Re-implementation of Memory Networks

Aykut Aykut
Koç University
aaykut13@ku.edu.tr

April 30, 2017

Abstract

In this paper, I describe one of the learning models called memory networks. I will show the power of memory networks in the context of Question Answering. Memory networks are able to use related sentences in order to answer questions. They infer from sentences, chain different sentences, use induction and deduction in order to answer questions. Memory networks have an external memory and this memory is used as a knowledge base, and the output is textual response to the questions. Memory networks are tested with 20 different types of questions which require understanding of the sentences and the questions and memory networks are able to answer questions about each type.

1 Introduction

In this paper, the results of Memory Networks work of Weston et al. is replicated. Memory networks have hidden states and weights. The aim is to make machine learn the meaning and inference of sentences by training it with fully supervised learning. After the training, the machine starts to infer meaning from a given story and become able to answer questions about that particular story. The machine memory can be read and written.

Memory networks have a larger storage compared to other approaches which tries to solve same kind of Question Answering problems. RNNs are also used for solving question answering problems. They are trained and then they predict the words for a given question. However, memory of RNNs is not enough to remember facts from past accurately. On the other hand, memory networks are able to write to and read from an external memory to remember the past. The model is trained fully supervised techniques in order to use its memory effectively. I introduce the model in Section 2, describe the experiments in Section 3, analysis and contribution in Section 4. I discuss related work in Section 5 and finally conclude in Section 6.

2 Model

2.1 Model Architecture

A memory network has 4 components I, G, O and R as follows:

I (input feature map): Takes the input, in this case lines in a text, and returns its feature representation.

G (generalization): Takes the feature representation of a line and adds it to the memory.

O (output feature map): Takes the feature representation of a question, finds all of the related memory states with the question and returns them.

R (response): Takes both the feature representation of the question and the related memory states and finds the answer of this question among the dictionary.

The following is the flow of a given a sentence x inside the model:

1. *x_feature* = I(x) converts x into the internal feature representation.
2. G(*x_feature*, *memory*) updates the memory by adding the *x_feature* to the given *memory.*
3. *output_feature* = O(*question_feature*, *memory*) computes the output features which are highly related with *question_feature.*
4. R(*output_feature*, *vocabulary*) produces the textual response according to the *output_feature.*

During training and testing, the flow above is followed. The only difference between training and testing is the parameter update. Parameters are updated while training, however; they are not updated during the test time. The detailed role of each component is as follows:

I component: This component uses Bag of Words approach to convert a sentence into an inner feature representation. Firstly, the dictionary created by assigning a number beginning from 1 to each unseen word in the text. This component uses this dictionary to find which vocabularies are present in a given sentence x. It returns a matrix of dimension $V \times 1$ where V is the length of the dictionary. The values in this matrix are all 0 except the indexes which correspond a word in the sentence x (these are 1).

G component: This component adds the feature representation of new sentence to the next available slot in the memory. In this case, the memory is reseted at the beginning of each story and in this way, the memory does not become huge and unrelated memory slots are removed. If a larger memory is necessary for the problem such as large-scale QA dataset case where there are 14M statements, a better approach would be fixing the size of the memory and the forgetting the least used memory slot when the memory is full.

O component: This component scores each memory slot according to a given question and finds the most relevant memory slots. The dataset consist of 20 different types of QA task and each requires different number of supporting facts to answer them. For instance, if the following story is told, it is enough to remember the third and six sentences in order to answer the question:

```
1 Mary moved to the bathroom.
2 Sandra journeyed to the bedroom.
3 Mary got the football there.
4 John went to the kitchen.
5 Mary went back to the kitchen.
6 Mary went back to the garden.
7 Where is the football? garden 3 6
```

For the above QA problem, each line is added to the memory and when the seventh line is the input, x = "Where is the football?", this O component finds the first relevant fact from the story using the following calculation:

$$o_1 = O(x, m) = \underset{i=1,\ldots,N}{\operatorname{argmax}} \; s_o(x,\, m_i)$$

where s_o is a function that scores the match between x and m_i. After the first supporting fact is found as o_1, the second supporting fact is found using the following calculation:

$$o_2 = O([x,\, m_{o1}], m) = \underset{i=1,\ldots,N}{\operatorname{argmax}} \; s_o([x,\, m_{o1}],\, m_i)$$

In this way, O component finds the both the first and the second supporting facts and the final output of this component is $[x,\, m_{o1},\, m_{o2}]$, and this the input for the R component.

