Reasoning, Attention and Memory

Sumit Chopra
Facebook AI Research
Deep Learning for Vision

What if we treat an existing deep model as a black box in pedestrian detection?


Figure Credit: Xiaogang Wang
Deep Learning for Speech

Application: Speech

“...can for example present significant universitywide issues to the senate.”

Spectrogram: window in time → vector of frequencies; slide; repeat

Figure Credit: Nvidia
“The movie was not bad at all. I had fun.”
Deep Models

"The movie was not bad at all. I had fun."

- Input Representation
- $G_{W_2}$ Classifier/Regressor (decoder)
- $F_{W_1}$ Feature Extractor (encoder)
- Loss Function

Typically a Linear Projection with some non-linearity (log-soft-max)

Fully Connected Network
Convolution Network
Recurrent Network

can be seen as a prior on the type of transformation you want
Deep Models

Loss Function

$G_{W_2}$

Classifier/Regressor (decoder)

Typically a Linear Projection with some non-linearity (log-soft-max)

$W_1$

Feature Extractor (encoder)

Inputs: generally considered I.I.D.

Outputs: classification or regression

“The movie was not bad at all. I had fun.”

Learnable parametric function

Inputs: generally considered I.I.D.

Outputs: classification or regression

“The movie was not bad at all. I had fun.”

Embedding Matrix
Scenario 1

Joe went to the kitchen. Fred went to the kitchen. Joe picked up the milk. Joe travelled to the office. Joe left the milk. Joe went to the bathroom.
Joe went to the kitchen. Fred went to the kitchen. Joe picked up the milk. Joe travelled to the office. Joe left the milk. Joe went to the bathroom.

Where is the milk now?
Where is Joe?
Where was Joe before the office?
Scenario 1

Joe went to the kitchen. Fred went to the kitchen. Joe picked up the milk. Joe travelled to the office. Joe left the milk. Joe went to the bathroom.
Where is the milk now? A: office
Where is Joe?
Where was Joe before the office?
Scenario 1

Joe went to the kitchen. Fred went to the kitchen. Joe picked up the milk. Joe travelled to the office. Joe left the milk. Joe went to the bathroom.

Where is the milk now? A: office
Where is Joe? A: bathroom
Where was Joe before the office?
Scenario 1

Joe went to the kitchen. Fred went to the kitchen. Joe picked up the milk. Joe travelled to the office. Joe left the milk. Joe went to the bathroom.
Where is the milk now?  A: office
Where is Joe?  A: bathroom
Where was Joe before the office?  A: kitchen
Scenario 2

S: 1 Mr. Cropper was opposed to our hiring you.
   2 Not, of course, that he had any personal objection to you, but he is set
      against female teachers, and when a Cropper is set there is nothing on earth can
      change him.
   3 He says female teachers can't keep order.
   4 He 's started in with a spite at you on general principles, and the boys know
      it.
   5 They know he 'll back them up in secret, no matter what they do, just to prove
      his opinions.
   6 Cropper is sly and slippery, and it is hard to corner him. '
   7 "" Are the boys big? ""
   8 queried Esther anxiously.
   9 "" Yes.
   10 Thirteen and fourteen and big for their age.
   11 You can't whip 'em -- that is the trouble.
   12 A man might, but they 'd twist you around their fingers.
   13 You 'll have your hands full, I 'm afraid.
   14 But maybe they 'll behave all right after all. '
   15 Mr. Baxter privately had no hope that they would, but Esther hoped for the
      best.
   16 She could not believe that Mr. Cropper would carry his prejudices into a
      personal application.
   17 This conviction was strengthened when he overtook her walking from school the
      next day and drove her home.
   18 He was a big, handsome man with a very suave, polite manner.
   19 He asked interestedly about her school and her work, hoped she was getting on
      well, and said he had two young rascals of his own to send soon.
   20 Esther felt relieved.
Scenario 2

S: 1 Mr. Cropper was opposed to our hiring you.
   2 Not, of course, that he had any personal objection to you, but he is set
      against female teachers, and when a Cropper is set there is nothing on earth can
      change him.
   3 He says female teachers can’t keep order.
   4 He’s started in with a spite at you on general principles, and the boys know
      it.
   5 They know he’ll back them up in secret, no matter what they do, just to prove
      his opinions.
   6 Cropper is sly and slippery, and it is hard to corner him."
   7 "" Are the boys big?"
   8 queried Esther anxiously.
   9 "" Yes.
   10 Thirteen and fourteen and big for their age.
   11 You can’t whip ‘em -- that is the trouble.
   12 A man might, but they’d twist you around their fingers.
   13 You’ll have your hands full, I’m afraid.
   14 But maybe they’ll behave all right after all."
   15 Mr. Baxter privately had no hope that they would, but Esther hoped for the
      best.
   16 She could not believe that Mr. Cropper would carry his prejudices into a
      personal application.
   17 This conviction was strengthened when he overtook her walking from school the
      next day and drove her home.
   18 He was a big, handsome man with a very suave, polite manner.
   19 He asked interestingly about her school and her work, hoped she was getting on
      well, and said he had two young rascals of his own to send soon.
   20 Esther felt relieved.

q: She thought that Mr. ______ had exaggerated matters a little.
Scenario 2

S: 1 Mr. Cropper was opposed to our hiring you.
   2 Not, of course, that he had any personal objection to you, but he is set
   against female teachers, and when a Cropper is set there is nothing on earth can
   change him.
   3 He says female teachers can't keep order.
   4 He 's started in with a spite at you on general principles, and the boys know
   it.
   5 They know he 'll back them up in secret, no matter what they do, just to prove
   his opinions.
   6 Cropper is sly and slippery, and it is hard to corner him. '
   7 `` Are the boys big? '''
   8 queried Esther anxiously.
   9 `` Yes.
   10 Thirteen and fourteen and big for their age.
   11 You can't whip 'em -- that is the trouble.
   12 A man might, but they 'd twist you around their fingers.
   13 You 'll have your hands full, I 'm afraid.
   14 But maybe they 'll behave all right after all. '''
   15 Mr. Baxter privately had no hope that they would, but Esther hoped for the
   best.
   16 She could not believe that Mr. Cropper would carry his prejudices into a
   personal application.
   17 This conviction was strengthened when he overtook her walking from school the
   next day and drove her home.
   18 He was a big, handsome man with a very suave, polite manner.
   19 He asked interestingly about her school and her work, hoped she was getting on
   well, and said he had two young rascals of his own to send soon.
   20 Esther felt relieved.

q: She thought that Mr. Baxter had exaggerated matters a little.
Scenario 3

**Shaolin Soccer** directed by **Stephen Chow**
**Shaolin Soccer** written by **Stephen Chow**
**Shaolin Soccer** starred actors **Stephen Chow**
**Shaolin Soccer** release year 2001
**Shaolin Soccer** has genre comedy
**Shaolin Soccer** has tags martial arts, kung fu soccer, **stephen chow**

**Kung Fu Hustle** directed by **Stephen Chow**
**Kung Fu Hustle** written by **Stephen Chow**
**Kung Fu Hustle** starred actors **Stephen Chow**
**Kung Fu Hustle** has genre comedy action
**Kung Fu Hustle** has imdb votes famous

**Kung Fu Hustle** has tags comedy, action, martial arts, kung fu, china, soccer, hong kong, **stephen chow**
The God of Cookery directed by **Stephen Chow**
The God of Cookery written by **Stephen Chow**
The God of Cookery starred actors **Stephen Chow**
The God of Cookery has tags hong kong **Stephen Chow**
From Beijing with Love directed by **Stephen Chow**
From Beijing with Love written by **Stephen Chow**
From Beijing with Love starred actors **Stephen Chow**, Anita Yuen

...<and more> ...
Scenario 3

Who wrote Kung Fu Hustle?
Scenario 3

| Shaolin Soccer directed_by Stephen Chow |
| Shaolin Soccer written_by Stephen Chow |
| Shaolin Soccer starred_actors Stephen Chow |
| Shaolin Soccer release_year 2001 |
| Shaolin Soccer has_genre comedy |
| Shaolin Soccer has_tags martial arts, kung fu soccer, stephen chow |
| Kung Fu Hustle directed_by Stephen Chow |
| Kung Fu Hustle written_by Stephen Chow |
| Kung Fu Hustle starred_actors Stephen Chow |
| Kung Fu Hustle has_genre comedy action |
| Kung Fu Hustle has_imdb_votes famous |
| Kung Fu Hustle has_tags comedy, action, martial arts, kung fu, china, soccer, hong kong, stephen chow |
| The God of Cookery directed_by Stephen Chow |
| The God of Cookery written_by Stephen Chow |
| The God of Cookery starred_actors Stephen Chow |
| The God of Cookery has_tags hong kong Stephen Chow |
| From Beijing with Love directed_by Stephen Chow |
| From Beijing with Love written_by Stephen Chow |
| From Beijing with Love starred_actors Stephen Chow, Anita Yuen |

...<and more> ...

I’m interested in watching a Stephen Chow movie other than Kung Fu Hustle. Can you suggest something?
<table>
<thead>
<tr>
<th>User</th>
<th>Bot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hi!</td>
<td>Hello. What can I help you with today?</td>
</tr>
<tr>
<td>I'd like to book a table for 6 in an expensive restaurant which serves French food.</td>
<td>I'm on it. Where should it be?</td>
</tr>
<tr>
<td>In London</td>
<td>Ok. Let me look into some options for you.</td>
</tr>
<tr>
<td>Actually I would prefer for four.</td>
<td>Sure. Is there anything else to update?</td>
</tr>
<tr>
<td>No.</td>
<td>Ok. Let me look into some options for you.</td>
</tr>
<tr>
<td>What do you think of The_Fancy_Pub</td>
<td>No. I don't like that.</td>
</tr>
<tr>
<td>Lets do it</td>
<td>Sure. Let me find another option.</td>
</tr>
<tr>
<td>Can you provide me with the address?</td>
<td>How about The_palace</td>
</tr>
<tr>
<td>You rock!</td>
<td>Great. Let me make the reservation.</td>
</tr>
<tr>
<td>Here it is: The_Palace_Address You roci</td>
<td></td>
</tr>
<tr>
<td>You rock!</td>
<td>Is there anything else i can help you with</td>
</tr>
<tr>
<td>No. Thanks.</td>
<td>You are welcome.</td>
</tr>
</tbody>
</table>
What is Required?

Not all problems can be mapped to $y = f(x)$

The model needs to **remember external context**

Given an input, the model needs to know **where to look for in the context**

It needs to know **what to look for in the context**

It needs to know **how to reason** using this external context

It needs to handle the **potentially changing external context**
Not all problems can be mapped to $y = f(x)$

The model needs to remember external context

Given an input, the model needs to know where to look for in the context

It needs to know what to look for in the context

It needs to know how to reason using this external context

It needs to handle the potentially changing external context

Needs to have a notion of Memory
Possible Solution

Hidden states of RNNs have memory

Run an RNN on the context/story/KB and get its representation

Use the representation to map question to answers/response
Possible Solution

Hidden states of RNNs have memory

Run an RNN on the context/story/KB and get its representation

Use the representation to map question to answers/response

We know this will not scale!
Outline

Memory Networks
Fully Supervised MemNNs
End2End MemNNs
Key-Value MemNNs
Architecture - How to reason - Advantages/Disadvantages

Neural Turing Machines
Architecture - How to reason - Advantages/Disadvantages

Stack/List/Queue Augmented RNNs
If time permits - otherwise you’ll hear about this in lot more detail tomorrow
Controller takes external inputs and controls the heads

Heads read from and write to the memory

Controller combines memory reads with external input to produce an external output

What goes inside each of these components defines the model
Memory Networks

Class of models which combine large memory with learning component which can read and write to it.

Incorporates reasoning via attention over memory.

The model framework is flexible enough to store rich representations of input in memory.

Models are scalable - can store and read large amount of data in memory - entire KB.

