the world.
6 Do you know what it is?
7 It is called love.
8 JOHNNY CHUCK PROVES HIS LOVE These spring days were beautiful days on the Green Meadows.
9 It seemed to Johnny Chuck that the Green Meadows never had been so lovely or the songs of the birds so sweet.
10 He had forgotten all about his old friends, Jimmy Skunk and Peter Rabbit and the other little meadow people.
11 You see, he couldn't think of anybody but Polly Chuck, and he didn't want to be with anybody but Polly Chuck.
12 He had even forgotten that he had started out to see the world.
13 He didn't care anything more about the world.
14 All he wanted was to be where Polly Chuck was.
15 Then he was perfectly happy.
16 That was because Johnny Chuck had found the greatest thing in the world, which is love.
17 But Johnny still had one great wish, the wish that he might show Polly Chuck just how brave and strong he was and how well he could take care of her.
18 One morning they were feasting in a patch of sweet clover over near an old stone wall.
19 It was the same stone wall in which Johnny Chuck had escaped from old Whitetail the Marshhawk, when Johnny was a very little fellow.
20 Suddenly Polly gave a little scream of fright.
21 Johnny XXXXX looked up to see a dog almost upon her.

Candidate Answers:
Chuck|Johnny|Marshhawk|Meadows|Polly|Rabbit|Skunk|days|first|morning

Correct Answer:
Chuck

Fig. 1. A sample of CBT dataset.

B. *Model*

Representation of the document: we regard each document as an ordered sequence of word streams and each word in a sequence corresponds to a time step input of the encoder. Similarly, questions can also be dealt with in that way. Then we use a bidirectional fast associative memory model to encode the document. After bidirectional encoding, each hidden layer unit can fuse the semantics of the word itself and its context. The detailed procedure is described below (Fig. 2.).

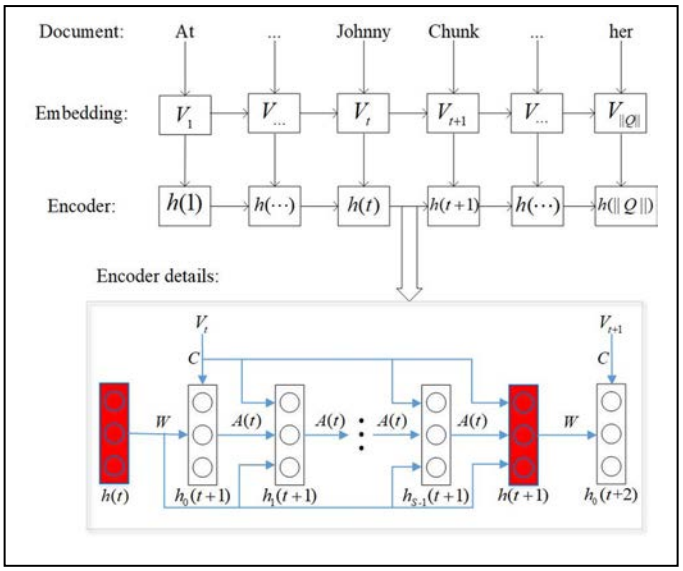

Fig. 2. Fast Associative Attentive Memory Network architecture.

In each time step, we input one word in document to our model. First, we can obtain the word embeddings V by one-hot and lookup table techniques where each row represents a word embedding V_t. Then we utilize fast associative memory model (show as "Encoder" in Fig. 2.) to encode input V_t.

Unlike RNN or LSTM, in this paper, we proposed an encoder that adds an inner loop with S steps between two hidden layers. Specifically, we can get hidden state $h(t+1)$ from last hidden units $h(t)$ after S time steps inner loop. In an inner loop, the initial vector can be calculated as $h_0(t+1) = f(Wh(t) + CV_t)$, where f is a nonlinear function used in each hidden state, W and C are weight matrix for hidden state and input respectively and we call them "slow weights". Then the next inner loop hidden state can be got as follows:

$$h_1(t+1) = f([Wh(t) + CV_t] + A(t)h_0(t+1)) \quad (1)$$

$$\cdots\cdots$$

$$h_{S-1}(t+1) = f([Wh(t) + CV_t] + A(t)h_{S-2}(t+1)) \quad (2)$$

$$h_S(t+1) = f([Wh(t) + CV_t] + A(t)h_{S-1}(t+1)) \quad (3)$$

$$h_S(t+1) \to h(t+1) \quad (4)$$

In particular, we provide matrix A in the loop as additional input to next hidden states and we call it "fast weights". In traditional neural network, each hidden state $h(t)$ is decided by two parts: (1) matrix C : represents the weight of input V_{t-1};

(2) matrix W : represents the weight of last hidden state $h(t-1)$. The matrix C and W are called "slow weights" because they are updated only after a batch finished.