R component: This component scores each vocabulary in the dictionary according to the input which contains the question feature representation and the supporting facts, and finds the most likely answer to the question. The following formula ranks the words:

$$r = \arg\max_w \; s_r([x, m_{o1}, m_{o2}], w)$$

In this way, R component returns the answer from the vocabulary by ranking them.

Both s_o and s_r functions use a fundamental scoring function called s which scores x according to y:

$$s(x, y) = \phi_x(\mathrm{x})^T \, U^T \, U \phi_y(\mathrm{y})$$

where U is a $n \times D$ matrix where D is the number of features and n is the embedding dimension. Both ϕ_x and ϕ_y are responsible for mapping the feature representation of the text into D-dimensional feature space where $D = (\textit{number of supporting facts + 1})|\mathrm{W}|$ where W is the length of dictionary. O and R components are uses different weight matrices and they are trained separately.

This above example explanation about the O and R components is for just 2 supporting facts case. If the necessary number of supporting facts changes then D changes. The implementation of memory networks for different number of supporting facts is publicly available at GitHub `https://github.com/aykutaaykut/Memory-Networks`

2.2 Training

Memory networks are trained using fully supervised settings. During the training O component finds the supporting facts; and its loss and accuracy is calculated; then the correct supporting facts with the question are given to R component as its parameters in order to calculate its loss and accuracy accurately. O component weight and R component weight are trained separately using softmax. In this way, they become expert on their tasks and the risk of misleading R component due to the incorrect supporting fact choice of O component is eliminated. However; during the test time, the correct supporting facts and response are used only for accuracy calculation. The output of O component is directly given to the R component. Knet.Adam is used in order to update the weights of O and R components.

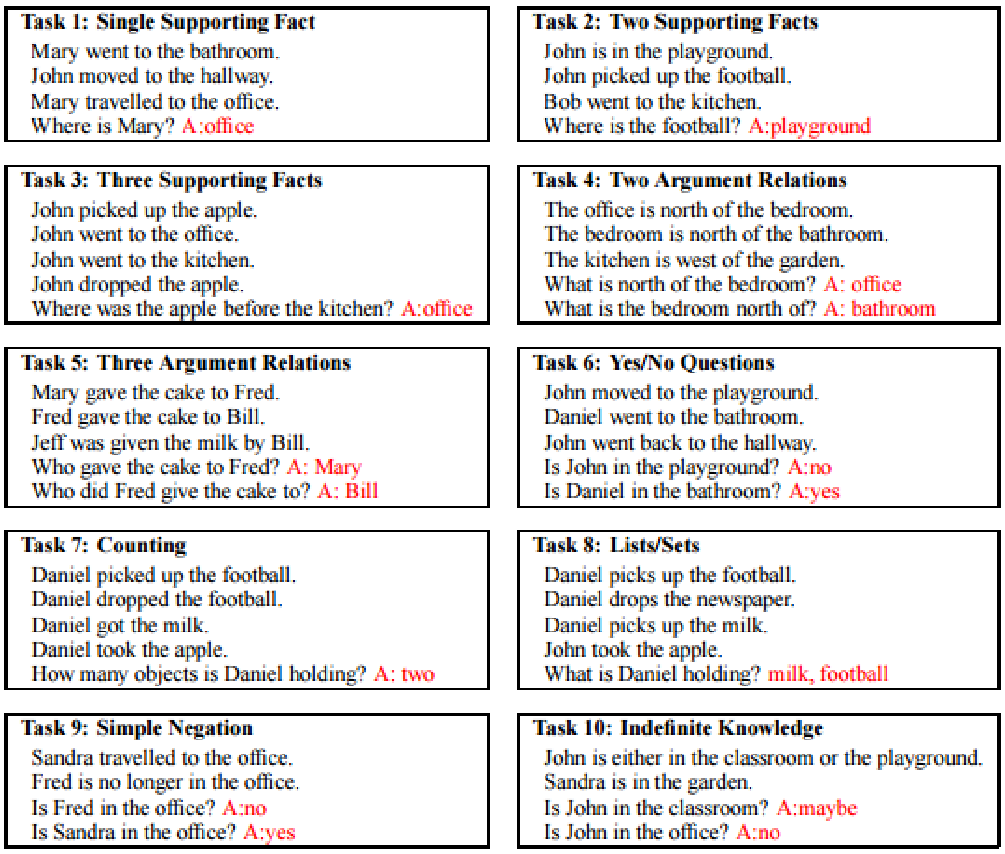
The actual paper uses margin rankig loss and Stochastic Gradient Descent while training, but when I implemented my model with them, the loss quantities fluctuates from one epoch to other epoch. Therefore, I tried to implement memory networks with softloss and achieved better experiment results in most of the experiments.