Memory specification is flexible - can have both long-term memory and short-term memory - consider dialog modeling.
Memory Networks

Step 1: controller converts incoming data to internal feature representation (I)

Step 2: write head updates the memories and writes the data into memory (G)

Step 3: given the external input, the read head reads the memory and fetches relevant data (O)

Step 4: controller combines the external data with memory contents returned by read head to generate output (O, R)
Memory Networks (Fully Supervised)

John was in the bathroom.
   Bob was in the office.
   John went to kitchen.
Bob travelled back home.
John was in the bathroom.
Bob was in the office.
John went to kitchen.
Bob travelled back home.
Where is John? A: kitchen
John was in the bathroom.
   Bob was in the office.
   John went to kitchen.
Bob travelled back home.
Where is John? A: kitchen

Context
Supporting Fact
Question, Answer Pair
John was in the bathroom.
Bob was in the office.
John went to kitchen.
Bob travelled back home.
Where is John? A: kitchen

Memories

\[ m_i = f(\text{John was in the bathroom.}) \]
\[ m_{i+1} = f(\text{Bob was in the office.}) \]
\[ m_{i+2} = f(\text{John went to the kitchen.}) \]
\[ m_{i+3} = f(\text{Bob travelled back home.}) \]

Step 1
Store the representations of facts in the memory
Free to choose what representations you store
Individual words - window of words - full sentences
Bag-of-words - CNN - RNN - LSTM
John was in the bathroom.
Bob was in the office.
John went to kitchen.
Bob travelled back home.
Where is John? A: kitchen

Memories

\[ m_i = f(\text{John was in the bathroom.}) \]
\[ m_{i+1} = f(\text{Bob was in the office.}) \]
\[ m_{i+2} = f(\text{John went to the kitchen.}) \]
\[ m_{i+3} = f(\text{Bob travelled back home.}) \]

\[ x = f(\text{Where is John?}) \]

Step 2
Represent the question using similar function.
Memory Networks (Fully Supervised)

John was in the bathroom.
Bob was in the office.
John went to kitchen.
Bob travelled back home.
Where is John? A: kitchen

Memories

\[
\begin{align*}
  m_i &= f(\text{John was in the bathroom.}) \\
  m_{i+1} &= f(\text{Bob was in the office.}) \\
  m_{i+2} &= f(\text{John went to the kitchen.}) \\
  m_{i+3} &= f(\text{Bob travelled back home.}) \\
  x &= f(\text{Where is John?})
\end{align*}
\]

Step 3
Define a scoring function $S$ and score the memories with the question
Scoring function should be such that it gives a high score to the relevant memories:

$S(\text{Where is John?, John went to the kitchen.}) > S(\text{Where is John?, Bob travelled back home.})$
Memory Networks (Fully Supervised)

John was in the bathroom.
Bob was in the office.
John went to kitchen.
Bob travelled back home.
Where is John?

Example Choices

\[
q^t U d
\]

\[
G_w(q, d)
\]

Scoring function should be such that it gives a high score to the relevant memories:

\[
S(\text{Where is John?}, \text{John went to the kitchen.}) > S(\text{Where is John?}, \text{Bob travelled back home.})
\]
Memory Networks (Fully Supervised)

John was in the bathroom. Bob was in the office. John went to kitchen. Bob travelled back home. Where is John? A: kitchen

Memories

\[ m_i = f(John \text{ was in the bathroom.}) \]
\[ m_{i+1} = f(Bob \text{ was in the office.}) \]
\[ m_{i+2} = f(John \text{ went to the kitchen.}) \]
\[ m_{i+3} = f(Bob \text{ travelled back home.}) \]

\[ x = f(Where \text{ is John?}) \]

Step 4

Define another parametric function which maps the current question and relevant memories to the final response.

In the first experiments, this was another scoring function which scored all possible responses against the given input and memories.
John was in the bathroom.
Bob was in the office.
John went to kitchen.
Bob travelled back home.
Where is John? A: kitchen

Memories

\[ m_i = f(\text{John was in the bathroom.}) \]
\[ m_{i+1} = f(\text{Bob was in the office.}) \]
\[ m_{i+2} = f(\text{John went to the kitchen.}) \]
\[ m_{i+3} = f(\text{Bob travelled back home.}) \]

\[ x = f(\text{Where is John?}) \]

Inference

Given the question, pick the memory which scores the highest
Use the selected memory and the question to generate the answer
Memory Networks (Fully Supervised)

Training

It involves training the memory representations and the scoring functions to generate answer.

We do so by minimizing the following loss:

\[
L = \sum_{f \neq m_{o1}} \max(0, \gamma - S_o(x, m_{o1}) + S_o(x, f)) + \\
\sum_{\bar{r} \neq r} \max(0, \gamma - S_r([x, m_{o1}], r) + S_r([x, m_{o1}], \bar{r}))
\]

Memories

\[
m_i = f(John was in the bathroom.)
\]

\[
m_{i+1} = f(Bob was in the office.)
\]

\[
m_{i+2} = f(John went to the kitchen.)
\]

\[
m_{i+3} = f(Bob travelled back home.)
\]

\[
x = f(Where is John?)
\]
Memory Networks (Fully Supervised)

Training

It involves training the memory representations and the scoring functions to generate answer.

We do so by minimizing the following loss:

$$L = \sum_{\stackrel{f \neq m_{o1}}{f \neq \bar{r}}} \max(0, \gamma - S_o(x, m_{o1}) + S_o(x, \bar{f})) +$$

$$\sum_{\bar{r} \neq r} \max(0, \gamma - S_r([x, m_{o1}], r) + S_r([x, m_{o1}], \bar{r}))$$

We had access to true supporting fact during training, that’s what we mean by “Fully Supervised”.

Memories

- $m_i = f(John \ was \ in \ the \ bathroom.)$
- $m_{i+1} = f(Bob \ was \ in \ the \ office.)$
- $m_{i+2} = f(John \ went \ to \ the \ kitchen.)$
- $m_{i+3} = f(Bob \ travelled \ back \ home.)$

$x = f(Where \ is \ John?)$

$S_o$: scoring function for memories
$S_r$: scoring function for responses
Memory Networks (Fully Supervised)

**Training**

It involves training the memory representations and the scoring functions to generate answer. We do so by minimizing the following loss:

\[
L = \sum_{f \neq m_{o1}} \max(0, \gamma - S_o(x, m_{o1}) + S_o(x, \bar{f})) + \\
\sum_{\bar{r} \neq r} \max(0, \gamma - S_r([x, m_{o1}], r) + S_r([x, m_{o1}], \bar{r}))
\]

We had access to true supporting fact during training, that's what we mean by “Fully Supervised”.

**Memories**

\[
m_i = f(John was in the bathroom.)
\]

\[
m_{i+1} = f(Bob was in the office.)
\]

\[
m_{i+2} = f(John went to the kitchen.)
\]

\[
m_{i+3} = f(Bob travelled back home.)
\]

\[
x = f(Where is John?)
\]

\[
S_o: \text{scoring function for memories}
\]

\[
S_r: \text{scoring function for responses}
\]

This was the case when we have a single supporting fact!
Memory Networks (Fully Supervised)

John is in the playground.  
Bob is in the office.  
John picked up the football.  
Bob went to the kitchen.  
Where is the football? A: playground.
Memory Networks (Fully Supervised)

John is in the playground.  
Bob is in the office.  
John picked up the football.  
Bob went to the kitchen.  
Where is the football? A: playground.

The current loss function will not work

\[
L = \sum_{\bar{f} \neq m_{o1}} \max(0, \gamma - S_o(x, m_{o1}) + S_o(x, \bar{f})) + \\
\sum_{\bar{r} \neq r} \max(0, \gamma - S_r([x, m_{o1}], r) + S_r([x, m_{o1}], \bar{r}))
\]
Memory Networks (Fully Supervised)

John is in the playground.  
Bob is in the office.  
John picked up the football.  
Bob went to the kitchen.  
Where is the football? A: playground.

The current loss function will not work

\[
L = \sum_{\bar{f} \neq m_{o1}} \max(0, \gamma - S_o(x, m_{o1}) + S_o(x, \bar{f})) + \sum_{\bar{r} \neq r} \max(0, \gamma - S_r([x, m_{o1}], r) + S_r([x, m_{o1}], \bar{r}))
\]

But the cool thing is that we can iterate!
Memory Networks (Fully Supervised)

John is in the playground.  
   Bob is in the office.  
John picked up the football.  
   Bob went to the kitchen.  
Where is the football? A: playground.

$$Loss = \sum_{\bar{f} \neq m_0} max(0, \gamma - S_o(x, m_0) + S_o(x, \bar{f}))$$
$$+ \sum_{\bar{f}' \neq m_2} max(0, \gamma - S_o([x, m_0], m_2) + S_o([x, m_0], \bar{f}'))$$
$$+ \sum_{\bar{r} \neq r} max(0, \gamma - S_r([x, m_0, m_2], r) + S_r([x, m_0, m_2], \bar{r}))$$
Memory Networks (Fully Supervised)

\[
Loss = \sum_{\bar{f} \neq m_0} \max(0, \gamma - S_o(x, m_0) + S_o(x, \bar{f})) \\
+ \sum_{\bar{f}' \neq m_2} \max(0, \gamma - S_o([x, m_0], m_2) + S_o([x, m_0], \bar{f}')) \\
+ \sum_{\bar{r} \neq r} \max(0, \gamma - S_r([x, m_0, m_2], r) + S_r([x, m_0, m_2], \bar{r}))
\]

John is in the playground.
Bob is in the office.
John picked up the football.
Bob went to the kitchen.
Where is the football? \textbf{A: playground.}
Memory Networks (Fully Supervised)

John is in the playground.  
Bob is in the office.  
John picked up the football.  
Bob went to the kitchen.  
Where is the football? A: playground.

Loss = \sum_{\bar{f} \neq m_{o1}} max(0, \gamma - S_o(x, m_{o1}) + S_o(x, \bar{f})) 
+ \sum_{\bar{f}' \neq m_{o2}} max(0, \gamma - S_o([x, m_{o1}], m_{o2}) + S_o([x, m_{o1}], \bar{f}')) 
+ \sum_{\bar{r} \neq r} max(0, \gamma - S_r([x, m_{o1}, m_{o2}], r) + S_r([x, m_{o1}, m_{o2}], \bar{r}))
Memory Networks (Fully Supervised)

John is in the playground.
Bob is in the office.
John picked up the football.
Bob went to the kitchen.

Where is the football?
A: playground.

Supporting Fact 1
Loss = \bar{f}_6 = m_1 \max(0, S_o(x, m_1), \bar{f}) + S_o([x, m_1], \bar{f})

Supporting Fact 2
+ \sum_{\bar{f} \neq m_2} \max(0, \gamma - S_o([x, m_1], m_2) + S_o([x, m_1], \bar{f}))
+ \sum_{\bar{r} \neq r} \max(0, \gamma - S_r([x, m_1, m_2], r) + S_r([x, m_1, m_2], \bar{r}))

We call these “Hops”
And they are not limited to two
bAbI Dataset: Slight Digression

While working on MemNNs we also defined 20 simulated tasks to test models which have long-term memory — can do complex reasoning using those memories.

The objective was to generate a set of tasks which can act “unit tests” in software engineering.

Each task would test a single (or may be a couple of) “skills” which we think are natural to humans w.r.t. text understanding and reasoning.

Language skills - conjunction, coreference, negation etc

Reasoning skills - counting, path finding etc
bAbI Dataset: Simulator

- go <place>
- get <object>
- get <object1> from <object2>
- put <object1> in/on <object2>
- give <object> to <person>
- drop <object>
- look
- inventory
- examine <object>

+ 2 commands for "gods" (superusers):
  - create <object>
  - set <obj1> <relation> <obj2>
bAbI Dataset: Simulator

Example

Command format

jason go kitchen
jason get milk
jason go office
jason drop milk
jason go bathroom
where is milk? A: office
where is jason? A: bathroom

Story

Jason went to the kitchen.
Jason picked up the milk.
Jason travelled to the office.
Jason left the milk there.
Jason went to the bathroom.
Where is the milk now? A: office
Where is Jason? A: bathroom
bAbI Dataset

Factoid QA with Single Supporting Fact

Questions where a single supporting fact is used and it is given in the context

We test this by asking for location of a person

John is in the playground.
Bob is in the office.
Where is John? A: playground

We could use supporting facts for supervision at training time, but are not known at test time (we call this “strong supervision”). However weak supervision is much better!
Factoid QA with Two Supporting Facts

Questions where two supporting facts have to be chained together in order to find the answer

John is in the playground.
Bob is in the office.
John picked up the football.
Bob went to the kitchen.
Where is the football? A: playground

Factoid QA with Three Supporting Facts

Questions where Three supporting facts have to be chained together in order to find the answer

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
Two Argument Relations: Subject vs. Object

Questions where the model learns the ability to differentiate and recognize subjects and objects

We make the problem harder by having sentences which have re-ordered words

For example the two questions below have same words but different meaning

The office is north of the bedroom.
The bedroom is north of the bathroom.
What is north of the bedroom? A: office
What is the bedroom north of? A: bathroom
bAbI Dataset

Three Argument Relations

Questions where the model learns the ability to differentiate and recognize two subjects and an object

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
bAbI Dataset

Yes/No Questions

Questions where the model learns answer true/false type questions

Start with the simple case of a single supporting fact

| John is in the playground.  
| Daniel picks up the milk.  
| Is John in the classroom? A:no  
| Does Daniel have the milk? A:yes |
bAbI Dataset

Counting
Questions where the model learns to count

- 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

Lists/Sets
Questions where the model learns to generate a set or list of answers

- Daniel picks up the football.
- Daniel drops the newspaper.
- Daniel picks up the milk.

