Project Report

On

One Shot Memory Networks for Question Answering

Submitted by

Siddhanth Pillay - 15IT129

Samarth Mohan - 15IT236

Hrishikesh Thakkar - 15IT120

Under the Guidance of

Dr Sowmya Kamath

Dept. of Information Technology, NITK, Surathkal

Department of Information Technology

National Institute of Technology Karnataka, Surathkal.

May 2018

Certificate

This is to certify that the project entitled One Shot Memory Networks for Question Answering has been presented by , Siddhanth Pillay, Samarth Mohan, Hrishikesh Thakkar students of third year, B.Tech (IT), Department of Information Technology, National Institute of Technology Karnataka, Surathkal, on April 2018, during the even semester of the academic year 2017- 2018, in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Information Technology at NITK, Surathkal.

Place: NITK, Surathkal
Date: 30th April,2018

(Signature of Examiner)

Declaration

This is to state that the contents of this report are an instance of original work carried out by the following students as a part of the mini project assigned during the completion of the course titled Information Retrieval, course code - IT362, carried out under the guidance of Dr.Sowmya Kamath.

Siddhanth Pillay 15IT129
Samarth Mohan 15IT236
Hrishikesh Thakkar 15IT120

Abstract

The human mind is able to grasp the concepts of a new fact without the need of repeatedly iterating over the same fact.Hence we decided to incorporate one shot learning principles as it mimics the way humans process new facts. An added bonus is that since the training data is minimal it is feasible for anyone to use this model without the need of having a large amount of data for training. Dynamic Memory Networks make use of Gated Recurrent Units in order to implement the complex task of Question Answering over natural language. However Gated Recurrent Units require a lot of training data, hence we have implemented two modifications in the DMN model: We redefine the way input is processed, we reduce the data dependency by using SHTM, which is a neocortex inspired algorithm for one-shot text generation.

Contents

List of Figures

1 Introduction

Neural Networks have been used previously for a wide variety of applications such as image classification and text classification. However it is only recently that significant advances have been made to the application of Neural Networks in more complex tasks. This is only because of two concepts: memory and attention. These concepts are used in several complex tasks like reasoning over several facts written in natural language, machine translation and image captioning models.

Gated Recurrent Networks usually called GRUs are a special kind of RNN (Recurrent Neural Networks) capable of learning long-term dependencies. These networks work tremendously work well on a large number of problems.GRUs are specifically designed to avoid long-term dependency problem. GRUs are a big step in what we can accomplish using RNNs. GRU has Turing completeness in the sense that given enough network units it can compute any result that a conventional computer can compute, provided it has the proper weight matrix, which may be viewed as its program. This is a very useful property of the GRU that can be used to obtain the desired results.

The dynamic memory network(DMN) makes the use of GRUs to implement the memory component and the attention mechanism. This has yielded state of the art results on question answering, sentiment analysis and part-of-speech tagging.

In our implementation, we have made modifications to the input module and the memory module. We implemented a new input module which uses a two level encoder with a sentence reader and input fusion layer to allow for information flow between sentences. For the memory module we incorporated a module called episodic module that reduces data dependency and produces better results than GRU on One-Shot Learning tasks.

2 Literature Review

2.1 Background

Since dynamic memory network is related to two lines of work: memory and attention mechanism, we have analyzed both of them in our literature review

2.1.1 Memory Models

The earliest work in this category was presented by Weston et. al ,2015 [4] called Memory Networks. They have portrayed another class of learning models called memory networks. Memory networks prevail upon surmising parts joined with a long term memory segment; they figure out how to utilize these mutually. The long term memory can be perused and written to, with the objective of utilizing it for expectation. We research these models with regards to question answering (QA) where the long term memory successfully goes about as a (dynamic) information base, and the output is a printed reaction. We assess them on an expansive scale QA errand, and a smaller, yet more mind boggling, toy assignment produced from a simulated world. In the latter, we demonstrate the reasoning power of such models by chaining different supporting sentences to answer questions that require understanding the goal of verbs.

2.1.2 Attention Mechanism

Attention is basically a vector, often the yields of dense layer utilizing softmax function.

