ISSN:0975-9646
Vinay Chandragiri/ (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 7 (5) , 2016, 2231-2234

Intelligent Question Answering System

Vinay Chandragiri[#1]
[#]*Department of Computer Science and Engineering, IIT Guwahati*
Guwahati, Assam, 781039, India

***Abstract*— In this work we aim to develop an Intelligent Q & A system, which takes in a fact database as input and answers questions of varied range of complexity. Our work is di- vided into two steps, initial one being the implementation of Dynamic Memory Networks (DMN) that passes the face- book bAbi tasks. Next step is to use this model to evaluate on data set released for Allen Institute ARISTO challenge which contains science questions of increasing difficulty from 8th standard examination. We implemented a machine that takes in the input facts and questions and converts them into computable format so as to search for a specific answer based on our question. These above mentioned tasks help us in evaluating whether our system is able to answer questions ranging from simple deduction like yes/no or answers that can be deduced based on single supporting fact to largely complex ones that need chaining of facts and indefinite knowledge. We use data sets from facebook bAbi tasks as knowledge base for training purposes.**

***Keywords*—Q & A system, DMN, bAbi tasks.**

I. INTRODUCTION

Interesting aspect of NLP is that most of the tasks in it can be modeled into a Q & A model. For example, if your problem of interest is sequence modeling task (NER), then the way to model it would be by giving the data as input and formulating a question What named entity tag sequence suits the best?. Similarly, if it is a translation problem then asking a question What is the translation in Dutch? will do our task. Hence developing an intelligent Q & A system in NLP is of great importance. Developing a system that consistently performs well on questions ranging from straight forward knowledge based yes/no type to the ones requiring deduction from 3-4 arguments is a tedious task.

II. BABI TASKS

- Case 1 : In this a single fact is sufficient among other irrelevant facts to deduce the answer. Ex: Jane had milk. Vinay went to sleep. Who had milk?
- Case 2 : This type of task needs 2 to 3 facts together to arrive at a conclusion. To use multiple facts together it's important to realize the components of the facts effectively. Ex : Reena opened the door lock with key. Reena went into kitchen. Abhishek entered the home.

Where is the key?

- Case 3 : These are the cases where bag of words representations fail. Reordering of words in sentence is done and hence it is very important for our system to capture this information effectively. Ex : Which pillar is to the south of bedroom? and Bedroom is to the south of which pillar? Both of the questions contains same set of words but have a different meaning all together.
- Case 4 : This again is a simple task, but has a fixed output set of yes or no. Does Vasu drink? Here the answer is either a yes or no.
- Case 5 : This handles all the counting cases, like How many times has Vinay shouted? or handling an out- put using multiple linked statements like, Vinay ordered food. Vinay shouted in lab. Vinay slept off after writing the code. Q : What is Vinay doing now?

Based on the above mentioned cases, 20 tasks known as facebook bAbi-tasks[4] have been developed to evaluate the progress of a model. Once we develop our model we go on to test them on it. For a system to perform the above mentioned tasks, it must possess the ability to understand the meaning of the facts given, along with the ability to retrieve proper relevant data according to the question asked. In some cases it might have to move a step further and reason over the facts to deduce the answer.

Neural networks have shown reliable outputs in text classification tasks. But they have exhibited average performance when it comes to increasingly complex tasks like facebook bAbi-tasks. LSTMs have only passed 4/20 tasks in them. Although LSTMs exceed the performance of RNNs in sequence prediction and tasks similar to above discusses, they fail to perform well here because of their weakly supervised nature over answers. They work through running over the input data till the required answer is obtained. However, recently there has been an immense progress in this area after neural networks with attention model and episodic memory model (DMNs) have been introduced. As far as ARISTO challenge data is concerned, main difficulty lies in the understanding of the question clearly and outputting the most probable answer to it. Neural network based methods have made phenomenal progress in image and text classification.

However, only recently there has been an immense progress on complex tasks that require logical reasoning. The reason behind this success is due to the addition of memory and attention components to complex neural networks. Efficient working of question-answering model requires certain reasoning capabilities which are exhibited by the Neural net- work architectures with memory and attention mechanisms.