What is Daniel holding? A: milk, football
bAbI Dataset

Indefinite Knowledge

Questions where the model learns to answer under uncertainty

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
bAbI Dataset

Basic Coreference
Questions where the model learns to recognize coreferences of a single entity

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

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

Compound Coreference
Questions where the model learns to recognize coreferences of multiple entities
bAbI Dataset

Time Manipulation
While we have an implicit notion of time already in our tasks, this particular one tests understanding the use of explicit time expressions.

Basic Deduction
Questions where the model learns basic deduction via inheritance of properties.

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

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

Deduction for MemNNs should be hard because it effectively involves search.
bAbI Dataset

Positional Reasoning
Questions where the model learns to do spatial reasoning

Reasoning About Size
Questions where the model learns to reason about relative sizes of objects.

Inspired by the commonsense reasoning examples in the Winograd Schema Challenge

Task of three supporting facts and Yes/No questions are prerequisites.

| 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                |

| The football fits in the suitcase.                                   |
| The suitcase fits in the cupboard.                                   |
| The box of chocolates is smaller than the football.                 |
| Will the box of chocolates fit in the suitcase? A:yes               |
bAbI Dataset

Path Finding
Questions in which the model learns to find a path between two locations.

The kitchen is north of the hallway.
The den is east of the hallway.
How do you go from den to kitchen?  A: west, north

Path Finding for MemNNs should be hard because it effectively involves search.
Agent’s Motivation

Questions in which the model learns to find the reason behind an agent’s action

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
MemNNs on bAbI

Baselines
Structured SVM with a collection of hand coded features - classic NLP stack
LSTM
ngram classifiers
### MemNNs on bAbI

#### Baselines

Structured SVM with a collection of hand-coded features - classic NLP stack

LSTM

ngram classifiers

<table>
<thead>
<tr>
<th>TASK</th>
<th>Weakly Supervised</th>
<th>Uses External Resources</th>
<th>Strong Supervision (using supporting facts)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>N-gram Classifier</td>
<td>LSTM</td>
<td>MemNN Hardware et al. (2015)</td>
</tr>
<tr>
<td>1 - Single Supporting Fact</td>
<td>36</td>
<td>50</td>
<td>99</td>
</tr>
<tr>
<td>2 - Two Supporting Facts</td>
<td>2</td>
<td>20</td>
<td>74</td>
</tr>
<tr>
<td>3 - Three Supporting Facts</td>
<td>7</td>
<td>20</td>
<td>17</td>
</tr>
<tr>
<td>4 - Two Arg. Relations</td>
<td>50</td>
<td>61</td>
<td>98</td>
</tr>
<tr>
<td>5 - Three Arg. Relations</td>
<td>20</td>
<td>70</td>
<td>52</td>
</tr>
<tr>
<td>6 - Yes/No Questions</td>
<td>49</td>
<td>48</td>
<td>53</td>
</tr>
<tr>
<td>7 - Counting</td>
<td>52</td>
<td>49</td>
<td>86</td>
</tr>
<tr>
<td>8 - Lists/Sets</td>
<td>40</td>
<td>45</td>
<td>88</td>
</tr>
<tr>
<td>9 - Simple Negation</td>
<td>62</td>
<td>64</td>
<td>94</td>
</tr>
<tr>
<td>10 - Indefinite Knowledge</td>
<td>45</td>
<td>44</td>
<td>100</td>
</tr>
<tr>
<td>11 - Basic Coreference</td>
<td>29</td>
<td>72</td>
<td>100</td>
</tr>
<tr>
<td>12 - Conjunction</td>
<td>9</td>
<td>74</td>
<td>100</td>
</tr>
<tr>
<td>13 - Compound Coref.</td>
<td>26</td>
<td>94</td>
<td>100</td>
</tr>
<tr>
<td>14 - Time Reasoning</td>
<td>19</td>
<td>27</td>
<td>100</td>
</tr>
<tr>
<td>15 - Basic Deduction</td>
<td>20</td>
<td>21</td>
<td>100</td>
</tr>
<tr>
<td>16 - Basic Induction</td>
<td>43</td>
<td>23</td>
<td>100</td>
</tr>
<tr>
<td>17 - Positional Reasoning</td>
<td>46</td>
<td>51</td>
<td>100</td>
</tr>
<tr>
<td>18 - Size Reasoning</td>
<td>52</td>
<td>52</td>
<td>100</td>
</tr>
<tr>
<td>19 - Path Finding</td>
<td>0</td>
<td>8</td>
<td>100</td>
</tr>
<tr>
<td>20 - Agent’s Motivations</td>
<td>76</td>
<td>91</td>
<td>100</td>
</tr>
<tr>
<td>Mean Performance</td>
<td>34</td>
<td>49</td>
<td>79</td>
</tr>
</tbody>
</table>

#### Accuracy:

- **Mean Performance**: 79%
- **Test accuracy (%) on our 20 Tasks for various methods**: (1000 training examples each)

#### MultiTask Training

- **AM + NG + NL**: 100%
- **AM + NONLINEAR**: 100%
- **AM + NG + NL + Failure**: 100%

#### MemNN Hardware et al. (2015)

- **Additive Memory**: 100%
- **MemNN Adaptive Memory**: 100%
MemNNs on bAbI

Baselines

Structured SVM with a collection of hand coded features - classic NLP stack

LSTM

ngram classifiers

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - Single Supporting Fact</td>
<td>36</td>
<td>50</td>
<td>99</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>250 ex.</td>
</tr>
<tr>
<td>2 - Two Supporting Facts</td>
<td>2</td>
<td>20</td>
<td>74</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>500 ex.</td>
</tr>
<tr>
<td>3 - Three Supporting Facts</td>
<td>7</td>
<td>20</td>
<td>17</td>
<td>20</td>
<td><strong>100</strong></td>
<td>99</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>500 ex.</td>
</tr>
<tr>
<td>4 - Two Arg. Relations</td>
<td>50</td>
<td>61</td>
<td>98</td>
<td>71</td>
<td>69</td>
<td><strong>100</strong></td>
<td>73</td>
<td>73</td>
<td>73</td>
<td>500 ex.</td>
</tr>
<tr>
<td>5 - Three Arg. Relations</td>
<td>20</td>
<td>70</td>
<td>83</td>
<td>83</td>
<td>83</td>
<td>86</td>
<td>86</td>
<td>86</td>
<td>98</td>
<td>1000 ex.</td>
</tr>
<tr>
<td>6 - Yes/No Questions</td>
<td>49</td>
<td>48</td>
<td>99</td>
<td>47</td>
<td>52</td>
<td>53</td>
<td><strong>100</strong></td>
<td><strong>100</strong></td>
<td><strong>100</strong></td>
<td>500 ex.</td>
</tr>
<tr>
<td>7 - Counting</td>
<td>52</td>
<td>49</td>
<td>69</td>
<td>68</td>
<td>78</td>
<td>86</td>
<td>83</td>
<td>85</td>
<td>FAIL</td>
<td>86</td>
</tr>
<tr>
<td>8 - Lists/Sets</td>
<td>40</td>
<td>45</td>
<td>70</td>
<td>77</td>
<td>90</td>
<td>88</td>
<td>94</td>
<td>91</td>
<td>FAIL</td>
<td>93</td>
</tr>
<tr>
<td>9 - Simple Negation</td>
<td>62</td>
<td>64</td>
<td>100</td>
<td>65</td>
<td>71</td>
<td>63</td>
<td><strong>100</strong></td>
<td><strong>100</strong></td>
<td><strong>100</strong></td>
<td>500 ex.</td>
</tr>
<tr>
<td>10 - Indefinite Knowledge</td>
<td>45</td>
<td>44</td>
<td>99</td>
<td>59</td>
<td>57</td>
<td>54</td>
<td>97</td>
<td><strong>98</strong></td>
<td>1000 ex.</td>
<td><strong>98</strong></td>
</tr>
<tr>
<td>11 - Basic Coreference</td>
<td>29</td>
<td>72</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>250 ex.</td>
</tr>
<tr>
<td>12 - Conjunction</td>
<td>9</td>
<td>74</td>
<td>96</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>250 ex.</td>
</tr>
<tr>
<td>13 - Compound Coref.</td>
<td>26</td>
<td>94</td>
<td>99</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>250 ex.</td>
</tr>
<tr>
<td>14 - Time Reasoning</td>
<td>19</td>
<td>27</td>
<td>99</td>
<td>99</td>
<td>100</td>
<td>99</td>
<td>100</td>
<td>99</td>
<td>100</td>
<td>500 ex.</td>
</tr>
<tr>
<td>15 - Basic Deduction</td>
<td>20</td>
<td>21</td>
<td>96</td>
<td>74</td>
<td>73</td>
<td><strong>100</strong></td>
<td><strong>100</strong></td>
<td>77</td>
<td>77</td>
<td>100 ex.</td>
</tr>
<tr>
<td>16 - Basic Induction</td>
<td>43</td>
<td>23</td>
<td>24</td>
<td>27</td>
<td><strong>100</strong></td>
<td><strong>100</strong></td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100 ex.</td>
</tr>
<tr>
<td>17 - Positional Reasoning</td>
<td>46</td>
<td>51</td>
<td>61</td>
<td>54</td>
<td>46</td>
<td>49</td>
<td>57</td>
<td>65</td>
<td>FAIL</td>
<td>72</td>
</tr>
<tr>
<td>18 - Size Reasoning</td>
<td>52</td>
<td>52</td>
<td>62</td>
<td>57</td>
<td>50</td>
<td>74</td>
<td>54</td>
<td>95</td>
<td>1000 ex.</td>
<td>93</td>
</tr>
<tr>
<td>19 - Path Finding</td>
<td>0</td>
<td>8</td>
<td>49</td>
<td>0</td>
<td>9</td>
<td>3</td>
<td>15</td>
<td>36</td>
<td>FAIL</td>
<td>19</td>
</tr>
<tr>
<td>20 - Agent’s Motivations</td>
<td>76</td>
<td>91</td>
<td>95</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>250 ex.</td>
</tr>
</tbody>
</table>

Mean Performance | 34 | 49 | 79 | 75 | 79 | 83 | 87 | 93 | 92 |
MemNNs on bAbI

Baselines
Structured SVM with a collection of hand-coded features - classic NLP stack
LSTM
ngram classifiers

<table>
<thead>
<tr>
<th>TASK</th>
<th>Weakly Supervised</th>
<th>Uses External Resources</th>
<th>Strong Supervision (using supporting facts)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>N-gram Classifier</td>
<td>LSTM</td>
<td>Structured SVM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 - Single Supporting Fact</td>
<td>36</td>
<td>50</td>
<td>99</td>
</tr>
<tr>
<td>2 - Two Supporting Facts</td>
<td>2</td>
<td>20</td>
<td>74</td>
</tr>
<tr>
<td>3 - Three Supporting Facts</td>
<td>7</td>
<td>20</td>
<td>17</td>
</tr>
<tr>
<td>4 - Two Arg. Relations</td>
<td>50</td>
<td>61</td>
<td>98</td>
</tr>
<tr>
<td>5 - Three Arg. Relations</td>
<td>20</td>
<td>70</td>
<td>83</td>
</tr>
<tr>
<td>6 - Yes/No Questions</td>
<td>49</td>
<td>48</td>
<td>99</td>
</tr>
<tr>
<td>7 - Counting</td>
<td>52</td>
<td>49</td>
<td>69</td>
</tr>
<tr>
<td>8 - Lists/Sets</td>
<td>40</td>
<td>45</td>
<td>70</td>
</tr>
<tr>
<td>9 - Simple Negation</td>
<td>62</td>
<td>64</td>
<td>100</td>
</tr>
<tr>
<td>10 - Indefinite Knowledge</td>
<td>45</td>
<td>44</td>
<td>99</td>
</tr>
<tr>
<td>11 - Basic Coreference</td>
<td>29</td>
<td>72</td>
<td>100</td>
</tr>
<tr>
<td>12 - Conjunction</td>
<td>9</td>
<td>74</td>
<td>96</td>
</tr>
<tr>
<td>13 - Compound Coref.</td>
<td>26</td>
<td>94</td>
<td>99</td>
</tr>
<tr>
<td>14 - Time Reasoning</td>
<td>19</td>
<td>27</td>
<td>99</td>
</tr>
<tr>
<td>15 - Basic Deduction</td>
<td>20</td>
<td>21</td>
<td>96</td>
</tr>
<tr>
<td>16 - Basic Induction</td>
<td>43</td>
<td>23</td>
<td>24</td>
</tr>
<tr>
<td>17 - Positional Reasoning</td>
<td>46</td>
<td>51</td>
<td>61</td>
</tr>
<tr>
<td>18 - Size Reasoning</td>
<td>52</td>
<td>52</td>
<td>62</td>
</tr>
<tr>
<td>19 - Path Finding</td>
<td>0</td>
<td>8</td>
<td>49</td>
</tr>
<tr>
<td>20 - Agent’s Motivations</td>
<td>76</td>
<td>91</td>
<td>95</td>
</tr>
<tr>
<td>Mean Performance</td>
<td>34</td>
<td>49</td>
<td>79</td>
</tr>
</tbody>
</table>
MemNNs on bAbI