Before Attention component, interpretation depends on reading a total sentence and pack all data into a fixed length vector, as you can imagine, a sentence with many words represented by several words will without a doubt lead to information loss, insufficient translation, and so on.

However, attention in part settles this issue. It permits machine interpreter to investigate all the information the original sentence holds, then create the best possible word as per current word it deals with and the context. It can even enable machine translator to zoom in or out (center around local or global features).

Attention isn't baffling or complex. It is only an interface formulated by parameters and sensitive math. You could plug it anyplace you think that its reasonable, and potentially, the outcome might be change. Attention Mechanism is used in different applications such as Image Classification, Captioning of Images, Machine Translation etc. This concept has been incorporated in many architectures such as Neural Turing Machines, Neural GPUs, Stack-Augmented RNNs.

2.1.3 Question Answering in NLP

Question answering frameworks (QASs) produce answers of questions asked in regular languages. Early QASs were produced for confined spaces and have constrained abilities. Current QASs center around sorts of questions for the most part asked by users, characteristics of information sources consulted, and types of right answers generated. Research in the region of QASs started in 1960s and from that point onward, a substantial number of QASs have been created. To recognize the future extent of research around this area , the need of a thorough overview on QASs arises naturally.

Question Answering in NLP has been tackled using different approaches. If we have a very large text corpus, it is merely a task of information retrieval and extraction. More recent Deep-Learning approaches include Reasoning over a Knowledge Bases or directly via sentences for trivia competitions.

2.2 Identified Gaps

- In the proposed model, we are generating one word answers to a set of facts that are given as input and a question that is used as a query
- Integration of the efficient episodic memory module which optimizes the manner in which input facts are stored and processed
- Evaluation of the model against Memory Networks evaluated on babi-10k dataset

2.3 Problem Statement

"Building a Question-Answering Module that generates multi-word answers, if necessary, and reducing data dependency in comparison to existing architectures".

2.4 Objectives

- The input module converts the sentences into the vector space by making use of a distributed representation learning technique.
- Converting the question into the vector space by making use of the distributed representation learning technique.
- Embedding of input sentences and questions into the same space by the memory module by making the use of matrices

- Calculating the matching probabilities between the sentences and the questions by taking the inner product and making use of the softmax classifier
- Displaying the sum over input sentence representations weighted by the matching probability vector
- Facts and Question Vectors calculated are inputted in the episodic memory module
- Outer GRU state initialized with question vector
- Outer GRU generates the final memory vector working over a sequence of *episodes*
- Inner GRU generates episodes by passing over the input facts and considering the output of the attention function on some current fact
- Final state of the memory is fed into the answer module, which produces output by using a softmax classifier

3 Methodology

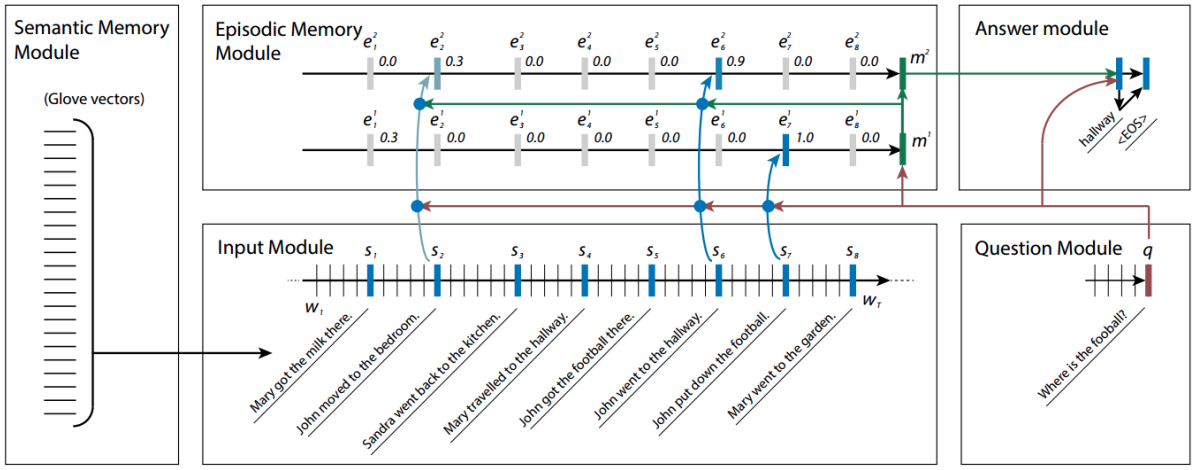

Figure 1: Dynamic Memory Networks architecture

The proposed model consists of four modules

1. Input Module
2. Episodic Memory Module
3. Question Module
4. Answer Module

We will elaborate on input module and episodic module more because the proposed implementation is different from the base paper's implementation.

