Question Answering using Dynamic Memory Network

Anupam Asthana[#], Akshit Gupta[#], Akshay Kumar Singh[#], Ekant Bajaj[#], Nishu Bansal[*]

*Asst. Professor

[*, #] *Department of Computer Science, Ajay Kumar Garg Engineering College, Ghaziabad, U.P., India*

Abstract-- **Dynamic memory network has emerged as a next step towards the success of question answering. With synergy of Dynamic memory network [1] and Facebook bAbI dataset[7] , we aim at high performance result for the question answering task. The main focus of this project is to implement our own Dynamic Memory Network to achieve prominent results in question answering task. In Question Answering task, the Dynamic Memory Network takes input in form of story or paragraph through input module , processes the input and question asked and with episodic memory try to give the answer of the particular question.**

***Keywords*— Natural Language Processing, Question Answering, Gated Recurrent Unit, Recurrent Neural Network, Dynamic Memory Network.**

I. INTRODUCTION

One of the oldest problems in NLP is that of Question Answering (QA). Actually, most problems in NLP can be regarded as QA tasks. Furthermore, today QA systems have risen in popularity and much attention is being given to them, the most popular being Apple's Siri, IBM Watson, Microsoft's Cortana and Amazon's Alexa. In its early days, QA systems consisted of structured knowledge databases developed by experts that usually catered to a specific domain. Here, questions asked in natural language are parsed and converted into machine understandable queries and that would further generate an appropriate answer. Nowadays, there is an extremely huge amount of natural language information on the internet. Hence, modern QA systems mainly focus on information extraction from these documents. As a result, these QA systems essentially have information retrieval methods such as

- A question processing module for formulating a query.
- An information-retrieval module for selecting the appropriate document and passage.
- An answer processing module to generate the suitable answer in the required language.

Much of these systems of domain-neutral, i.e. they can answer questions from various topics. With the recent breakthroughs in deep learning, papers have been published that demonstrate the utilization of recurrent neural networks for question answering tasks. These networks spawn underlying representations of natural language texts rather than depending upon parts of speech tagging, parsing, named entity recognition and other extracted features. These networks show much less pre-processing overhead and have recently matched and even surpassed the results of other models. The current state-of-the-art model is Dynamic Memory Networks which contains 4 modules: input, episodic memory, question, and answer. Each module consists of a recurrent neural network optimized for the corresponding sub-task.

II. RELATED WORK

Prior to Dynamic Memory Networks, the question answering task got good results using Memory Network on bAbI dataset by Weston et al. [4]. Memory network was presented as a way to use a long-term memory component as a dynamic knowledge base for question answering. Memory networks have multiple components: an input component, response component, generalization component, and output feature map. The generalization component and output feature map, which also aim to iteratively retrieve facts from the set of input facts, are functionally replaced by the episodic memory module in DMN. Memory Networks also require supporting facts during training. Along with memory mechanism work had been done related to attention mechanism. They have been recently used in various applications like image captioning, text translation, speech recognition etc. Dynamic Memory Networks also use attention mechanism for question answering to iteratively focus on certain sentences in the input text.

Dynamic Memory Networks gained prominence with Kumar et al's [1] publication in 2015. They present the Dynamic Memory Network model by applying it to a variety of language tasks including the Facebook bAbI dataset. This model iterates over facts represented as distributed vectors, computing a gate for each fact to determine whether or not the fact is relevant to the reservoir of knowledge the module has already built up in prior iterations, known as a "memory" in the paper.

The facts and their computed gates are fed through a specialized weighted Gated Recurrent Unit to compute an episode, which is then used to compute the next memory. The purpose of the model is to iteratively retrieve more and more information relevant to the original query, using newly discovered knowledge found in previous iterations to inform

on what facts should be considered important for the next iteration.[1] This model was able to achieve state of the art results on many tasks in the bAbI dataset offered by Facebook Research.

Xiong et al [3] showed that Dynamic Memory Networks receive strong results when supporting facts are not marked during training, proposed improvements on the memory and input modules, and illustrated that the models do well for visual question answering in addition to textual question answering.

III. DATASET

3.1. *Facebook bAbI Dataset*

The Facebook's bAbI-10k dataset is commonly used as a benchmark in many question answering papers. It consists of 20 tasks. Each task has differing types of questions such as single supporting fact questions, two supporting fact questions, yes no questions, counting questions, etc.