Baselines
Structured SVM with a collection of hand coded features - classic NLP stack
LSTM ngram classifiers

<table>
<thead>
<tr>
<th>TASK</th>
<th>Weakly Supervised</th>
<th>Uses External Resources</th>
<th>Strong Supervision (using supporting facts)</th>
</tr>
</thead>
<tbody>
<tr>
<td>N-gram Classifier LSTM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 - Single Supporting Fact</td>
<td>36 50</td>
<td>99</td>
<td>100</td>
</tr>
<tr>
<td>2 - Two Supporting Facts</td>
<td>2 20</td>
<td>74</td>
<td>100</td>
</tr>
<tr>
<td>3 - Three Supporting Facts</td>
<td>7 20</td>
<td>17</td>
<td>20</td>
</tr>
<tr>
<td>4 - Two Arg. Relations</td>
<td>50 61</td>
<td>98</td>
<td>71</td>
</tr>
<tr>
<td>5 - Three Arg. Relations</td>
<td>20 70</td>
<td>83</td>
<td>83</td>
</tr>
<tr>
<td>6 - Yes/No Questions</td>
<td>49 48</td>
<td>99</td>
<td>47</td>
</tr>
<tr>
<td>7 - Counting</td>
<td>52 49</td>
<td>69</td>
<td>68</td>
</tr>
<tr>
<td>8 - Lists/Sets</td>
<td>40 45</td>
<td>70</td>
<td>77</td>
</tr>
<tr>
<td>9 - Simple Negation</td>
<td>62 64</td>
<td>100</td>
<td>65</td>
</tr>
<tr>
<td>10 - Indefinite Knowledge</td>
<td>45 44</td>
<td>99</td>
<td>59</td>
</tr>
<tr>
<td>11 - Basic Coreference</td>
<td>29 72</td>
<td>100</td>
<td>59</td>
</tr>
<tr>
<td>12 - Conjunction</td>
<td>9 74</td>
<td>96</td>
<td>100</td>
</tr>
<tr>
<td>13 - Compound Coref.</td>
<td>26 94</td>
<td>99</td>
<td>100</td>
</tr>
<tr>
<td>14 - Time Reasoning</td>
<td>19 27</td>
<td>99</td>
<td>99</td>
</tr>
<tr>
<td>15 - Basic Deduction</td>
<td>20 21</td>
<td>96</td>
<td>74</td>
</tr>
<tr>
<td>16 - Basic Induction</td>
<td>43 23</td>
<td>24</td>
<td>27</td>
</tr>
<tr>
<td>17 - Positional Reasoning</td>
<td>46 51</td>
<td>61</td>
<td>54</td>
</tr>
<tr>
<td>18 - Size Reasoning</td>
<td>52 52</td>
<td>62</td>
<td>57</td>
</tr>
<tr>
<td>19 - Path Finding</td>
<td>0 8</td>
<td>49</td>
<td>0</td>
</tr>
<tr>
<td>20 - Agent’s Motivations</td>
<td>76 91</td>
<td>95</td>
<td>100</td>
</tr>
<tr>
<td>Mean Performance</td>
<td>34 49</td>
<td>79</td>
<td>75</td>
</tr>
</tbody>
</table>

Table 3: Under review as a conference paper at ICLR 2016

Mean Test Accuracy (%) on our 20 Tasks for various methods (1000 training examples)

<table>
<thead>
<tr>
<th>TASK</th>
<th>Weakly Supervised</th>
<th>Uses External Resources</th>
<th>Strong Supervision (using supporting facts)</th>
</tr>
</thead>
<tbody>
<tr>
<td>N-gram Classifier LSTM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 - Single Supporting Fact</td>
<td>36 50</td>
<td>99</td>
<td>100</td>
</tr>
<tr>
<td>2 - Two Supporting Facts</td>
<td>2 20</td>
<td>74</td>
<td>100</td>
</tr>
<tr>
<td>3 - Three Supporting Facts</td>
<td>7 20</td>
<td>17</td>
<td>20</td>
</tr>
<tr>
<td>4 - Two Arg. Relations</td>
<td>50 61</td>
<td>98</td>
<td>71</td>
</tr>
<tr>
<td>5 - Three Arg. Relations</td>
<td>20 70</td>
<td>83</td>
<td>83</td>
</tr>
<tr>
<td>6 - Yes/No Questions</td>
<td>49 48</td>
<td>99</td>
<td>47</td>
</tr>
<tr>
<td>7 - Counting</td>
<td>52 49</td>
<td>69</td>
<td>68</td>
</tr>
<tr>
<td>8 - Lists/Sets</td>
<td>40 45</td>
<td>70</td>
<td>77</td>
</tr>
<tr>
<td>9 - Simple Negation</td>
<td>62 64</td>
<td>100</td>
<td>65</td>
</tr>
<tr>
<td>10 - Indefinite Knowledge</td>
<td>45 44</td>
<td>99</td>
<td>59</td>
</tr>
<tr>
<td>11 - Basic Coreference</td>
<td>29 72</td>
<td>100</td>
<td>59</td>
</tr>
<tr>
<td>12 - Conjunction</td>
<td>9 74</td>
<td>96</td>
<td>100</td>
</tr>
<tr>
<td>13 - Compound Coref.</td>
<td>26 94</td>
<td>99</td>
<td>100</td>
</tr>
<tr>
<td>14 - Time Reasoning</td>
<td>19 27</td>
<td>99</td>
<td>99</td>
</tr>
<tr>
<td>15 - Basic Deduction</td>
<td>20 21</td>
<td>96</td>
<td>74</td>
</tr>
<tr>
<td>16 - Basic Induction</td>
<td>43 23</td>
<td>24</td>
<td>27</td>
</tr>
<tr>
<td>17 - Positional Reasoning</td>
<td>46 51</td>
<td>61</td>
<td>54</td>
</tr>
<tr>
<td>18 - Size Reasoning</td>
<td>52 52</td>
<td>62</td>
<td>57</td>
</tr>
<tr>
<td>19 - Path Finding</td>
<td>0 8</td>
<td>49</td>
<td>0</td>
</tr>
<tr>
<td>20 - Agent’s Motivations</td>
<td>76 91</td>
<td>95</td>
<td>100</td>
</tr>
<tr>
<td>Mean Performance</td>
<td>34 49</td>
<td>79</td>
<td>75</td>
</tr>
</tbody>
</table>
Structured SVM with a classic NLP stack

Baselines

Structured SVM with a collection of hand-coded features - classic NLP stack

LSTM

ngram classifiers

<table>
<thead>
<tr>
<th>TASK</th>
<th>Weakly Supervised</th>
<th>Uses External Resources</th>
<th>Strong Supervision (using supporting facts)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>N-gram Classifier</td>
<td>LSTM</td>
<td>Structured SVM</td>
</tr>
<tr>
<td>1 - Single Supporting Fact</td>
<td>36</td>
<td>50</td>
<td>99</td>
</tr>
<tr>
<td>2 - Two Supporting Facts</td>
<td>2</td>
<td>20</td>
<td>74</td>
</tr>
<tr>
<td>3 - Three Supporting Facts</td>
<td>7</td>
<td>20</td>
<td>17</td>
</tr>
<tr>
<td>4 - Two Arg. Relations</td>
<td>50</td>
<td>61</td>
<td>98</td>
</tr>
<tr>
<td>5 - Three Arg. Relations</td>
<td>20</td>
<td>70</td>
<td>83</td>
</tr>
<tr>
<td>6 - Yes/No Questions</td>
<td>49</td>
<td>48</td>
<td>99</td>
</tr>
<tr>
<td>7 - Counting</td>
<td>52</td>
<td>49</td>
<td>69</td>
</tr>
<tr>
<td>8 - Lists/Sets</td>
<td>40</td>
<td>45</td>
<td>70</td>
</tr>
<tr>
<td>9 - Simple Negation</td>
<td>62</td>
<td>64</td>
<td>100</td>
</tr>
<tr>
<td>10 - Indefinite Knowledge</td>
<td>45</td>
<td>44</td>
<td>99</td>
</tr>
<tr>
<td>11 - Basic Coreference</td>
<td>29</td>
<td>72</td>
<td>100</td>
</tr>
<tr>
<td>12 - Conjunction</td>
<td>9</td>
<td>74</td>
<td>96</td>
</tr>
<tr>
<td>13 - Compound Coref.</td>
<td>26</td>
<td>94</td>
<td>99</td>
</tr>
<tr>
<td>14 - Time Reasoning</td>
<td>19</td>
<td>27</td>
<td>99</td>
</tr>
<tr>
<td>15 - Basic Deduction</td>
<td>20</td>
<td>21</td>
<td>96</td>
</tr>
<tr>
<td>16 - Basic Induction</td>
<td>43</td>
<td>23</td>
<td>24</td>
</tr>
<tr>
<td>17 - Positional Reasoning</td>
<td>46</td>
<td>51</td>
<td>61</td>
</tr>
<tr>
<td>18 - Size Reasoning</td>
<td>52</td>
<td>52</td>
<td>62</td>
</tr>
<tr>
<td>19 - Path Finding</td>
<td>0</td>
<td>8</td>
<td>49</td>
</tr>
<tr>
<td>20 - Agent’s Motivations</td>
<td>76</td>
<td>91</td>
<td>95</td>
</tr>
</tbody>
</table>

Mean Performance: 34, 49, 79, 75, 79, 83, 87, 92
Full Supervision in MemNNs

John was in the bathroom.
Bob was in the office.
John went to kitchen.
Bob travelled back home.
Where is John? A: kitchen
Full Supervision in MemNNs

John was in the bathroom.
Bob was in the office.
John went to kitchen.
Bob travelled back home.

Where is John? \text{A: kitchen}

$m_i = f(\text{John was in the bathroom.})$
$m_{i+1} = f(\text{Bob was in the office.})$
$m_{i+2} = f(\text{John went to the kitchen.})$
$m_{i+3} = f(\text{Bob travelled back home.})$

$x = f(\text{Where is John?})$
Full Supervision in MemNNs

John was in the bathroom.
Bob was in the office.
John went to kitchen.
Bob travelled back home.
Where is John? A: kitchen

Context
Supporting Fact
Question, Answer Pair

\[ x = f(\text{Where is John?}) \]

\[ m_i = f(\text{John was in the bathroom.}) \]
\[ m_{i+1} = f(\text{Bob was in the office.}) \]
\[ m_{i+2} = f(\text{John went to the kitchen.}) \]
\[ m_{i+3} = f(\text{Bob travelled back home.}) \]
Full Supervision in MemNNs

John was in the bathroom.
Bob was in the office.
John went to kitchen.
Bob travelled back home.
Where is John? A: kitchen

\[ x = f(\text{Where is John?}) \]

\[ m_i = f(\text{John was in the bathroom.}) \]
\[ m_{i+1} = f(\text{Bob was in the office.}) \]
\[ m_{i+2} = f(\text{John went to the kitchen.}) \]
\[ m_{i+3} = f(\text{Bob travelled back home.}) \]
Full Supervision in MemNNs

John was in the bathroom.
Bob was in the office.
John went to kitchen.
Bob travelled back home.
Where is John? A: kitchen

That’s your retrieved memory whose score you want to push higher
Full Supervision in MemNNs

John was in the bathroom.
Bob was in the office.
John went to kitchen.
Bob travelled back home.
Where is John? A: kitchen

That’s your retrieved memory whose score you want to push higher

This is like hard attention except that you already know where to attend!
Full Supervision in MemNNs

**Drawbacks**

- Fairly hard assumption to make
- Not the most natural scenario
- Expensive to get such data in real world

This is like hard attention except that you already know where to attend!
End2End MemNNs

No current supporting fact supplied

Learns which parts of the memory are relevant

This is achieved by reading using soft attention as opposed to hard

Performs multiple lookups to refine its guess about memory relevance

The whole architecture is end-to-end differentiable

Only needs supervision at the final output
We now extend our model to handle different ways to combine inputs. During training, all three embedding matrices are jointly learned by minimizing a standard cross-entropy loss between the match between input embedding \( x_i \) and each memory \( m_i \).