3.1 Input Module

In the originally specified DMN, a single GRU processes all the words in a story and extracts representations by storing the hidden states produced at the end of sentence markers. It also allows for a temporal component because it allows the sentence to know the content of the sentences that appear before it. However this structure did not work very well for bAbI-10k. There are two main reasons for this:

1. GRU only allows sentences which came before it and not after
2. Supporting sentences can be very far from each other on a word level to allow interaction through word level GRU.

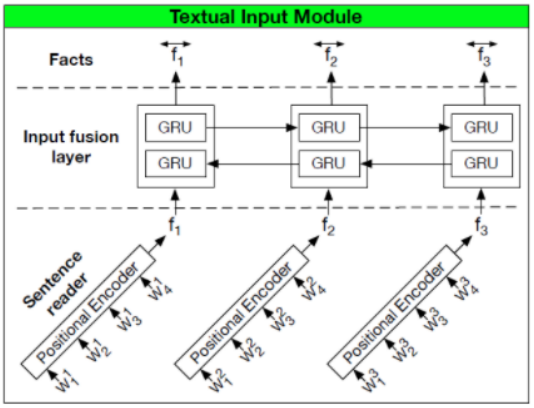

Figure 2: Modified Input Module

Hence it is proposed to replace the single GRU with two different components: The first component is a sentence reader which is responsible for encoding words into sentence embeddings. The second component is input fusion layer which allows for interaction between different sentences. In this component a bi-directional GRU is used, which allows information from both past as well as future sentences.

Since gradients did not need to propagate through the words between sentences the fusion layer allowed for distant supporting sentences to have a more direct interaction. For the positional encoding scheme, the sentence representation is produced by Equation 1 where o is element wise multiplication and l_j is a column vector with structure in Equation 2.where d is the embedding index and D is the dimension of the embedding. The input fusion layer takes these input facts and enables an information exchange between them by applying a bidirectional GRU as show in Equation 3.

$$\bar{f_i} = \sum_{M}^{j=1} l_j \circ w_j^i$$

Figure 3: Equation 1

$$l_{jd} = (1 - j/M) - (d/D)(1 - 2j/M)$$

Figure 4: Equation 2

$$\overrightarrow{f_i} = GRU_{fwd}(f_i, \overrightarrow{f_{i-1}})$$
$$\overleftarrow{f_i} = GRU_{bwd}(f_i, \overleftarrow{f_{i+1}})$$
$$\overleftrightarrow{f_i} = \overleftarrow{f_i} + \overrightarrow{f_i}$$