The English version of the dataset was used with 10,000 training examples and 1000 test examples. All examples consisted of an input-question-answer tuple. The input is a text passage of variable length. The type of question and answer depends on the given task. [7]

For example, some tasks have yes/no answers while others focus on positional reasoning or counting. The dataset also provides the line numbers of the input relevant to the answer, for each question-answer pair. Every answer in the bAbI dataset is one word. Some examples from the dataset are as follows:

Two supporting fact example	**Yes/no question example**
1 Mary got the ball there.	2 Jack moved to the roof.
2 Jason went to the roof.	3 Is Jack in the kitchen?
no 2	

3 Chester went back to the kitchen.
4 Mary travelled to the hallway.
5 Where is the milk? **hallway 1 4** [7]

IV. GATED RECURRENT NETWORK

Gated Recurrent Networks form the basis for many of the modules, and so we describe them in detail. Assume we have an input x_t and hidden state h_t at each time step t. Then, the GRU(Gated Recurrent Unit) can be described with the following equations:

$$z_t = \sigma(W^{(z)}x_t + U^{(z)}h_{t-1} + b^{(z)})$$
$$r_t = \sigma(W^{(r)}x_t + U^{(r)}h_{t-1} + b^{(r)})$$
$$\tilde{h}_t = \tanh(Wx_t + r_t \circ Uh_{t-1} + b^{(h)})$$
$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \tilde{h}_t$$

where $W^{(z)}, W^{(r)}, W \in \mathbb{R}^{n_H \times n_I}$ and $U^{(z)}, U^{(r)}, U \in \mathbb{R}^{n_H \times n_H}$. The n dimensions are hyperparameters (n_H being hidden size and n_I being input size) [5]

GRU is shown in Figure 1 below.

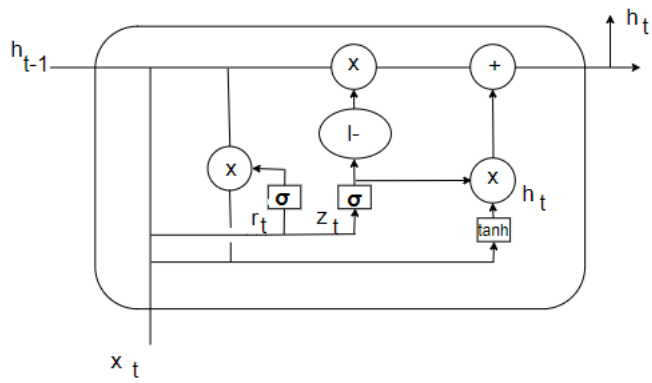

Figure 1: Gated Recurrent Unit [8]

V. DYNAMIC MEMORY NETWORKS

The Dynamic Memory Network model consists of 4 modules: input, question, episodic memory, and answer. This section explains the functionality of each of these modules. It is shown in Figure 2.

5.1 Input Module

The input module takes word vectors as input and feeds them through a GRU and outputs the hidden states at the end of each sentence for the episodic memory module for reasoning. More formally for a sequence of T_I words $w_1,...,w_{T_I}$ we update the state using

$$h_t = GRU(L[w_t], h_{t-1}) \quad [1]$$

Then say the T_I words consist of T_S sentences $s_1,...,s_{T_S}$. We then project the hidden states corresponding to the end of each sentence. So, the final output of the input module is $h_{s_1},...,h_{s_{T_S}}$

5.*2 Question Module*

The question module also runs a GRU on word vectors, however it just provides the final state of the GRU to encode the question. So, for a question of T_Q words $w_1,...,w_{T_Q}$ we update the state using

$$h_t = GRU(L[w_t], h_{t-1}) \quad [1]$$

The final output of the question module is h_{T_Q}.

5.3 Episodic Memory Module

The episodic memory module reasons over the sentence states from the input module as well as the question state from the question module and ultimately produces a final memory state that is forwarded to the answer module to generate an answer.