Input memory representation:
\[
(z_i) = A \cdot x_i
\]

Output memory representation:
\[
(p_i) = C \cdot m_i
\]

Because the function from input to output is smooth, we can easily compute gradients and backpropagate through it. Other recently proposed forms of memory or attention take this approach, notably Bahdanau et al. [2] and Graves [8], see also [9].

Suppose we are given an input set \( \{x_i\} \) and a question \( q \). Defined in this way, the simplest case by another embedding matrix
\[
(z_i) = A \cdot x_i
\]

\[
(p_i) = C \cdot m_i
\]

Generating the final prediction:
\[
p_i = \text{Softmax}(u^T m_i)
\]

Single Layer
End2End MemNNs

The overall model is shown in Fig. 1(a). During training, all three embedding matrices are jointly learned by minimizing a standard cross-entropy loss between the predicted label and the true label.

We now extend our model to handle multiple layers. We can constrain several of the embedding matrices to be the same (see Section 2.2).

Figure 1: (a): A single layer version of our model. (b): A three layer version of our model. In the simplest case, an input embedding matrix is used to generate a memory representation. The memory is then used to retrieve information relevant to the input and generate a prediction. Because the function from input to output is smooth, we can easily compute gradients and backtrack to adjust the input embedding matrix.

In the single layer case, the sum of the output vector is used to compute a probability distribution over the inputs.

The entire set of memory vectors is then passed through a final weight matrix to produce the final prediction.

```
\[ p_i = \text{Softmax}(u^T m_i) \]
```

```
\[ o = \sum_i p_i c_i \]
```

Single Layer
End2End MemNNs

The overall model is shown in Fig. 1(a). During training, all three embedding matrices are jointly learned by minimizing a standard cross-entropy loss between the match between \( u \) and \( c \).

\[
p_i = \text{Softmax}(u^Tm_i)
\]

\[
o = \sum_i p_i c_i
\]

\[
\hat{a} = \text{Softmax}(W(o + u))
\]
End2End MemNNs

\[
\hat{a} = \text{Softmax}(W \cdot k^{+1}u) = \text{Softmax}(W(ko + ku))
\]

\[
kpi = \text{Softmax}(ku^T \cdot km_i)
\]

\[
k_o = \sum_i kpikc_i
\]

\[
k^{+1}u = ku + ko
\]
E2EMemNNs: Other Details

Share the input and output embeddings or not

What to store in memories — individual words, word windows, full sentences

How to represent the memories — bag-or-words, RNN style reading at words or characters

Positional Encodings - instead of modeling the sentence as a bag, the word position was modeled by a multiplicative weights on each word vector with the value of the weight being depended on the position.
# E2E MemNNs: bAbI

<table>
<thead>
<tr>
<th>TASK</th>
<th>N-grams</th>
<th>LSTMs</th>
<th>MemN2N</th>
<th>Memory Networks</th>
<th>StructSVM +coref+srl</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1. Single supporting fact</td>
<td>36</td>
<td>50</td>
<td>PASS</td>
<td>PASS</td>
<td>PASS</td>
</tr>
<tr>
<td>T2. Two supporting facts</td>
<td>2</td>
<td>20</td>
<td>87</td>
<td>PASS</td>
<td>74</td>
</tr>
<tr>
<td>T3. Three supporting facts</td>
<td>7</td>
<td>20</td>
<td>60</td>
<td>PASS</td>
<td>17</td>
</tr>
<tr>
<td>T4. Two arguments relations</td>
<td>50</td>
<td>61</td>
<td>PASS</td>
<td>PASS</td>
<td>PASS</td>
</tr>
<tr>
<td>T5. Three arguments relations</td>
<td>20</td>
<td>70</td>
<td>87</td>
<td>PASS</td>
<td>83</td>
</tr>
<tr>
<td>T6. Yes/no questions</td>
<td>49</td>
<td>48</td>
<td>92</td>
<td>PASS</td>
<td>17</td>
</tr>
<tr>
<td>T7. Counting</td>
<td>52</td>
<td>49</td>
<td>83</td>
<td>85</td>
<td>69</td>
</tr>
<tr>
<td>T8. Sets</td>
<td>40</td>
<td>45</td>
<td>90</td>
<td>91</td>
<td>70</td>
</tr>
<tr>
<td>T9. Simple negation</td>
<td>62</td>
<td>64</td>
<td>87</td>
<td>PASS</td>
<td>PASS</td>
</tr>
<tr>
<td>T10. Indefinite knowledge</td>
<td>45</td>
<td>44</td>
<td>85</td>
<td>PASS</td>
<td>PASS</td>
</tr>
<tr>
<td>T11. Basic coreference</td>
<td>29</td>
<td>72</td>
<td>PASS</td>
<td>PASS</td>
<td>PASS</td>
</tr>
<tr>
<td>T12. Conjunction</td>
<td>9</td>
<td>74</td>
<td>PASS</td>
<td>PASS</td>
<td>PASS</td>
</tr>
<tr>
<td>T13. Compound coreference</td>
<td>26</td>
<td>PASS</td>
<td>PASS</td>
<td>PASS</td>
<td>PASS</td>
</tr>
<tr>
<td>T14. Time reasoning</td>
<td>19</td>
<td>27</td>
<td>PASS</td>
<td>PASS</td>
<td>PASS</td>
</tr>
<tr>
<td>T15. Basic deduction</td>
<td>20</td>
<td>21</td>
<td>PASS</td>
<td>PASS</td>
<td>PASS</td>
</tr>
<tr>
<td>T16. Basic induction</td>
<td>43</td>
<td>23</td>
<td>PASS</td>
<td>PASS</td>
<td>24</td>
</tr>
<tr>
<td>T17. Positional reasoning</td>
<td>46</td>
<td>51</td>
<td>49</td>
<td>65</td>
<td>61</td>
</tr>
<tr>
<td>T18. Size reasoning</td>
<td>52</td>
<td>52</td>
<td>89</td>
<td>PASS</td>
<td>62</td>
</tr>
<tr>
<td>T19. Path finding</td>
<td>0</td>
<td>8</td>
<td>7</td>
<td>36</td>
<td>49</td>
</tr>
<tr>
<td>T20. Agent's motivation</td>
<td>76</td>
<td>91</td>
<td>PASS</td>
<td>PASS</td>
<td>PASS</td>
</tr>
</tbody>
</table>

- **Weakly supervised**
- **Supervised Supp. Facts**
E2EMemNNs: bAbI

Samples from toy QA tasks

<table>
<thead>
<tr>
<th>Story (1: 1 supporting fact)</th>
<th>Support</th>
<th>Hop 1</th>
<th>Hop 2</th>
<th>Hop 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Daniel went to the bathroom.</td>
<td>0.00</td>
<td>0.00</td>
<td>0.03</td>
<td></td>
</tr>
<tr>
<td>Mary travelled to the hallway.</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>John went to the bedroom.</td>
<td>0.37</td>
<td>0.02</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>John travelled to the bathroom.</td>
<td>0.60</td>
<td>0.98</td>
<td>0.96</td>
<td></td>
</tr>
<tr>
<td>Mary went to the office.</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
</tbody>
</table>

Where is John? Answer: bathroom Prediction: bathroom

<table>
<thead>
<tr>
<th>Story (2: 2 supporting facts)</th>
<th>Support</th>
<th>Hop 1</th>
<th>Hop 2</th>
<th>Hop 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>John dropped the milk.</td>
<td>0.06</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>John took the milk there.</td>
<td>yes</td>
<td>0.88</td>
<td>1.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Sandra went back to the bathroom.</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>John moved to the hallway.</td>
<td>yes</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Mary went back to the bedroom.</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
</tbody>
</table>

Where is the milk? Answer: hallway Prediction: hallway

<table>
<thead>
<tr>
<th>Story (16: basic induction)</th>
<th>Support</th>
<th>Hop 1</th>
<th>Hop 2</th>
<th>Hop 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brian is a frog.</td>
<td>yes</td>
<td>0.00</td>
<td>0.98</td>
<td>0.00</td>
</tr>
<tr>
<td>Lily is gray.</td>
<td>0.07</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>Brian is yellow.</td>
<td>yes</td>
<td>0.07</td>
<td>0.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Julius is green.</td>
<td>0.06</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>Greg is a frog.</td>
<td>yes</td>
<td>0.76</td>
<td>0.02</td>
<td>0.00</td>
</tr>
</tbody>
</table>

What color is Greg? Answer: yellow Prediction: yellow

<table>
<thead>
<tr>
<th>Story (18: size reasoning)</th>
<th>Support</th>
<th>Hop 1</th>
<th>Hop 2</th>
<th>Hop 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>The suitcase is bigger than the chest.</td>
<td>yes</td>
<td>0.00</td>
<td>0.88</td>
<td>0.00</td>
</tr>
<tr>
<td>The box is bigger than the chocolate.</td>
<td>0.04</td>
<td>0.05</td>
<td>0.10</td>
<td></td>
</tr>
<tr>
<td>The chest is bigger than the chocolate.</td>
<td>yes</td>
<td>0.17</td>
<td>0.07</td>
<td>0.90</td>
</tr>
<tr>
<td>The chest fits inside the container.</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>The chest fits inside the box.</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
</tbody>
</table>

Does the suitcase fit in the chocolate? Answer: no Prediction: no

20 bAbI Tasks

<table>
<thead>
<tr>
<th>Model</th>
<th>Test Acc</th>
<th>Failed tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>MemNN</td>
<td>93.3%</td>
<td>4</td>
</tr>
<tr>
<td>LSTM</td>
<td>49%</td>
<td>20</td>
</tr>
<tr>
<td>MemN2N 1 hop</td>
<td>74.82%</td>
<td>17</td>
</tr>
<tr>
<td>2 hops</td>
<td>84.4%</td>
<td>11</td>
</tr>
<tr>
<td>3 hops</td>
<td>87.6%</td>
<td>11</td>
</tr>
</tbody>
</table>
E2EMemNNs: Language Modeling

Predict the next work given previous words in a word sequence.

Results on PennTree Bank and Text8 data (a subset of wikipedia)

<table>
<thead>
<tr>
<th></th>
<th>Penn Tree</th>
<th>Text8</th>
</tr>
</thead>
<tbody>
<tr>
<td>RNN</td>
<td>129</td>
<td>184</td>
</tr>
<tr>
<td>LSTM</td>
<td>115</td>
<td>154</td>
</tr>
<tr>
<td>MemN2N 2 hops</td>
<td>121</td>
<td>187</td>
</tr>
<tr>
<td>5 hops</td>
<td>118</td>
<td>154</td>
</tr>
<tr>
<td>7 hops</td>
<td>111</td>
<td>147</td>
</tr>
</tbody>
</table>

Hops vs. Attention: Average over (PTB)

Average over (Text8)
E2EMemNNs: Language Modeling

Same ballpark as LSTMs

For many words we don’t really need long term sequence

Might help for nouns or entities?

<table>
<thead>
<tr>
<th></th>
<th>Penn Tree</th>
<th>Text8</th>
</tr>
</thead>
<tbody>
<tr>
<td>RNN</td>
<td>129</td>
<td>184</td>
</tr>
<tr>
<td>LSTM</td>
<td>115</td>
<td>154</td>
</tr>
<tr>
<td>MemN2N 2 hops</td>
<td>121</td>
<td>187</td>
</tr>
<tr>
<td>5 hops</td>
<td>118</td>
<td>154</td>
</tr>
<tr>
<td>7 hops</td>
<td>111</td>
<td>147</td>
</tr>
</tbody>
</table>

Test perplexity

The goal is to predict the next word in a text sequence given the previous words. Results on the Penn Treebank and Text8 (Wikipedia-based) corpora.
Relevant Literature

RNNSearch (Bahdanau et. al.) for Machine Translation
Can be seen as a Memory Network with memory storing individual words and is only a single sentence long.
At inference it reads all the memories and performs Softmax to find best alignment. It is only 1 hop though.