In our paper, we adopt both slow weights and fast weights to determine the hidden state at the next time step. All weights will be updated fast when the new hidden state is brought in. We clarify the update rules of fast memory weight matrix A next.

$$A(t) = \lambda A(t-1) + \eta h(t) h^T(t) \tag{5}$$

Note that $A(0) = 0$ and $h(0) = 0$ at the beginning of the input sequence, λ is decay rate for the weights A and η is learning rate for hidden states.

After multiple recursion, we can get:

$$A(t) = \eta \sum_{i=1}^{i=t} \lambda^{t-i} h(i) h^T(i) \tag{6}$$

Beyond that, we adopt layer normalization after each inner loop to avoid gradient explosion.

$$h_S(t+1) = \frac{l(h_S(t+1) - \mu)}{\delta} + d \tag{7}$$

$$\mu = \overline{h_S(t+1)} \tag{8}$$

$$\delta = \sqrt{\overline{(h_S(t+1) - \mu)^2}} \tag{9}$$

We trained the model with learning rate *l*=0.5, decay rate *d*=0.95.

After encoding by our model, we can get representation of each word in document. We make use of bidirectional encoder (donated as F) to get full of context.

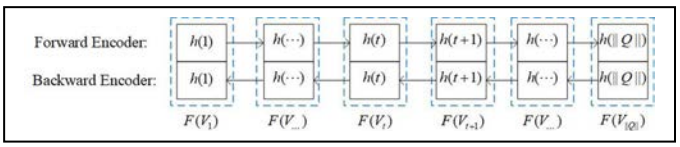

Fig. 3. Using bidirectional encoding to represent word in document.

The representation of word t in document is:

$$F(V_t) = \overrightarrow{h_t} \,||\, \overleftarrow{h_t} \tag{10}$$

$\overrightarrow{h_t}$ represents forward hidden layer state and $\overleftarrow{h_t}$ represents the backward hidden layer state. The symbol of "||" means vector concatenation.

1) Representation of the question: Similar to document representation, we use the bidirectional fast associative memory model to represent semantic information and context of each word in question. Instead, we connect the last hidden layer state $h_{\|Q\|}$ encoded by forward encoder and the first hidden layer state $\overleftarrow{h_1}$ encoded by backward encoder to represent the semantics of a question.

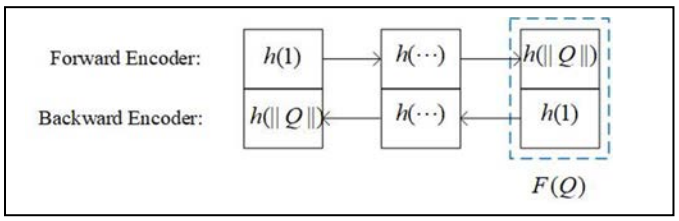

Fig. 4. Using bidirectional encoding to represent question.

$$F(Q) = \overrightarrow{h_{\|Q\|}} \,||\, \overleftarrow{h_1} \tag{11}$$

2) Match document with question: Matching function is used to compute the semantic matching of each word in the document and a question. Candidate answer with the highest matching score is chosen as the right answer. In this paper, we use dot product as matching function. Then matching score for each word are normalized by a softmax function. This process can be interpreted as attention.

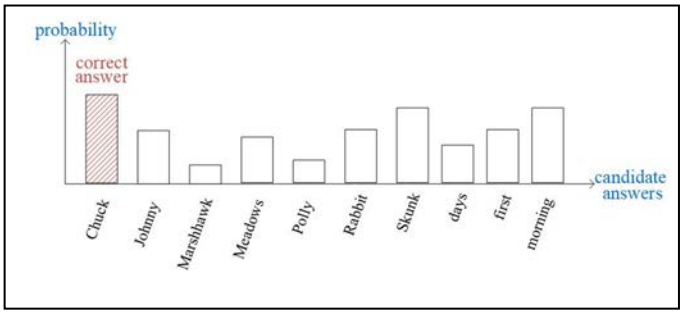

Fig. 5. Matching score between each candidate answer and question.