3 Experiments

The model is tested with 20 different tasks which each of them tests the different aspects of understanding of the story, question and the ability of relating the question to the sentences in the story. Although, the actual paper uses a different dataset, I preferred to do experiments with this dataset since it is publicly available at `https://research.fb.com/downloads/babi/` and categorizes QA tasks. Moreover, there is a repository in the GitHub `https://github.com/facebook/bAbI-tasks` contains the code for dataset generation. The 20 different toy tasks can be summarized as follows:

1. **Single Supporting Fact:** In order to answer a question, 1 supporting fact along with the question is enough. These questions generally ask the location of a person in given story.

2. **Two or There Supporting Facts:** These tasks are more difficult in comparison with the single supporting fact case. This time two or three supporting facts are necessary in order to answer the question. Two supporting fact questions are usually about the location of an object in the story and three supporting fact questions ask the location of an object before or after something happend such as "*Where was the apple before the kitchen?*"



Task 1: Single Supporting Fact
Mary went to the bathroom.
John moved to the hallway.
Mary travelled to the office.
Where is Mary? A:office

Task 2: Two Supporting Facts
John is in the playground.
John picked up the football.
Bob went to the kitchen.
Where is the football? A:playground

Task 3: Three Supporting Facts
John picked up the apple.
John went to the office.
John went to the kitchen.
John dropped the apple.
Where was the apple before the kitchen? A:office

Task 4: Two Argument Relations
The office is north of the bedroom.
The bedroom is north of the bathroom.
The kitchen is west of the garden.
What is north of the bedroom? A: office
What is the bedroom north of? A: bathroom

Task 5: Three Argument Relations
Mary gave the cake to Fred.
Fred gave the cake to Bill.
Jeff was given the milk by Bill.
Who gave the cake to Fred? A: Mary
Who did Fred give the cake to? A: Bill

Task 6: Yes/No Questions
John moved to the playground.
Daniel went to the bathroom.
John went back to the hallway.
Is John in the playground? A:no
Is Daniel in the bathroom? A:yes

Task 7: Counting
Daniel picked up the football.
Daniel dropped the football.
Daniel got the milk.
Daniel took the apple.
How many objects is Daniel holding? A: two

Task 8: Lists/Sets
Daniel picks up the football.
Daniel drops the newspaper.
Daniel picks up the milk.
John took the apple.
What is Daniel holding? milk, football

Task 9: Simple Negation
Sandra travelled to the office.
Fred is no longer in the office.
Is Fred in the office? A:no
Is Sandra in the office? A:yes

Task 10: Indefinite Knowledge
John is either in the classroom or the playground.
Sandra is in the garden.
Is John in the classroom? A:maybe
Is John in the office? A:no

Figure 1: Example stories for tasks 1 to 10.

3. **Two or Three Argument Relations:** In the two argument relations case, the order of the words in a sentence is important in order to answer the question, because two sentence which distinct meanings may have the same words. In the three argument relations, there are two people and one object and the goal is to understand which object is involved, who is the giver and the receiver.

4. **Yes/No Questions:** This is the simplest task among others. The questions are true/false questions and answers are either "yes" or "no".