- *Episode Update Mechanism*

Each episode reasons over the sentences and produces a final state for that pass over that data. It is updated for a new input sentence state c_t as given below:

$$z_t^i = [c_t, m, q, c_t \circ q, c_t \circ m, |c_t - q|, |c_t - m|]$$

$$Z_t^i = W^{(2)} \tanh\left(W^{(1)} z_t^i + b^{(1)}\right) + b^{(2)}$$

$$g_t^i = \frac{\exp(Z_t^i)}{\sum_{k=1}^{M_i} \exp(Z_k^t)}$$

$$h_t^i = g_t^i GRU(c_t, h_{t-1}^i) + (1 - g_t^i) h_{t-1}^i$$

[6]

So, the current sentence state c_t, the current memory state m, and the question state q are used together to determine if the current sentence is significant or not to the answer and encoded in g_t^i.

We see that if $g_t^i \approx 0$ then the previous state will be copied through and the sentence will be ignored, but if $g_t^i \approx 1$ the past will be ignored and a lot of attention will be placed on the current sentence.It is also important to note that we use the softmax function to determine the value of g_t^i to enable the attention to be visualized more easily, since it forces the sum of all attention gates to be 1 [3].

The final state for the episode is the state of the GRU after all the sentences have been seen is

$$e^i = h_{T_S}^i$$ [1]

Memory Update Mechanism

The memory is then updated using the current episode state and the previous memory state.

$m^t = GRU(e^t, m^{t-1})$ [6]

The final state of the memory after the maximum allowed passes over the data is then sent to the answer module to generate an answer.

6.4 Answer Module

The answer module is a simple linear layer with a softmax activation to produce a probability distribution over the answer tokens. This may be extended to an RNN for multiword answers, however, since the bAbI dataset has only one word answers so a simple softmax is implemented. [1]

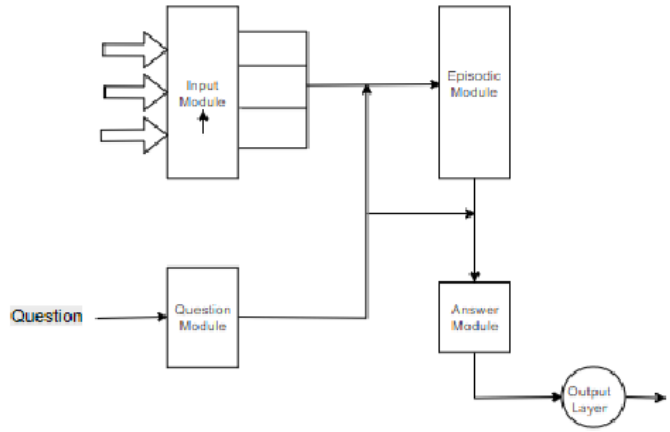

Figure 2: Dynamic Memory Networks

VI. CONCLUSION

The purpose for this paper was to explore Dynamic Memory Networks which is an improvement on the Memory Networks, and also to find methods to achieve greater accuracy on bAbI question answering tasks. We can certainly conclude that DMNs have achieved superior performance presumably due to using sequence models for each module of the entire model, helping it better capture position and temporality of natural language. The future scope of Dynamic Memory Network could be to put in additional layers to Gated Recurrent Unit, implement a bidirectional Recurrent Neural Network as well as its performance on more diverse datasets such as DeepMind.

REFERENCES

[1]. Kumar, Ankit et al. (2015) . Ask me anything: Dynamic memory networks for natural language processing. arXiv preprint arXiv:1506.07285.

[2]. Hermann, Karl Moritz et al. Teaching Machines to Read and Comprehend. https://arxiv.org/abs/1506.03340

[3]. Xiong, Caiming et al. (2016) Dynamic Memory Networks for Visual and Textual Question Answering. https://arxiv.org/abs/1603.01417

[4]. Wetson, Jason et al. (2015) Memory Networks. ICLR 2015. https://arxiv.org/abs/1410.3916

[5]. Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On ¨ the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259, 2014**.**

[6]. Raghuvanshi,Chase- Dynamic Memory Networks for Question Answering https://cs224d.stanford.edu/reports/RaghuvanshiChase.pdf

[7]. Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. Towards ai-complete question answering: A set of prerequisite toy tasks. arXiv preprint arXiv:1502.05698, 2015

[8]. http://colah.github.io/posts/2015-08-Understanding-LSTMs/