Figure 5: Equation 3

3.2 Episodic Memory Module

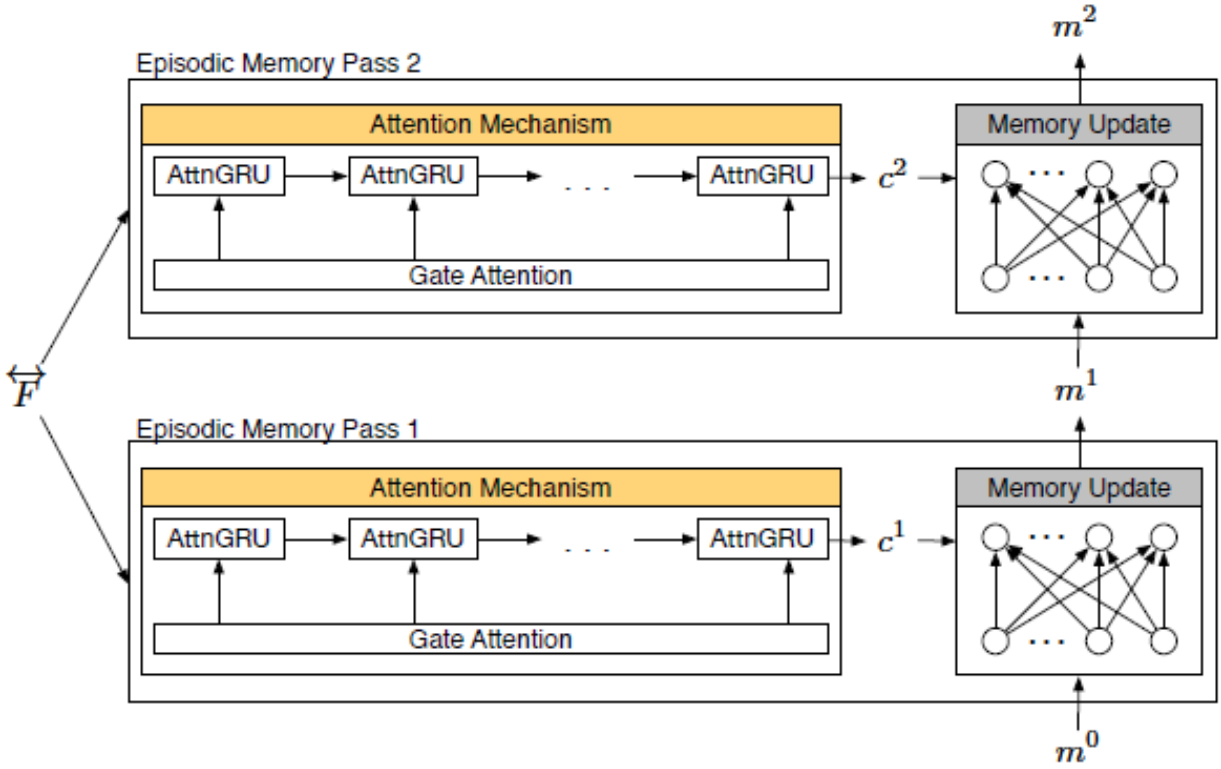

Figure 6: Episodic Memory Module

The episodic memory module, as depicted in Fig. 6, retrieves information from the input facts $\overleftrightarrow{F} = [\overleftrightarrow{f}1,...,\overleftrightarrow{f}n]$ and it focuses attention only on a subset of the facts provided. Attention is implemented by linking the attention gate g_i^t with each fact $\overleftrightarrow{f}_i$ during iteration t this value is attention gate value g_i^t by allowing interaction between the fact and question inner representation and episode memory state.

$$z_i^t = [\overleftrightarrow{f_i} \circ q; \overleftrightarrow{f_i} \circ m^{t-1}; |\overleftrightarrow{f_i} - q|; |\overleftrightarrow{f_i} - m^{t-1}|]$$
$$Z_i^t = W^{(2)} \tanh\left(W^{(1)} z_i^t + b^{(1)}\right) + b^{(2)}$$
$$g_i^t = \frac{\exp(Z_i^t)}{\sum_{k=1}^{M_i} \exp(Z_k^t)}$$

Figure 7: Equation 4,5,6

Attention Based GRU: We implement the Attention Mechanism by using an Attention based GRU. We modify the structure of the GRU by replacing update gate u_i with g_i^t, the value of the attention gate. Hence the GRU can now use the attention gate for updating its internal state.