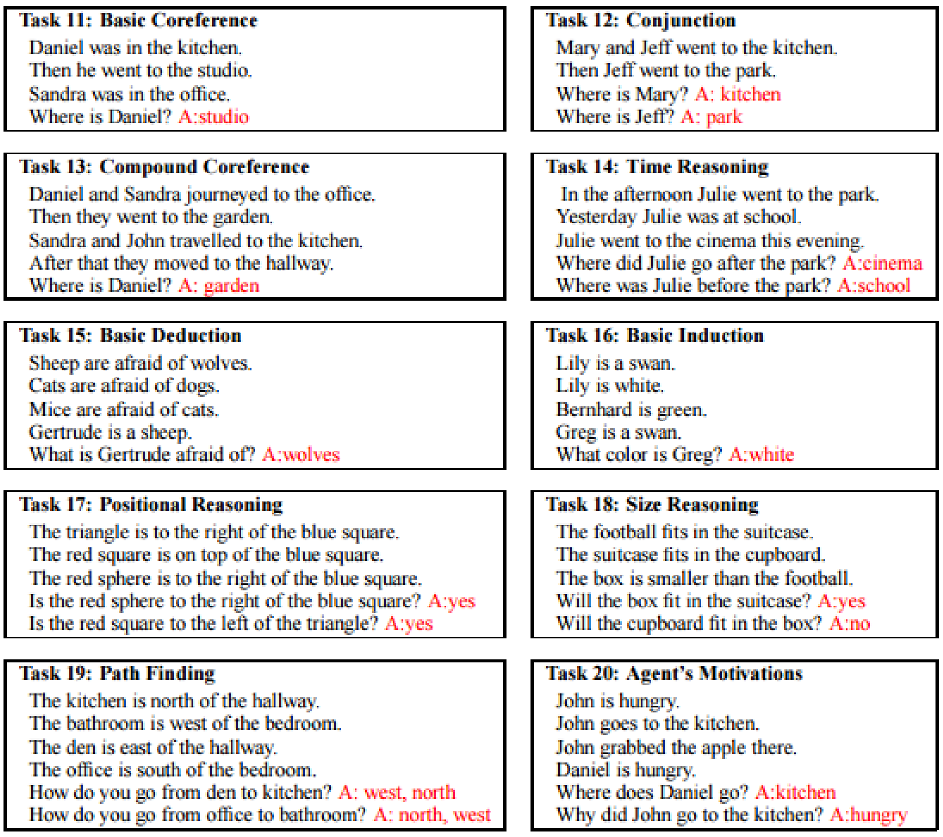
5. **Counting and List/Set:** Task 7 tests the model in terms of ability of counting objects. Questions about counting tasks are usually "*How many objects is Daniel holding?*". Task 8 is advanced version of the questions in Task 7, like "*What is Daniel holding?*".

6. **Simple Negation and Indefinite Knowledge:** Task 9 tests one of the simplest forms of negation, some supporting facts imply a statement is false. Task 10 asks questions which are about uncertainty such as given a fact that "*John is either in the classroom or the playground.*" and asked "*Is John in the classroom?*", then the correct answer is neither "yes" nor "no", the correct answer is "maybe".

7. **Basic Coreference, Conjunctions and Compound Coreference:** Task 11 tests the model whether it can find the nearest referent or not. Task 12 asks questions about conjunction of subjects in a sentence. Task 13 is a combination of Task 11 and task 12, this time one sentence has a referent and the referent has a reference to at least two subjects in another sentence.

8. **Time Reasoning:** Task 14 tests understanding the use of time expressions within the statements and answering questions according to their time order.

9. **Basic Deduction and Induction:** These tasks test the model's ability to understand basic inheritance of properties between sentences.