III. MEMORY NETWORKS

Memory networks are a recent class of neural networks that are pretty hot in nlp right now because of their excellent performance on reasoning tasks and their ability to do multiple tasks by end to end training on a single model.

Babi is basically a set of toy questions any AI system should be able to answer. For instance, you can give a system a set of input facts like:
Mary went to the bathroom. John went to the hallway. Mary travelled to the office. The system needs to reason over these facts and answer the questions over them, just like a human would. So you can ask the system:

Q: Where is Mary? A: Office
Memory networks generally perform pretty well on these 20 toy tasks, scoring just a 6.6 % mean error rate. If you compare it to LSTMs who have a 36.4 % mean error rate, it is pretty amazing.

Moreover LSTMs fail on 16/20 of these babi tasks whereas a carefully trained Memnet just gets 4/20 of these tasks wrong. Talk about being an overachiever!)

If you also happen to read up on their extension, dynamic memory networks, they have used memory networks to do various nlp tasks in the form of memory networks. For example:

I: Everybody is happy. Q: What is the sentiment? A: positive Q: What are the POS tags? A: NN VBZ JJ I: The answer is far from obvious Q: In French? A: La reponse est loin d'etre evidente.

Is this not incredible. In a single model, we have done sentiment analysis, POS tagging, machine translation. Moreover the model achieves state of the art results on all of these tasks.

How does Memory Networks work ?
To answer this question, let's consider first how a human would answer questions like these. For example, consider the following task on inductive reasoning:

Lily is a cthulhu. Lily is white. Bernhard is green.
Greg is a cthulhu.

Q: What color ir Greg? A: white
If given a set of input facts to memorize and then asked questions on, how would the human brain go about reasoning to obtain the final answer?

Greg.. Greg.. I remember reading something about Greg. Oh wait wasn't Greg a cthulhu?

So here we have retrieved our first piece of critical information that greg is a cthulhu. Now since we obviously don't know what a cthulhu looks like, we would go like:

Greg.. cthulhu.. color.. There was something else I remember about cthulhu. What was it? hmm. Wasn't Lily a cthulhu too?

Now we have retrieved two critical memories: Greg is a cthulhu, Lily is a cthulhu. But it is still not enough to answer we question about the color of Greg, so you go about fetching memories again.

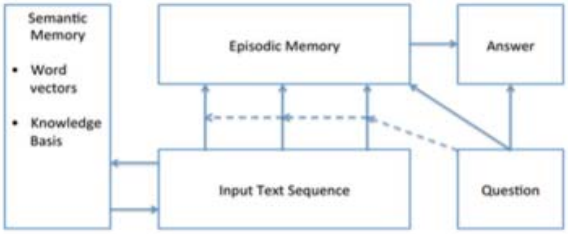
Greg.. Lily.. cthulhu.. color.. Can't think i remember anything else about Greg. hmm.. what about Lily. What else do i remember about Lily. Lily.. color.. Wasn't Lily white?

Awesome !! Now we have retrieved all three necessary memories to reason about the color of Greg. Now all our brain needs to do is perform inductive reasoning.

Greg is cthulhu, Lily is a cthulhu, Lily is white -> Greg is most probably white.

And this is what memory networks do exactly as well. They maintain a vector representation of memories (input facts). Then convert the question into a vector form as well. Then they try to find the first best memory which matches the question (by a cosine dot product between vectors)

IV. DYNAMIC MEMORY NETWORKS



The DMN is a neural network architecture that takes in relevant input data and creates episodic memory units over it. When questions are given in, they go on to trigger an attention process that runs iteratively for a definite no.of times which helps the model to focus its attention on the required part of input and figure out the relevant output. DMN working in a nutshell can be put forward as, "First all the facts and questions are taken in and a proper representation is generated for them. The question representations are then used as input triggers to the iterative attention module that helps the module to retrieve all the facts that are related. Then these facts obtained are put into the memory module which then takes up the task of generating vector representations to all these facts through proper reasoning. These vector representations are sent to the answer module which outputs the final answer." It consists of the following modules:

Input Module
This module takes in the provided input facts into the model and processes them into a distributed vectors. This is used later on by episodic memory module to iterate over. In NLP input sequence is taking as a sequence of words. In this module we process the input data and convert it into an ordered set of vectors termed as facts $F = [f_1 , f_2 , ..., f_N]$ where N is the total no. of facts. In case of a single fact input, output size is equal to the no.of tokens in input fact. The gate recurrent units (GRU) are used to process all the words in the text and extract the sentence form by

remembering the hidden states produced at the end of sentence markers. It allows sentences to know the content of previous sentences and thus provides a temporal component. For each time step i with input x_i and previous hidden state h_{i-1} ,we compute the updated hidden state h_i = RN N (x_i , h_{i-1}) by

$u_i = \sigma(W^u x_i + U^u h_{i-1} + b^u)$) (1)

$r_i = \sigma(W^r x_i + U^r h_{i-1} + b^r)$) (2)

$h_i = \tanh(Wx_i + r_i \circ Uh_{i-1} + b^h)$) (3)

$h_i = u_i \circ h_i + (1-u_i) \circ h_{i-1}$ (4)

where $\circ$ is the element wise product, σ is the sigmoid activation function, $W^z, W^r, W \varepsilon R^{nH \times nI}$,$U^z, U \varepsilon R^{nH \times nH}$,n_h is the hidden size and n_i is the input size.

Question Module

This module takes in the questions and gives them the distributed vector representations. The vector representation for each question forms the initial value for episodic memory module iterations. Similar to input module, here questions are processed using a gated recurrent network. At every time step the hidden state parameters are updated according to the equation x_t = GRU(E(i_t),x_{t-1}). The final hidden state $x_f \varepsilon R^{nH}$ of the GRU is computed over the text, where x is the vector representation of the question.

Episodic Memory Module

The output representations given by input module is used by this module to iterate over. This process of iteration is done in a constructive manner where each run uses the output of previous run episodic memory memory(i − 1) along with the question module output q and the fact representations F to generate the new episode for the current run episode(i). The aim of this module is to gather all the information that is essential to answer the question q by using the input facts. Multiple iterations over the input is done to improve our understanding of question and answer. The episode memory on the t^{th} pass over the input is referred to as m^t, where $m^t \varepsilon R^{nH}$, and initial value at t=0 is set to the question vector q. It contains a attention mechanism where we produce a contextual vector c^t , where $c^t \varepsilon R^{nH}$ is the collection of all relevant input for pass t with previous memory m^{t-1} and the memory update mechanism in which

we generate the memory m^t by using c^t and m^{t-1} such that at final pass T we have m^T which contains all information essential to answer q. This m^T is given to the answer module to extract the answer.

Attention Mechanism

Each time of iteration i, the model takes a fact c, question q and previous memory, memory(i-1) and computes a score S(c, memory(i-1), q). S is a function of two-layered feed forward neural network. In cases where we know a particular fact has more importance, then cross entropy function can be used to reach the best solution.

Answer Module

This module uses the value of q and m^T to get the models final answer. Again a GRU is used in this module. The first state is initialized to the end result of memory module. At each state the answer is concatenated with the question. This and the last input of the hidden state together are given as inputs to calculate the new hidden state value. The output obtained is trained with the cross entropy error function to achieve the optimal result.