$$h_i = g_i^t \circ \tilde{h}_i + (1 - g_i^t) \circ h_{i-1}$$

Figure 8: Equation 8

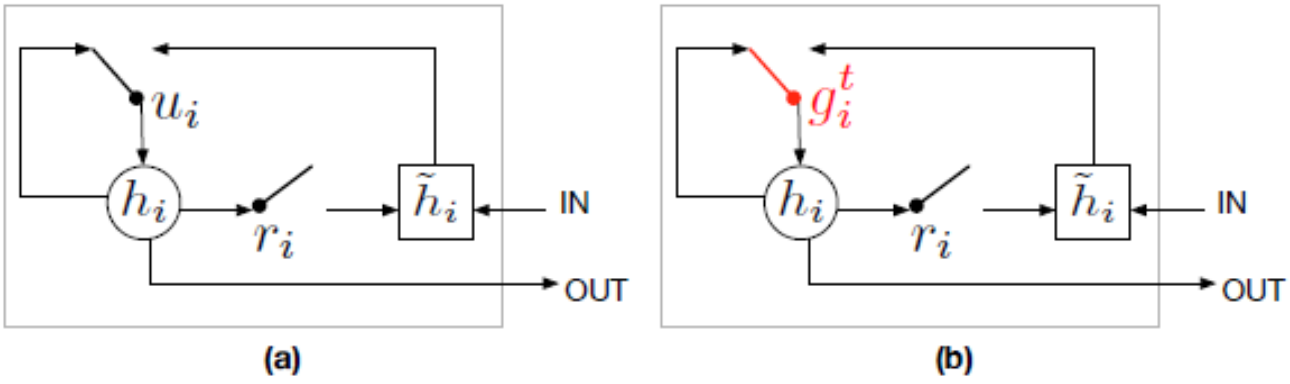

Figure 9: (a) Traditional GRU Model (b) Proposed Attention based GRU Model

Episode Memory Updates: After each iteration the previous memory episode m^{t-1} is updated with the newly generated contextual vector c^t producing m^tIn the DMN, a GRU with the underlying hidden state set to the question vector q is utilized for this reason. The episodic memory for pass t is processed by Equation 9. The final output of this layer is then passed to the answer module for answer generation.

$$m^t = GRU(c^t, m^{t-1})$$

Figure 10: Equation 9

3.3 Question Module

This module computes a vector representation q of the question, where q is the final hidden state of a GRU over the words in the question.

3.4 Answer Module

The answer module receives both q and m^t to generate the models predicted answer. For simple answers, such as a single word, a linear layer with softmax activation may be used. For tasks requiring a sequence output, an RNN may be used to decode a = $[q; m^t]$, the concatenation of vectors q and m T , to an ordered set of tokens.

4 Implementation

4.1 Work Done

We have implemented a dynamic memory network which consists of fours modules: input, question, answer and episodic memory.The base paper contains only a end to end encoder-decoder network.This is extremely limited as it cannot store long term memory, which means it cannot process facts that are temporally spaced apart . We have improved this by adding an episodic memory module inspired by the hippocampus area of our brain. All the implementations have been done in Python and Tensorflow. We have made use of the Adam optimizer as the optimizer.

In order to validate our work, we are making use of the bAbI dataset. bAbI dataset is a synthetic dataset released by Facebook AI Research in order to promote research in automatic text understanding and reasoning. The dataset consists of 20 different tasks. These tasks are of different nature and they evaluate different aspects of understanding of the given data. Tasks include single supporting fact answer, two or three supporting fact answer, yes/no questions, counting tasks, etc. The dataset comes in two different sizes, bAbI-1k and bAbI-10k. However, we have made use of bAbI-10k dataset.

4.2 Results and Analysis

We have trained all 20 tasks available in bAbI dataset.The following is the validation loss and test accuracy :

Task Id	Validation Loss	Test Accuracy
1	0.448	1.0
2	14.84	0.97
3	69.654	0.79
4	0.56	1.0
5	5.33	0.99
6	0.70	1.0
7	11.57	0.97
8	2.00	0.98
9	0.715	1.0
10	1.26	1.0
11	0.55	1.0
12	0.62	1.0
13	0.46	1.0
14	1.31	1.0
15	0.70	1.0
16	89.52	0.47
17	59.68	0.86
18	14.47	0.91
19	2.92	0.99
20	0.57	1.0

Task Id	ODMN	DMN+
2	0.36	0.03
3	0.42	0.21
5	0.001	0.01
6	0.357	0.0
7	0.08	0.03
8	0.016	0.02
9	0.033	0.0
10	0.006	0.0
14	0.036	0.0
16	0.551	0.53
17	0.396	0.14
18	0.093	0.09
20	0.0019	0.0

These were the results that were obtained by running the same tasks on the original DMN. Thus we see that our model performs significantly better than the original proposed model.