Task 11: Basic Coreference
Daniel was in the kitchen.
Then he went to the studio.
Sandra was in the office.
Where is Daniel? A:studio

Task 12: Conjunction
Mary and Jeff went to the kitchen.
Then Jeff went to the park.
Where is Mary? A: kitchen
Where is Jeff? A: park

Task 13: Compound Coreference
Daniel and Sandra journeyed to the office.
Then they went to the garden.
Sandra and John travelled to the kitchen.
After that they moved to the hallway.
Where is Daniel? A: garden

Task 14: Time Reasoning
In the afternoon Julie went to the park.
Yesterday Julie was at school.
Julie went to the cinema this evening.
Where did Julie go after the park? A:cinema
Where was Julie before the park? A:school

Task 15: Basic Deduction
Sheep are afraid of wolves.
Cats are afraid of dogs.
Mice are afraid of cats.
Gertrude is a sheep.
What is Gertrude afraid of? A:wolves

Task 16: Basic Induction
Lily is a swan.
Lily is white.
Bernhard is green.
Greg is a swan.
What color is Greg? A:white

Task 17: Positional Reasoning
The triangle is to the right of the blue square.
The red square is on top of the blue square.
The red sphere is to the right of the blue square.
Is the red sphere to the right of the blue square? A:yes
Is the red square to the left of the triangle? A:yes

Task 18: Size Reasoning
The football fits in the suitcase.
The suitcase fits in the cupboard.
The box is smaller than the football.
Will the box fit in the suitcase? A:yes
Will the cupboard fit in the box? A:no

Task 19: Path Finding
The kitchen is north of the hallway.
The bathroom is west of the bedroom.
The den is east of the hallway.
The office is south of the bedroom.
How do you go from den to kitchen? A: west, north
How do you go from office to bathroom? A: north, west

Task 20: Agent's Motivations
John is hungry.
John goes to the kitchen.
John grabbed the apple there.
Daniel is hungry.
Where does Daniel go? A:kitchen
Why did John go to the kitchen? A:hungry

Figure 2: Example stories for tasks 11 to 20.

10. **Positional and Size Reasoning:** Task 17 tests whether the model is able to find relative positions of colored blocks. Task 18 requires a general understanding of relative size of different objects.

11. **Path Finding:** The problem in Task 19 is about going somewhere to somewhere. It asks how do you get from one to other.

12. **Agent's Motivation:** Task 20 questions test the model understanding of the relation between certain actions and states. For example, the model learns that thirsty people are tend to go to the kitchen.

The dataset consists of both English and Hindi stories and reletad questions. There are 1000-question, 9000-question and shuffled versions in the dataset in both English and Hindi. In average each sentence has 6 words. Since the dataset is not unique and can be generated using the link `https://github.com/facebook/bAbI-tasks`, there is no absolute vocabulary size, however; Table 1 shows the vocabulary size of the dataset which is used in the experiments.

The actual paper suggests these parameters: 100 for the embedding dimension, 0.01 for learning rate and the model trained for 10 epochs. I used 100 for the embedding dimension as well, but I set the initial learning rate of O component and R component to 0.001, used normal distribution to initialize the weights and trained my model for 100 epochs.

4 Analysis and Contributions

In order to analyze the results and compare them with the results in the paper, I created a table for each task and plotted graphs using the table, it is publicly available at `https://drive.google.com/file/d/0B4pJdDuWybS9RnVoQUUxUGV3YlU/view` and Table 2 shows the maximum test accuracy achieved for each task:

Task	Vocabulary Size	Task	Vocabulary Size
1	19	11	26
2	33	12	20
3	34	13	26
4	15	14	27
5	39	15	25
6	35	16	17
7	43	17	20
8	45	18	20
9	24	19	32
10	25	20	39

Table 1: Vocabulary size in each task

Task	Results in the paper	Results of my implementation
1	100	62.5
2	100	52.8
3	20	35.5
4	71	69.2
5	83	80.4
6	47	53
7	68	62.2
8	77	39.9
9	65	68.7
10	59	55.3
11	100	73.8
12	100	48
13	100	57
14	99	100
15	74	100
16	27	97.4
17	54	54.2
18	57	53.1
19	0	12.9
20	100	100
Mean	**70.05**	**63.8**

Table 2: Test accuracy (%) on 20 Tasks (1000 training examples each)

The paper states the mean of the performances on 20 tasks is 75 %, however; when I do the calculation, I found 70.05 %. Here is the mean performance calculation of the paper results:

$$\frac{\sum_{t=1}^{20} accuracy\ on\ task\ t}{20} = \frac{1401}{20} = 70.05$$

I achieved better accuracy on 7 tasks and similar accuracy on 7 tasks, however; I got poor accuracy on 6 tasks. The followings can cause getting poor accuracy in these tasks:

1. Normally each tasks other than task 7 and task 8 require constant number of supporting facts to answer questions correctly.In task 7 and 8, the number of supporting facts needed changes for different questions, but my model assumes that each question requires the same number of supporting facts at the beginning of training, so it uses D = maximum number of supporting facts needed and updates it during the training. This also causes overfitting after a few epochs.