Matching score of each word V_t in the document is

$$p(V_t) \propto \exp(F(V_t) \cdot F(Q)) \tag{12}$$

Considering each candidate answer may appear many times in the document, we sum the matching score of this word as the probability of correct answer and choose the word with highest probability as the correct answer (*Fig.5*).

$$P(t \mid Q, D) = \sum_{V_t \in S(t,D)} p(V_t) \tag{13}$$

IV. EXPERIMENTS & RESULTS

A. Training Details

We use ADAM optimizer to update weights with learning rate of 0.00001 and cross entropy as loss function. We set the gradient clipping threshold to 10 and batch size as 32 during training. The number of hidden units (ehd) in each half of the bidirectional fast associative memory encoder is set to 128 and the source embedding dimension (sed) is set to 256. The embedding weights is initialized with uniformed distribution in [-0.1, 0.1].

In our fast weights encoder, we choose minimum 2 iteration steps in inner loop since we observed similar performance when adopting more iterations. The slow recurrent weights are drawn uniformly from $[-\sqrt{ehd}, \sqrt{ehd}]$, the fast weights initialize using identity scaled by 0.05. All bias initialized by constant 1. The learning rate and decay rate of fast weights are set to 0.5 and 0.95 respectively. In each hidden layer, we use ReLU activation. We implement our model using Tensorflow.

B. Results

The size of each dataset sample is about 100kb. Due to insufficient time, we only selected the first 1000 lines in each question types of train, valid and test dataset as our train, valid and test samples respectively. However, the preliminary result is promising. We compared our model FAA to bidirectional LSTM and GRU, the results are given in Table 1.

TABLE I. ACCURACY ON FOUR QUESTION TYPES OF CBT

	Named entities	***Common nouns***	***Verbs***	***Prepositions***
LSTM	0.34	0.18	0.24	0.38
GRU	0.32	0.27	0.22	0.42
FAA	**0.45**	**0.29**	**0.36**	**0.49**

It is clear that our FAA model achieves better performance than LSTM and GRU effectively for each question type. Especially for word types 'Named entities' & 'Verbs', we obtain about 12% and 13% improvement due to the fast associative with attention memory via fast weights between hidden layers. We can find that all models can be well predicted for prepositions, because this type is usually local contexts dependent and appears with high frequency. While the predictions on other types require a deeper understanding of the document that we need further study. More time will be needed to test the entire dataset, we plan to explore it further in the future.

V. CONCLUSION

In this paper, we present a novel neural network, fast associative with attention memory reader, for cloze-style reading comprehension task. We use fast weights to store associative memory for the recent past. The new hidden states are attracted to the recent hidden states, while the strength of the attention is determined by the scalar product of the recent hidden states certify by formula (5). The final predictions are made by summing all attentions. Preliminary experimental results indicate that our model outperforms regular RNN and LSTM.

To improve our model performance, we plan to combine with other attention mechanism and training strategy. Moreover, stability of the model could be an issue for larger training set size and we will address this in our future work.

ACKNOWLEDGMENT

The authors would like to thank Mr. Minxiang Ye for helpful discussion.

REFERENCES

[1] Hill F, Bordes A, Chopra S, et al. The Goldilocks Principle: Reading Children's Books with Explicit Memory Representations. arXiv preprint arXiv:1511.02301(2015).

[2] Jason Weston, Sumit Chopra, Antoine Bordes. Memory networks. arXiv preprintarXiv:1410.3916(2014).

[3] Hermann K M, Kocisky T, Grefenstette E, et al. Teaching machines to read and comprehend. Advances in Neural Information Processing Systems 2015, pp. 1693-1701(2015).

[4] Kalchbrenner N, Grefenstette E, Blunsom P. A convolutional neural network for modelling sentences. arXiv preprint arXiv:1404.2188(2014).

[5] Chen D, Bolton J, Manning C D. A thorough examination of the cnn/daily mail reading comprehension task. arXiv preprint arXiv:1606.02858(2016).

[6] Hochreiter S, Schmidhuber J. Long short-term memory. Neural computation9(8), 1735-1780(1997).

[7] Chung J, Gulcehre C, Cho K H, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555(2014).

[8] Ba J, Hinton G E, Mnih V, et al. Using fast weights to attend to the recent past. Advances In Neural Information Processing Systems 2016, pp. 4331-4339(2016).

[9] Kadlec R, Schmid M, Bajgar O, et al. Text understanding with the attention sum reader network. arXiv preprint arXiv:1603.01547(2016).

[10] Cui Y, Chen Z, Wei S, et al. Attention-over-attention neural networks for reading comprehension[J]. arXiv preprint arXiv:1607.04423, 2016.