Generating Sequences With RNNs (Graves., 13)
Also does alignment with previous sentence to generate handwriting

Neural Turing Machines (Graves at. al., 14)
Has read/write operations over fixed small sized memory.
Until recently has only been used for toy tasks - copy, sorting etc

Earlier works by Das et. al., 92, Schmidhuber et. al., 93, DISCERN by Miikkulainen, 90) and others fall into this category
Large Scale Memories

So far we’ve only dealt with limited sized memory module
Large Scale Memories

So far we’ve only dealt with limited sized memory module

<table>
<thead>
<tr>
<th>Movie</th>
<th>Director</th>
<th>Writer</th>
<th>Actor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shaolin Soccer</td>
<td>Stephen Chow</td>
<td>Stephen Chow</td>
<td>Stephen Chow</td>
</tr>
<tr>
<td>Kung Fu Hustle</td>
<td>Stephen Chow</td>
<td>Stephen Chow</td>
<td>Stephen Chow</td>
</tr>
<tr>
<td>The God of Cookery</td>
<td>Stephen Chow</td>
<td>Stephen Chow</td>
<td>Stephen Chow</td>
</tr>
<tr>
<td>From Beijing with Love</td>
<td>Stephen Chow</td>
<td>Stephen Chow</td>
<td>Anita Yuen</td>
</tr>
</tbody>
</table>
Large Scale Memories

Write into the memories more intelligently

During the write operation, hash the memories to store in buckets

The hash functions could be a function of words in the statement: buckets would correspond to topics

Or it could be a function of the embeddings of words

The result is you avoid reading from all the memories - not only it is inefficient, it is also hard to train
Reverb Dataset

Paraphrase Driven Learning for Open Question Answering: Fader et. al., 2013

14 million facts stored as triples [subject, relation, object]

Triples are REVERB extractions mined from ClueWeb09

Statements cover diverse topics:
[milne, authored, winnie-the-pooh]
[sheep, be-afraid-of, wolf]

Training set: weakly labeled QA pairs and 35M paraphrased questions from WikiAnswers

Who wrote the Winnie the Pooh books?
Who is Pooh’s creator?
MemNNs on Reverb Dataset

Paraphrase Driven Learning for Open Question Answering: Fader et. al., 2013

- 14 million facts stored in memory
- Single hop processing. Embedding dimension = 128
- Outputs top scoring statement
- Also tried adding BoW features

<table>
<thead>
<tr>
<th>Method</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Fader et al., 2013)</td>
<td>0.54</td>
</tr>
<tr>
<td>(Bordes et al., 2014)</td>
<td>0.73</td>
</tr>
<tr>
<td>MemNN</td>
<td>0.72</td>
</tr>
<tr>
<td>MemNN (with BoW features)</td>
<td>0.82</td>
</tr>
</tbody>
</table>
MemNNs on Reverb Dataset

QA reference - complete the reference

Scoring all 14 million facts in memory hard and slow

So we hash based on:

- Words in the statement: inverted index
- K-means in embedding space (k=1000)

<table>
<thead>
<tr>
<th>Method</th>
<th>Embedding</th>
<th>Embed+BoW</th>
<th>candidates</th>
</tr>
</thead>
<tbody>
<tr>
<td>MemNN (no hashing)</td>
<td>0.72</td>
<td>0.82</td>
<td>14M</td>
</tr>
<tr>
<td>MemNN (word hash)</td>
<td>0.63</td>
<td>0.68</td>
<td>13k (1000x)</td>
</tr>
<tr>
<td>MemNN (clust hash)</td>
<td>0.71</td>
<td>0.80</td>
<td>177k (80x)</td>
</tr>
</tbody>
</table>
Multitasked MemNNs: bAbI + Reverb

Story told to the model after training

Antoine went to the kitchen.
Antoine picked up the milk.
Antoine travelled to the office

Where is the milk? : office
Where was Antoine before the office?: kitchen
Where does milk come from?: milk come from cow
What is cow a type of?: cow be female of cattle
Where are cattle found?: cattle farm become widespread in Brazil
What does milk taste like?: milk taste like milk
What does milk go well with?: milk go with coffee
Cloze Style QA

Teaching a machine to understand language is hard

One way is to read a comprehension and answer questions pertaining to it

However the questions should be such that they cannot be answered using external world knowledge - Cloze Style QA

Until recently only small sized dataset existed - which were primarily used for testing - nothing to train on

Two primary efforts in this direction

Teaching Machines to Read and Comprehend: Hermann et. al. 2015

The Goldilocks Principle: Reading Children’s Books with Explicit Memory Representation: Hill et. al., 2015
CBT: Children’s Book Dataset

Dataset built from 118 freely available books from project Gutenberg

Children stories provide a clear narrative structure

Can make the role of context more salient

The Goldilocks Principle: Reading Children’s Books with Explicit Memory Representation: Hill et. al., 2015
A single word from the 21st sentence is removed, which becomes the answer.

The model must identify the answer word from a selection of 10 provided candidates.
MemNNs for Story Understanding

Figure: Jason Weston
MemNNs for Story Understanding

Figure: Jason Weston
## MemNNs for Story Understanding

The Goldilocks Principle: Reading Children’s Books with Explicit Memory Representation: Hill et. al., 2015

<table>
<thead>
<tr>
<th>METHODS</th>
<th>NAMED ENTITIES</th>
<th>COMMON NOUNS</th>
<th>VERBS</th>
<th>PREPOSITIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>HUMANS (QUERY) (*)</td>
<td>0.520</td>
<td>0.644</td>
<td>0.716</td>
<td>0.676</td>
</tr>
<tr>
<td>HUMANS (CONTEXT+QUERY) (*)</td>
<td>0.816</td>
<td>0.816</td>
<td>0.828</td>
<td>0.708</td>
</tr>
<tr>
<td>MAXIMUM FREQUENCY (CORPUS)</td>
<td>0.120</td>
<td>0.158</td>
<td>0.373</td>
<td>0.315</td>
</tr>
<tr>
<td>MAXIMUM FREQUENCY (CONTEXT)</td>
<td>0.335</td>
<td>0.281</td>
<td>0.285</td>
<td>0.275</td>
</tr>
<tr>
<td>SLIDING WINDOW</td>
<td>0.168</td>
<td>0.196</td>
<td>0.182</td>
<td>0.101</td>
</tr>
<tr>
<td>WORD DISTANCE MODEL</td>
<td>0.398</td>
<td>0.364</td>
<td>0.380</td>
<td>0.237</td>
</tr>
<tr>
<td>KNESEMER-NEY LANGUAGE MODEL</td>
<td>0.390</td>
<td>0.544</td>
<td>0.778</td>
<td>0.768</td>
</tr>
<tr>
<td>KNESEMER-NEY LANGUAGE MODEL + CACHE</td>
<td>0.439</td>
<td>0.577</td>
<td>0.772</td>
<td>0.679</td>
</tr>
<tr>
<td>EMBEDDING MODEL (CONTEXT+QUERY)</td>
<td>0.253</td>
<td>0.259</td>
<td>0.421</td>
<td>0.315</td>
</tr>
<tr>
<td>EMBEDDING MODEL (QUERY)</td>
<td>0.351</td>
<td>0.400</td>
<td>0.614</td>
<td>0.535</td>
</tr>
<tr>
<td>EMBEDDING MODEL (WINDOW)</td>
<td>0.362</td>
<td>0.415</td>
<td>0.637</td>
<td>0.589</td>
</tr>
<tr>
<td>EMBEDDING MODEL (WINDOW+POSITION)</td>
<td>0.402</td>
<td>0.506</td>
<td>0.736</td>
<td>0.670</td>
</tr>
<tr>
<td>LSTMs (QUERY)</td>
<td>0.408</td>
<td>0.541</td>
<td>0.813</td>
<td>0.802</td>
</tr>
<tr>
<td>LSTMs (CONTEXT+QUERY)</td>
<td>0.418</td>
<td>0.560</td>
<td>0.818</td>
<td>0.791</td>
</tr>
<tr>
<td>CONTEXTUAL LSTMs (WINDOW CONTEXT)</td>
<td>0.436</td>
<td>0.582</td>
<td>0.805</td>
<td>0.806</td>
</tr>
<tr>
<td>MEMNNs (LEXICAL MEMORY)</td>
<td>0.431</td>
<td>0.562</td>
<td>0.798</td>
<td>0.764</td>
</tr>
<tr>
<td>MEMNNs (WINDOW MEMORY)</td>
<td>0.493</td>
<td>0.554</td>
<td>0.692</td>
<td>0.674</td>
</tr>
<tr>
<td>MEMNNs (SENTENTIAL MEMORY + PE)</td>
<td>0.318</td>
<td>0.305</td>
<td>0.502</td>
<td>0.326</td>
</tr>
<tr>
<td>MEMNNs (WINDOW MEMORY + SELF-SUP.)</td>
<td><strong>0.666</strong></td>
<td><strong>0.630</strong></td>
<td>0.690</td>
<td>0.703</td>
</tr>
</tbody>
</table>
# MemNNs for Story Understanding

**The Goldilocks Principle: Reading Children’s Books with Explicit Memory Representation: Hill et. al., 2015**

<table>
<thead>
<tr>
<th>Methods</th>
<th>Named Entities</th>
<th>Common Nouns</th>
<th>Verbs</th>
<th>Prepositions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Humans (query)</strong>(*)</td>
<td>0.520</td>
<td>0.644</td>
<td>0.716</td>
<td>0.676</td>
</tr>
<tr>
<td><strong>Humans (context+query)</strong>(*)</td>
<td><strong>0.816</strong></td>
<td><strong>0.816</strong></td>
<td><strong>0.828</strong></td>
<td>0.708</td>
</tr>
<tr>
<td><strong>Maximum frequency (corpus)</strong></td>
<td>0.120</td>
<td>0.158</td>
<td>0.373</td>
<td>0.315</td>
</tr>
<tr>
<td><strong>Maximum frequency (context)</strong></td>
<td>0.335</td>
<td>0.281</td>
<td>0.285</td>
<td>0.275</td>
</tr>
<tr>
<td><strong>Sliding window</strong></td>
<td>0.168</td>
<td>0.196</td>
<td>0.182</td>
<td>0.101</td>
</tr>
<tr>
<td><strong>Word distance model</strong></td>
<td>0.398</td>
<td>0.364</td>
<td>0.380</td>
<td>0.237</td>
</tr>
<tr>
<td><strong>Kneser-Ney language model</strong></td>
<td>0.390</td>
<td>0.544</td>
<td>0.778</td>
<td>0.768</td>
</tr>
<tr>
<td><strong>Kneser-Ney language model + cache</strong></td>
<td>0.439</td>
<td>0.577</td>
<td>0.772</td>
<td>0.679</td>
</tr>
<tr>
<td><strong>Embedding Model (context+query)</strong></td>
<td>0.253</td>
<td>0.259</td>
<td>0.421</td>
<td>0.315</td>
</tr>
<tr>
<td><strong>Embedding Model (query)</strong></td>
<td>0.351</td>
<td>0.400</td>
<td>0.614</td>
<td>0.535</td>
</tr>
<tr>
<td><strong>Embedding Model (window)</strong></td>
<td>0.362</td>
<td>0.415</td>
<td>0.637</td>
<td>0.589</td>
</tr>
<tr>
<td><strong>Embedding Model (window+position)</strong></td>
<td>0.402</td>
<td>0.506</td>
<td>0.736</td>
<td>0.670</td>
</tr>
<tr>
<td><strong>LSTMs (query)</strong></td>
<td>0.408</td>
<td>0.541</td>
<td>0.813</td>
<td>0.802</td>
</tr>
<tr>
<td><strong>LSTMs (context+query)</strong></td>
<td>0.418</td>
<td>0.560</td>
<td><strong>0.818</strong></td>
<td>0.791</td>
</tr>
<tr>
<td><strong>Contextual LSTMs (window context)</strong></td>
<td>0.436</td>
<td>0.582</td>
<td>0.805</td>
<td><strong>0.806</strong></td>
</tr>
<tr>
<td><strong>MemNNs (lexical memory)</strong></td>
<td>0.431</td>
<td>0.562</td>
<td>0.798</td>
<td>0.764</td>
</tr>
<tr>
<td><strong>MemNNs (window memory)</strong></td>
<td>0.493</td>
<td>0.554</td>
<td>0.692</td>
<td>0.674</td>
</tr>
<tr>
<td><strong>MemNNs (sentential memory + PE)</strong></td>
<td>0.318</td>
<td>0.305</td>
<td>0.502</td>
<td>0.326</td>
</tr>
<tr>
<td><strong>MemNNs (window memory + self-sup.)</strong></td>
<td><strong>0.666</strong></td>
<td><strong>0.630</strong></td>
<td>0.690</td>
<td>0.703</td>
</tr>
</tbody>
</table>
Self Supervision in MemNNs

During training we have knowledge about the correct answer word.

We can treat all the memories in which the answer word appears as the relevant supporting fact.

Bump up the scores of these memories.

This speeds up training.