2. The inner feature representation of sentences could be another reason. I could not find the code of authors' implementation on the Internet, thus I implemented my memory network by understanding the paper and watching video lectures online about memory networks.

3. The paper suggests an improvement for memory networks which is called time feature. A memory network with time feature knows which sentence comes first and scores memories according to this fact while finding supporting facts, therefore relative time is used. Adding time feature requires using hinge loss as the loss function, however; hinge loss is not suitable in my implementation since it is not differentiable. Thus, I used absolute time with softloss instead of relative time with hinge loss.

4. In the original paper, margin ranking loss and SGD is used for training, but I used softloss and Knet.Adam for training.

My iplementation have achieved higher accuracy on the following 7 tasks:

- Task 3: Three Supporting Facts **35.5 %**
- Task 6: Yes/No Questions **53 %**
- Task 9: Simple Negation **68 %**
- Task 14: Time Reasoning **100 %**
- Task 15: Basic Deduction **100 %**
- Task 16: Basic Induction **97.4 %**
- Task 19: Path Finding **12.9 %**

Higher accuracies are obtained on especially Task 3, Task 15, Task 16 and Task 19 in my implementation. My implementation is not good enough to answer all questions on Task 3 as well, but it has achieved higher accuracy. The paper's implementation is not good enough to answer all of the questions on Task 15 and Task 16, and obtains poor accuracy on these tasks; on the other hand, my implementation of memory networks is better on these tasks and answers approximately all of the questions correctly. If the Task 19 is considered, the paper's implementation obtained 0 % accuracy on this task and agent never finds the correct path. My implementation also does not always give the correct path to the agent, but at least 12.9 % of the time the agent reaches wherever it wants. Possible reasons why my implementation has achieved higher accuracies on these tasks:

1. I used Knet.Adam as the parameter optimizer.
2. I used softloss instead of margin ranking loss.
3. My internal representation might be more appropriate for these tasks.

Please see the Figure 3 and Figure 4 for the graphs of softloss at each task on the next page.

5 Related Work

Question Answering is one of the fundamental problems in machine learning. There are various approaches in the past which include using of RNN, LSTM, SVM, etc. Memory Networks work of Weston et al. introduced a new approach called MemNN which is the abbreviation of Memory Neural Networks. They tested their approach with large-scale and toy task QA tasks, and proved MemNN are able to solve these problems. I used their approach to solve their QA problems with absolute time and tested my model with the dataset which is used for Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks. The dataset is more generalizable, comprehensive and splits the tasks into categories.