Task Id	E2E	DMN+
2	0.003	0.03
3	0.021	0.21
5	0.008	0.01
6	0.001	0.0
7	0.02	0.03
8	0.009	0.02
9	0.003	0.0
10	0.001	0.0
14	0.001	0.0
16	0.518	0.53
17	0.186	0.14
18	0.053	0.09
20	0.023	0.0

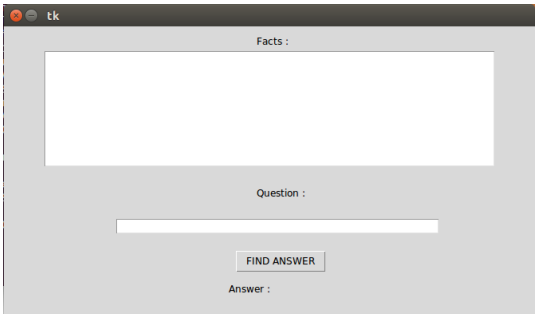

Figure 11: Demo of the CLI

These were the results that were obtained by running the same tasks on the End to End Memory Network. Thus we see that our model performs significantly better than a simple Memory Network.

4.3 Innovative Work

We have added an episodic memory module to the existing end to end model. The advantage of this allows us to infer the relationship between texts. This leads to higher accuracy in prediction.

4.4 Details of Individual Work

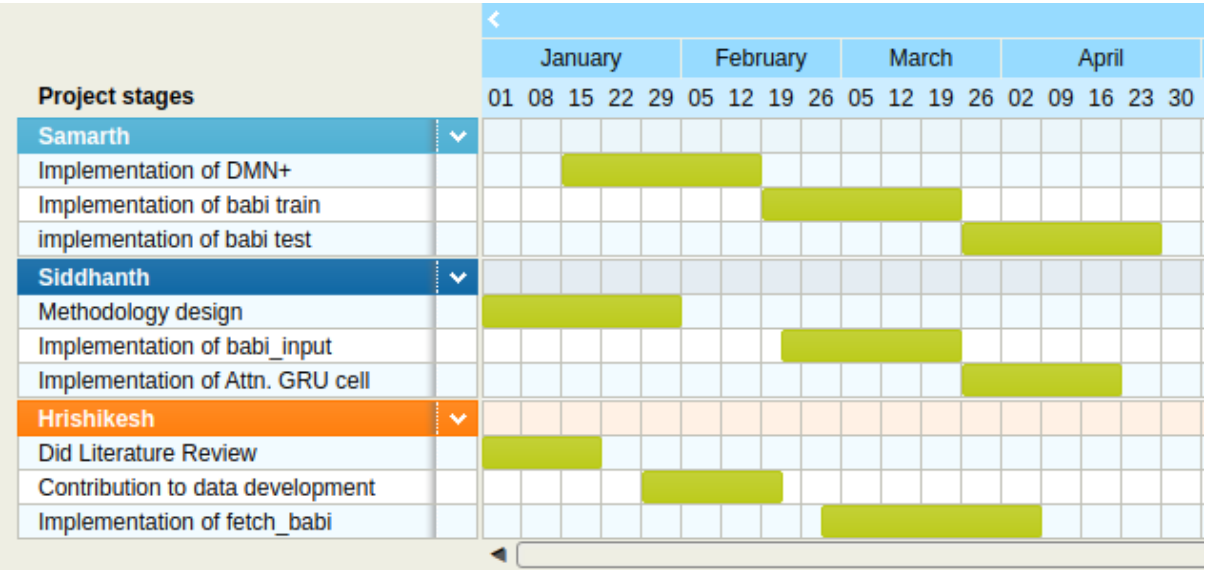

Figure 12: Gantt chart of individual contributions

5 Conclusion and Future Work

In conclusion our method shows that adding an external memory module increases the long term memory storage.This can be extremely useful in machine understanding architectures.The future would include extending this architecture from just natural language processing to image processing and understanding the events in an image.

References

[1] Caiming Xiong, Stephen Merity, Richard Socher Dynamic Memory Networks for Visual and Textual Question Answering

[2] Yuwei Wang, Yi Zeng, Bo Xu SHTM: A Neocortex-inspired Algorithm for One-shot Text Generation *2016, IEEE International Conference on Systems,Man and Cybernetics*

[3] Jeff Hawkins, Subutai Ahmad Why Neurons Have Thousands of Synapses, A Theory of Sequence Memory in Neocortex

[4] Jason Weston, Sumit Chopra, Antoine Bordes Memory Networks