Of course this knowledge is not available at test time - so you simply pick the most relevant memory to generate your answer.
### QA on News Articles

**Teaching Machines to Read and Comprehend: Hermann et. al.2015**

<table>
<thead>
<tr>
<th></th>
<th>CNN</th>
<th>Daily Mail</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>train</td>
<td>valid</td>
</tr>
<tr>
<td># months</td>
<td>95</td>
<td>1</td>
</tr>
<tr>
<td># documents</td>
<td>90,266</td>
<td>1,220</td>
</tr>
<tr>
<td># queries</td>
<td>380,298</td>
<td>3,924</td>
</tr>
<tr>
<td>Max # entities</td>
<td>527</td>
<td>187</td>
</tr>
<tr>
<td>Avg # entities</td>
<td>26.4</td>
<td>26.5</td>
</tr>
<tr>
<td>Avg # tokens</td>
<td>762</td>
<td>763</td>
</tr>
<tr>
<td>Vocab size</td>
<td>118,497</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Corpus statistics. Articles were collected starting in April 2007 for CNN and June 2010 for the Daily Mail, both until the end of April 2015. Validation data is from March, test data from April 2015. Articles of over 2000 tokens and queries whose answer entity did not appear in the context were filtered out.

We evaluate our models on this dataset as well.
## QA on News Articles

**Teaching Machines to Read and Comprehend: Hermann et. al. 2015**

<table>
<thead>
<tr>
<th>Original Version</th>
<th>Anonymised Version</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Context</strong></td>
<td></td>
</tr>
<tr>
<td>The BBC producer allegedly struck by Jeremy Clarkson will not press charges against the “Top Gear” host, his lawyer said Friday. Clarkson, who hosted one of the most-watched television shows in the world, was dropped by the BBC Wednesday after an internal investigation by the British broadcaster found he had subjected producer Oisin Tymon to an unprovoked physical and verbal attack.” …</td>
<td>the ent381 producer allegedly struck by ent212 will not press charges against the “ent153” host, his lawyer said friday. ent212, who hosted one of the most-watched television shows in the world, was dropped by the ent381 wednesday after an internal investigation by the ent180 broadcaster found he had subjected producer ent193 “to an unprovoked physical and verbal attack.” …</td>
</tr>
<tr>
<td><strong>Query</strong></td>
<td></td>
</tr>
<tr>
<td>Producer X will not press charges against Jeremy Clarkson, his lawyer says.</td>
<td>producer X will not press charges against ent212, his lawyer says.</td>
</tr>
<tr>
<td><strong>Answer</strong></td>
<td></td>
</tr>
<tr>
<td>Oisin Tymon</td>
<td>ent193</td>
</tr>
</tbody>
</table>
# QA on News Articles

<table>
<thead>
<tr>
<th>Methods</th>
<th>Validation</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum frequency (article) *</td>
<td>0.305</td>
<td>0.332</td>
</tr>
<tr>
<td>Sliding window</td>
<td>0.005</td>
<td>0.006</td>
</tr>
<tr>
<td>Word distance model *</td>
<td>0.505</td>
<td>0.509</td>
</tr>
<tr>
<td>Deep LSTMs (article+query) *</td>
<td>0.550</td>
<td>0.570</td>
</tr>
<tr>
<td>Contextual LSTMs (“Attentive reader”) *</td>
<td>0.616</td>
<td>0.630</td>
</tr>
<tr>
<td>Contextual LSTMs (“Impatient reader”) *</td>
<td>0.618</td>
<td>0.638</td>
</tr>
<tr>
<td>MemNNs (window memory)</td>
<td>0.580</td>
<td>0.606</td>
</tr>
<tr>
<td>MemNNs (window memory + self-sup.)</td>
<td>0.634</td>
<td>0.668</td>
</tr>
<tr>
<td>MemNNs (window memory + ensemble)</td>
<td>0.612</td>
<td>0.638</td>
</tr>
<tr>
<td>MemNNs (window memory + self-sup. + ensemble)</td>
<td>0.649</td>
<td>0.684</td>
</tr>
<tr>
<td>MemNNs (window + self-sup. + ensemble + exclud. coocurrences)</td>
<td><strong>0.662</strong></td>
<td><strong>0.694</strong></td>
</tr>
</tbody>
</table>
Dialog Modeling

So far we have focused on a single step QA potentially with long term context

How about Dialog Modeling?

We have built another large scale dataset focussed towards movie domain

Ask about movies — Ask about movie recommendation — Have dialog which combines facts and opinions — General chit-chat about movies

75k entities, and 3.5M exchanges

Evaluating Prerequisite Qualities for Learning End-to-End Dialog Systems: Dodge et. al., 2016
Task 1: QA on Movies

What movies are about open source? Revolution OS
Ruggero Raimondi appears in which movies? Carmen
Can you name a film directed by Stuart Ortiz? Grave Encounters
Who directed the film White Elephant? Pablo Trapero
What is the genre of the film Dial M for Murder? Thriller, Crime
What language is Whity in? German

Evaluating Prerequisite Qualities for Learning End-to-End Dialog Systems: Dodge et. al., 2016
Dialog Modeling

Task 2: Movie Recommendation


Some movies I like are Heat, Kids, Fight Club, Shaun of the Dead, The Avengers, Skyfall, and Jurassic Park. Can you suggest something else I might like? Ocean's Eleven

Evaluating Prerequisite Qualities for Learning End-to-End Dialog Systems: Dodge et. al., 2016
Task 3: Combining QA and Movie Recommendation

I loved Billy Madison, Blades of Glory, Bio-Dome, Clue, and Happy Gilmore. I'm looking for a Music movie. School of Rock
What else is that about? Music, Musical, Jack Black, school, teacher, Richard Linklater, rock, guitar
I like rock and roll movies more. Do you know anything else? Little Richard

Evaluating Prerequisite Qualities for Learning End-to-End Dialog Systems: Dodge et. al., 2016
I think the Terminator movies really suck, I mean the first one was kinda ok, but after that they got really cheesy. Even the second one which people somehow think is great. And after that... forgeddabotit.
C’mon the second one was still pretty cool.. Arny was still so badass, as was Sararah Connor’s character.. and the way they blended real action and effects was perhaps the last of its kind...

Evaluating Prerequisite Qualities for Learning End-to-End Dialog Systems: Dodge et. al., 2016
Memory Networks for Dialog

Memories $h_i$

| Shaolin Soccer written by Stephen Chow |
| Shaolin Soccer starred_actors Stephen Chow |
| Shaolin Soccer release_year 2001 |
| Shaolin Soccer has_genre comedy |
| Shaolin Soccer has_tags martial arts, kung fu soccer, stephen chow |
| Kung Fu Hustle directed_by Stephen Chow |
| Kung Fu Hustle written_by Stephen Chow |
| Kung Fu Hustle starred_actors Stephen Chow |
| Kung Fu Hustle has_genre comedy action |
| Kung Fu Hustle has_imdb_votes famous |
| Kung Fu Hustle has_tags comedy, action, martial arts, kung fu, china, soccer, hong kong, stephen chow |
| The God of Cookery directed_by Stephen Chow |
| The God of Cookery written_by Stephen Chow |
| The God of Cookery starred_actors Stephen Chow |
| The God of Cookery has_tags hong kong Stephen Chow |
| From Beijing with Love directed_by Stephen Chow |
| From Beijing with Love written_by Stephen Chow |
| From Beijing with Love starred_actors Stephen Chow, Anita Yuen |

... <and more> ...

Short-Term $c_1^t$

1) I’m looking a fun comedy to watch tonight, any ideas?
2) Have you seen Shaolin Soccer? That was zany and great.. really funny but in a whacky way.

Memories $c_1^r$

3) Yes! Shaolin Soccer and Kung Fu Hustle are so good I really need to find some more Stephen Chow films I feel like there is more awesomeness out there that I haven’t discovered yet ...
# Results

<table>
<thead>
<tr>
<th>Methods</th>
<th>QA Task (Hits@1)</th>
<th>RECS Task (Hits@100)</th>
<th>QA+RECS Task (Hits@10)</th>
<th>Reddit Task (Hits@10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>QA System (Bordes et al., 2014)</td>
<td>90.7</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>SVD</td>
<td>N/A</td>
<td>19.2</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>IR</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>23.7</td>
</tr>
<tr>
<td>LSTM</td>
<td>6.5</td>
<td>27.1</td>
<td>19.9</td>
<td>11.8</td>
</tr>
<tr>
<td>Supervised Embeddings</td>
<td>50.9</td>
<td>29.2</td>
<td>65.9</td>
<td>27.6</td>
</tr>
<tr>
<td>MemN2N</td>
<td>79.3</td>
<td>28.6</td>
<td>81.7</td>
<td>29.2</td>
</tr>
<tr>
<td>Joint Supervised Embeddings</td>
<td>43.6</td>
<td>28.1</td>
<td>58.9</td>
<td>14.5</td>
</tr>
<tr>
<td>Joint MemN2N</td>
<td>83.5</td>
<td>26.5</td>
<td>78.9</td>
<td>26.6</td>
</tr>
</tbody>
</table>
Key-Value MemNNs

Key Value Memory Networks for Directly Reading Documents: Miller et. al., 2016

Facts are stored in a key value structured memory

Memory is designed so that the model learns to use keys to address relevant memories with respect to the question

Structure allows the model to encode prior knowledge for the considered task

Structure also allows to leverage possibly complex transforms between key and value

Example: for a KB triple [subject, relation, object], Key could be [subject,relation] and value could be [object] or vice versa
Key-Value MemNNs

What year was the movie Blade Runner released?

1982

Tron
1982
police
Tom Cruise
...
## Key-Value MemNNs

Test results on WikiQA

<table>
<thead>
<tr>
<th>Method</th>
<th>MAP</th>
<th>MRR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word Cnt</td>
<td>0.4891</td>
<td>0.4924</td>
</tr>
<tr>
<td>Wgt Word Cnt</td>
<td>0.5099</td>
<td>0.5132</td>
</tr>
<tr>
<td>2-gram CNN (Yang et al., 2015)</td>
<td>0.6520</td>
<td>0.6652</td>
</tr>
<tr>
<td>AP-CNN (Santos et al., 2016)</td>
<td>0.6886</td>
<td>0.6957</td>
</tr>
<tr>
<td>Attentive LSTM (Miao et al., 2015)</td>
<td>0.6886</td>
<td>0.7069</td>
</tr>
<tr>
<td>Attentive CNN (Yin and Schütze, 2015)</td>
<td>0.6921</td>
<td>0.7108</td>
</tr>
<tr>
<td>L.D.C. (Wang et al., 2016)</td>
<td>0.7058</td>
<td>0.7226</td>
</tr>
<tr>
<td>Memory Network</td>
<td>0.5170</td>
<td>0.5236</td>
</tr>
<tr>
<td>Key-Value Memory Network</td>
<td><strong>0.7069</strong></td>
<td><strong>0.7265</strong></td>
</tr>
</tbody>
</table>
Dynamic MemNNs

Ask Me Anything: Dynamic Memory Networks for Natural Language Processing: Kumar et. al., 2016

MemNN framework allows freedom of how to represent memories, how to represent questions, and how to get the answers given the question and the input.

Dynamic MemNNs is a recently proposed extension along these lines. It has four modules — Input Module — Question Module — Episodic Memory Module — Answer Module.

![Diagram of Dynamic MemNNs](image-url)
Dynamic MemNNs

Ask Me Anything: Dynamic Memory Networks for Natural Language Processing: Kumar et. al., 2016

**Input Module**
Generates and stores the representations of input statements (stories) — output of an RNN as the input representation — GRU

**Question Module**
Similar to the Input Module — output of an RNN as the question representation — GRU
Dynamic MemNNs

Ask Me Anything: Dynamic Memory Networks for Natural Language Processing: Kumar et. al., 2016

Figure 3.

Real example of an input list of sentences and the attention gates that are triggered by a specific question from the bAbI tasks (Weston et al., 2015a). Gate values $g_i$ are shown above the corresponding vectors. The gates change with each search over inputs. We do not draw connections for gates that are close to zero. Note that the second iteration has wrongly placed some weight in sentence 2, which makes some intuitive sense, as sentence 2 is another place John had been.

2.3. Episodic Memory Module

The episodic memory module iterates over representations outputted by the input module, while updating its internal episodic memory. In its general form, the episodic memory module is comprised of an attention mechanism as well as a recurrent network with which it updates its memory. During each iteration, the attention mechanism attends over the fact representations $c$ while taking into consideration the question representation $q$ and the previous memory $m_i$ to produce an episode $e_i$. The episode is then used, alongside the previous memories $m_i$, to update the episodic memory $m_i = \text{GRU}(e_i, m_i)$. The initial state of this GRU is initialized to the question vector itself: $m_0 = q$. For some tasks, it is beneficial for episodic memory module to take multiple passes over the input. After $T_M$ passes, the final memory $m_{T_M}$ is given to the answer module.