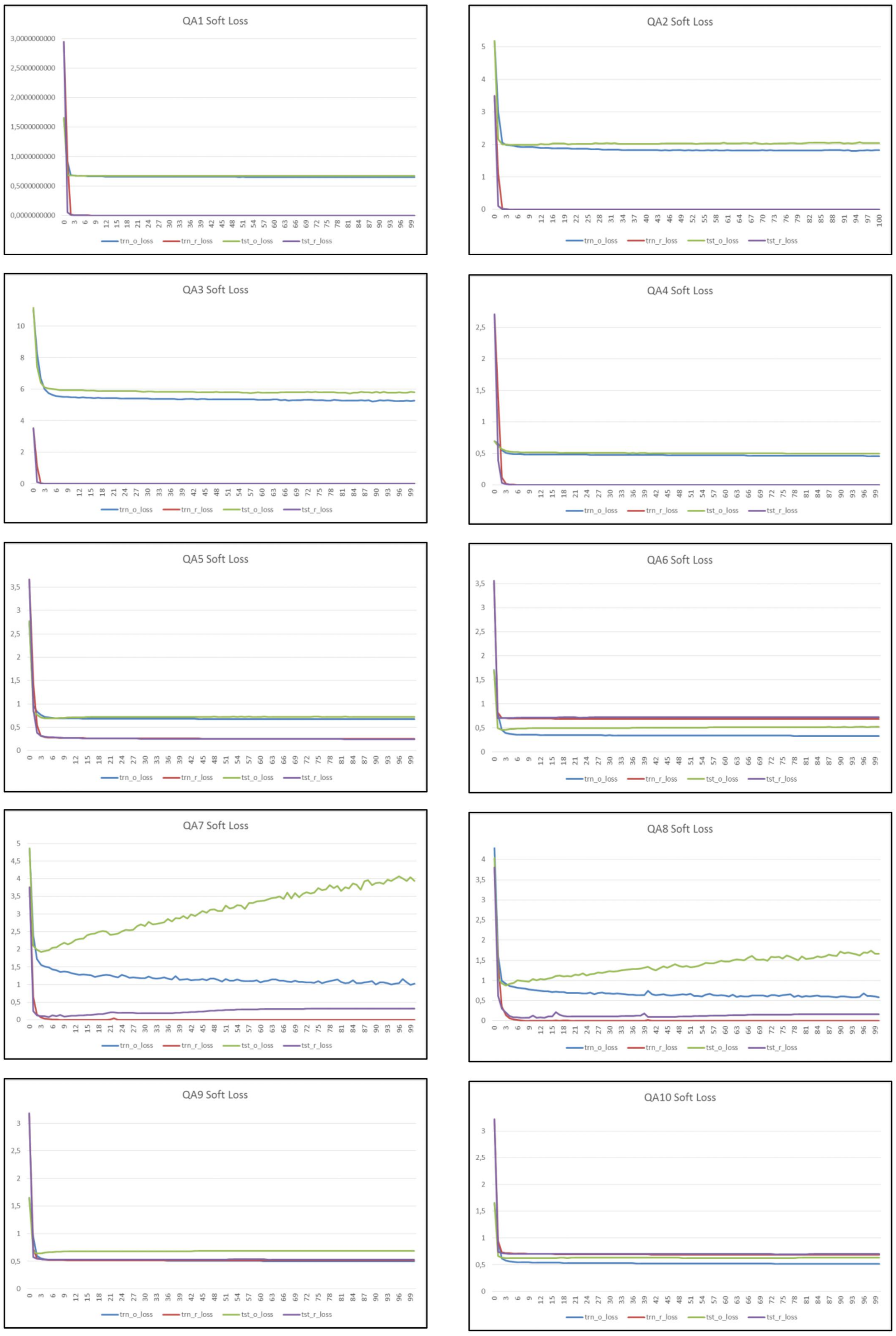


Figure 3: Loss graphs of tasks 1 to 10

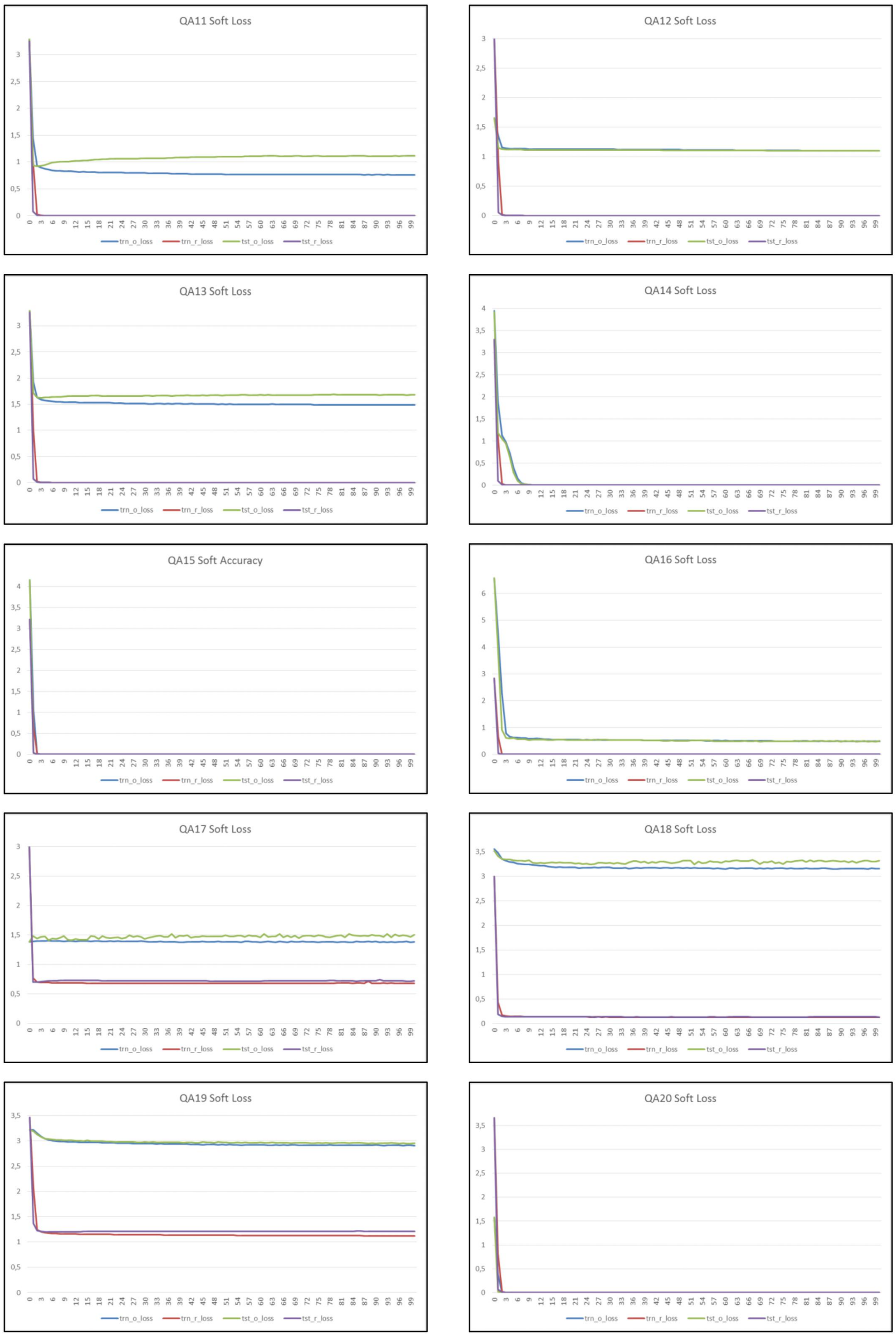


Figure 4: Loss graphs of tasks 11 to 20

6 Conclusion

In conclusion, recently introduced new class of learning models memory networks are re-implemented and tested. As stated in the Analysis and Contributions section, the model has achieved similar and better accuracies on most of the tasks (14 out of 20 tasks) In the other tasks, I obtained poor accuracies according to the stated accuracies in Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks. The possible reasons for this results are explained in the Analysis and Contributions section. In summary, the internal representation, absolute time vs. relative time and different number of supporting facts cause the accuracy difference. The results in the Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks is for another implementation of memory networks which uses relative time. Overall the model achieved nearly same accuracy in terms of mean performance and fulfilled the task of reimplementation of MemNN.

References

[1] Weston, Jason, Chopra, Sumit, and Bordes, Antoine. 2015. Memory Networks. In *ICLR 2015*. `https://arxiv.org/pdf/1410.3916.pdf`

[2] Weston, Jason, Bordes, Antoine, Chopra, Sumit, Rush, Alexander M., Merrienboer, Bart van, Joulin Armand, and Mikolov Tomas. 2015. Towards AI-Complete Question Answering: A Set of Preprequisite Toy Tasks. In *ICLR 2016*. `https://arxiv.org/pdf/1502.05698.pdf`

[3] Video Lecture with Sumit Chopra, `http://videolectures.net/deeplearning2016_chopra_attention_memory/`

[4] Memory Networks from Jason Weston, `https://www.youtube.com/watch?v=Xumy3Yjq4zk`