V. RESULTS

We have run our implementation on data testing for 20 bAbi-tasks. Results are as follows:

- Task 1 – Single supporting facts: Answers that require knowledge from single fact, placed amongst other irrelevant facts. Accuracy: 98.60%, Loss: 0.0394
- Task 2 – Two supporting facts: Answers that require two facts which are placed among other facts which are of no use. Accuracy: 78.60%, Loss: 0.5556
- Task 3 – Three supporting facts: Answers that require knowledge from three facts which are placed among other facts which are of no use. Accuracy: 84.20%, Loss: 0.4958
- Task 4 – Two argument relations : Answers that require the model to detect the relation between any two entities in our facts. This is the example where bag of words approach fails. Accuracy: 97.50%, Loss: 0.0802
- Task 5 – Three argument relations : Exactly similar to above task, but with three arguments now. Accuracy: 98.90%, Loss: 0.06997
- Task 6 – Yes/No : The model needs to answer in one word, either a yes or no for the question asked. Accuracy: 99.20%, Loss: 0.01893
- Task 7 – Counting : These questions generally target the systems intelligence to determine a specific count value, i.e that involves keeping track of multiple facts. Accuracy: 87.40%, Loss: 0.5198
- Task 8 – Chain of inferences : This involves connecting of multiple tasks efficiently and extracting knowledge of where what is in the current state. Accuracy: 95.30%, Loss: 0.2115
- Task 9 – Detecting negative sentiments : When asked about a particular thing, the system is expected to detect the tone of the entity that is in attention and reply accordingly. Accuracy: 98.00%, Loss: 0.0762
- Task 10 – Unsure or Incomplete knowledge : This is when the system has to work with indefinite information. Accuracy: 95.40%, Loss: 0.2216
- Task 11 – Co-referencing : Instances of co-referencing are to be dealt with. System has to figure out what is being talked about every particular entity using co-referencing. Accuracy: 85.30%, Loss: 0.8618
- Task 12 – Conjunction : This is when simultaneous

action of multiple subjects are occurring and the system has to deal with it and keep track of who is doing what. Accuracy: 96.00%, Loss: 0.1144

- Task 13 – Co-referencing with multiple subjects : Same as task 11 but a bit more complex now because of multiple subjects coming into picture. Accuracy: 92.40%, Loss: 0.3715
- Task 14 – Time reasoning : This involves realizing the time periods in which specific events have occurred and there by predicting the sequence of their occurrence. Accuracy: 98.30%, Loss: 0.0487
- Task 15 – Deduction : This generally involves matching of a specific entity with its class, based on deduction from some specific facts. Accuracy: 98.20%, Loss: 0.0704
- Task 16 – Induction : Simple induction on entities involved. Accuracy: 65.90%, Loss: 1.6947
- Task 17 – Position detection : Based on the given facts, the system is supposed to predict the position of objects of interest. Accuracy: 63%, Loss: 1.7996
- Task 18 – Size reasoning : The ability of the system to compare two objects size and give a judgement is evaluated in this case. Accuracy: 96.60%, Loss: 0.1244
- Task 19 – Path Finding : When asked for, the correctness of the directions given by the model between two points is evaluated. Accuracy: 45.60%, Loss: 2.1480
- Task 20 – Agent's motivation : Here the system outputs a reason for a particular objects action or an incident that have occurred. Accuracy: 99.30%, Loss: 0.0201

Total Accuracy: 88.685000009834766

DMN has shown a decline in performance in tasks 2 & 3 in comparison with Memory neural networks using N-gram method. These both tasks share the common feature of having long input sequences which might be the reason for the difficulty as a recurrent model is used in DMN and very long inputs make it complex to perform. Whereas tasks 7 & 8 have improved a lot and this confirms the strength of multiple iterations and episodic memory module. No significant improvement has been shown in case of tasks 17 & 19.

REFERENCES

[1] Jason Weston, Antoine Bordes, Sumit Chopra. Toward AI-complete question answering. Facebook AI Research

[2] Caiming Xiong, Stephen Merity, Richard Socher. Dynamic Memory Networks for Visual and Textual Question Answering. [http://arxiv.org/pdf/1603.01417v1.pdf].

[3] Ankit Kumar, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, Richard Socher Dynamic Memory Networks for Natural Language Processing. MetaMind, Palo Alto, CA USA

[4] J Weston, A Bordes, S Chopra, Alexander M. Rush, Bart van Merrienboer, A Joulin & T Mikolov. Towards AI- Complete question answering : A set of prerequisite toy tasks Facebook AI research, New York, USA