**Need for Multiple Episodes:**

The iterative nature of this module allows it to attend to different inputs during each pass. It also allows for a type of transitive inference, since the first pass may uncover the need to retrieve additional facts. For instance, in the example in Fig. 3, we are asked Where is the football? In the first iteration, the model ought to attend to sentence 7 (John put down the football.), as the question asks about the football. Only once the model sees that John is relevant can it reason that the second iteration should retrieve where John was. Similarly, a second pass may help for sentiment analysis as we show in the experiments section below.

**Attention Mechanism:**

In our work, we use a gating function as our attention mechanism. For each pass $i$, the mechanism takes as input a candidate fact $c_t$, a previous memory $m_i$, and the question $q$ to compute a gate: $g_i = G(c,t,m_i,q)$. The scoring function $G$ takes as input the feature set $z(c,m,q)$ and produces a scalar score. We first define a large feature vector that captures a variety of similarities between input, memory and question vectors:

$$z(c,m,q) = h_{c,m,q} || c^T W(b) || m_i^T W(b) \text{, (5)}$$

where $|$ is the element-wise product. The function $G$ is a simple two-layer feed forward neural network.
Dynamic MemNNs

Ask Me Anything: Dynamic Memory Networks for Natural Language Processing: Kumar et. al., 2016

Given a vector the answer modules maps it to the final answer
Depending on the task the answer module is either triggered once at the end of the episode or at every time step
A typical module would have an RNN whose initial hidden state is the final memory, the inputs are the question word sequence and outputs are the answer words
### Dynamic MemNNs Experiments

**Ask Me Anything: Dynamic Memory Networks for Natural Language Processing: Kumar et. al., 2016**

<table>
<thead>
<tr>
<th>Task</th>
<th>MemNN</th>
<th>DMN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: Single Supporting Fact</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>2: Two Supporting Facts</td>
<td>100</td>
<td>98.2</td>
</tr>
<tr>
<td>3: Three Supporting Facts</td>
<td>100</td>
<td>95.2</td>
</tr>
<tr>
<td>4: Two Argument Relations</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>5: Three Argument Relations</td>
<td>98</td>
<td>99.3</td>
</tr>
<tr>
<td>6: Yes/No Questions</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>7: Counting</td>
<td>85</td>
<td>96.9</td>
</tr>
<tr>
<td>8: Lists/Sets</td>
<td>91</td>
<td>96.5</td>
</tr>
<tr>
<td>9: Simple Negation</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>10: Indefinite Knowledge</td>
<td>98</td>
<td>97.5</td>
</tr>
<tr>
<td>11: Basic Coreference</td>
<td>100</td>
<td>99.9</td>
</tr>
<tr>
<td>12: Conjunction</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>13: Compound Coreference</td>
<td>100</td>
<td>99.8</td>
</tr>
<tr>
<td>14: Time Reasoning</td>
<td>99</td>
<td>100</td>
</tr>
<tr>
<td>15: Basic Deduction</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>16: Basic Induction</td>
<td>100</td>
<td>99.4</td>
</tr>
<tr>
<td>17: Positional Reasoning</td>
<td>65</td>
<td>59.6</td>
</tr>
<tr>
<td>18: Size Reasoning</td>
<td>95</td>
<td>95.3</td>
</tr>
<tr>
<td>19: Path Finding</td>
<td>36</td>
<td>34.5</td>
</tr>
<tr>
<td>20: Agent’s Motivations</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td><strong>Mean Accuracy (%)</strong></td>
<td>93.3</td>
<td>93.6</td>
</tr>
</tbody>
</table>

**bAbI Dataset**

**Question:** Where was Mary before the Bedroom?  
**Answer:** Cinema.

<table>
<thead>
<tr>
<th>Facts</th>
<th>Episode 1</th>
<th>Episode 2</th>
<th>Episode 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yesterday Julie traveled to the school.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yesterday Marie went to the cinema.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>This morning Julie traveled to the kitchen.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bill went back to the cinema yesterday.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mary went to the bedroom this morning.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Julie went back to the bedroom this afternoon.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[done reading]</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Dynamic MemNNs Experiments

Ask Me Anything: Dynamic Memory Networks for Natural Language Processing: Kumar et. al., 2016

Stanford Sentiment Treebank

Dynamic MemNNs Experiments

Ask Me Anything: Dynamic Memory Networks for Natural Language Processing: Kumar et. al., 2016

WSJ-PTB Part of Speech Tagging Task

<table>
<thead>
<tr>
<th>Model</th>
<th>Acc (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVMTool</td>
<td>97.15</td>
</tr>
<tr>
<td>Sogaard</td>
<td>97.27</td>
</tr>
<tr>
<td>Suzuki et al.</td>
<td>97.40</td>
</tr>
<tr>
<td>Spoustova et al.</td>
<td>97.44</td>
</tr>
<tr>
<td>SCNN</td>
<td>97.50</td>
</tr>
<tr>
<td>DMN</td>
<td>97.56</td>
</tr>
</tbody>
</table>

*Table 3. Test accuracies on WSJ-PTB*
MemNNs Summary

Models which augments a standard deep network with an external readable and writable memory

These memories are learnt and used effectively in solving reasoning tasks which require long term knowledge

The architecture is quite flexible in how one represents the memories and how they are used to solve the final task
MemNNs Shortcomings

While the model is quite rich one significant drawback is that it cannot write to memory intelligently.

Given a new statement it simply writes it at the next available slot. If the memory is full it will cycle.

One cannot erase memories

One cannot compress memories
Neural Turing Machines

Follows the standard architecture of MemNNs

The primary difference is in the way it writes to the memory
NTM: Read Mechanism

Neural Turing Machines: Graves, Wayne, Danihelka 2015

$w_t$: weight vector over $N$ memory locations emitted by the read head at time $t$

$$\sum_{i=1}^{N} w_t(i) = 1, \quad 0 \leq w_t(i) \leq 1, \quad \forall i$$

$$r_t \leftarrow \sum_{i=1}^{N} w_t(i) M_t(i), \quad r_t \in \mathcal{R}^d$$

the read vector

contents of the $i$-th slot of memory at time $t$
NTM: Write Mechanism

Neural Turing Machines: Graves, Wayne, Danihelka 2015

$w_t$: weight vector over N memory locations emitted by the write head at time $t$

$e_t$: erase vector

$a_t$: add vector

\[
\tilde{M}_t(i) \leftarrow M_{t-1}(i)[1 - w_t(i)e_t]
\]

\[
M_t(i) \leftarrow \tilde{M}_t(i) + w_t(i)a_t
\]
NTM: Addressing Mechanism

Neural Turing Machines: Graves, Wayne, Danihelka 2015

How are the weight vectors computed?

A combination of content based addressing and location based addressing

Content based is the usual stuff: attention based on content

Location based is different. Allows for single step jumps or random location jumps
NTM: Addressing Mechanism

Neural Turing Machines: Graves, Wayne, Danihelka 2015

Content Based

\[ w_t^c(i) \leftarrow \frac{\exp \left( \beta_t K[k_t, M_t(i)] \right)}{\sum_j \exp \left( \beta_t K[k_t, M_t(j)] \right)}. \]

Scoring function

\[ K[u, v] = \frac{u \cdot v}{\|u\| \cdot \|v\|}. \]
Step 1: compute an interpolation vector

$$w_t^g \leftarrow g_t w_t^c + (1 - g_t) w_{t-1}.$$
NTM: Addressing Mechanism

Neural Turing Machines: Graves, Wayne, Danihelka 2015

Location Based

Step 1: compute an interpolation vector

$$w_t^g \leftarrow g_t w_t^c + (1 - g_t) w_{t-1}.$$ 

Step 2: convolve using the shift vector

$$\tilde{w}_t(i) \leftarrow \sum_{j=0}^{N-1} w_t^g(j) s_t(i - j)$$
NTM: Addressing Mechanism

Neural Turing Machines: Graves, Wayne, Danihelka 2015

Location Based

Step 1: compute an interpolation vector

\[ w_t^g \leftarrow g_t w_t^c + (1 - g_t) w_{t-1}. \]

Step 2: convolve using the shift vector

\[ \tilde{w}_t(i) \leftarrow \sum_{j=0}^{N-1} w_t^g(j) s_t(i - j) \]

Step 3: sharpen the weight vector

\[ w_t(i) \leftarrow \frac{\tilde{w}_t(i)^{\gamma_t}}{\sum_j \tilde{w}_t(j)^{\gamma_t}} \]
The preceding analysis suggests that NTM, unlike LSTM, has learned some form of
iterative difference in the way the two models solve the problem.

The networks were trained to reproduce sequences of size eight random
binary vectors, where both the sequence length and the number of repetitions were chosen
randomly from one to ten. The input representing the repeat number was normalised to
a scalar value indicating the desired number of copies, which appears on a separate input
channel. To emit the end marker at the correct time the network must be both able to
execute a “for loop” containing any subroutine it has already learned.

The repeat copy task extends copy by requiring the network to output the copied sequence a
specified number of times and then emit an end-of-sequence marker. The main motivation
was to see if the NTM could learn a simple nested function. Ideally, we would like it to be
able to use relative shifts from the
content-based addressing (to jump to start of the sequence) and location-based address-
ing (to move along the sequence). Also note that the algorithm combines both
randomly from one to ten. The input representing the repeat number was normalised to
a scalar value indicating the desired number of copies, which appears on a separate input
channel. To emit the end marker at the correct time the network must be both able to
execute a “for loop” containing any subroutine it has already learned.

The repeat copy task extends copy by requiring the network to output the copied sequence a
specified number of times and then emit an end-of-sequence marker. The main motivation
was to see if the NTM could learn a simple nested function. Ideally, we would like it to be
able to use relative shifts from the

The plots show inputs and outputs to the network (top), the vectors added to memory (middle) and the corresponding
outputs from the network (bottom). Note the translation of the focal point over time, reflects the network's use of iterative
localisation (Equation 3) and global error: at the point indicated by the red arrow at the bottom, a single vector is duplicated,
though global error: at the point indicated by the red arrow at the bottom, a single vector is duplicated,
NTM: Experiments

Copy Experiment

Read the input sequence and re-generate it after finishing reading it

Figure 5: LSTM Generalisation on the Copy Task. The plots show inputs and outputs for the same sequence lengths as Figure 4. Like NTM, LSTM learns to reproduce sequences of up to length 20 almost perfectly. However, it clearly fails to generalise to longer sequences. Also note that the length of the accurate prefix decreases as the sequence length increases, suggesting that the network has trouble retaining information for long periods.

Figure 6: NTM Memory Use During the Copy Task. The plots in the left column depict the inputs to the network (top), the vectors added to memory (middle) and the corresponding write weightings (bottom) during a single test sequence for the copy task. The plots on the right show the outputs from the network (top), the vectors read from memory (middle) and the read weightings (bottom). Only a subset of memory locations are shown. Notice the sharp focus of all the weightings on a single location in memory (black is weight zero, white is weight one). Also note the translation of the focal point over time, reflects the network’s use of iterative shifts for location-based addressing, as described in Section 3.3.2. Lastly, observe that the read locations exactly match the write locations, and the read vectors match the add vectors. This suggests that the network writes each input vector in turn to a specific memory location during the input phase, then reads from the same location sequence during the output phase.
NTM: Experiments

Repeat Copy Experiment

Read the input sequence and re-generate it after finishing reading it N number of times

Figure 7: Repeat Copy Learning Curves.

Figure 7 shows that NTM learns the task much faster than LSTM, but both were able to solve it perfectly.

3 The difference between the two architectures only becomes clear when they are asked to generalise beyond the training data. In this case we were interested in generalisation along two dimensions: sequence length and number of repetitions. Figure 8 illustrates the effect of doubling first one, then the other, for both LSTM and NTM. Whereas LSTM fails both tests, NTM succeeds with longer sequences and is able to perform more than ten repetitions; however it is unable to keep count of how many repeats it has completed, and does not predict the end marker correctly. This is probably a consequence of representing the number of repetitions numerically, which does not easily generalise beyond a fixed range.

4.3 Associative Recall

The previous tasks show that the NTM can apply algorithms to relatively simple, linear data structures. The next order of complexity in organising data arises from "indirection"—that is, when one data item points to another. We test the NTM's capability for learning an instance of this more interesting class by constructing a list of items so that querying with one of the items demands that the network return the subsequent item. More specifically, we define an item as a sequence of binary vectors that is bounded on the left and right by delimiter symbols. After several items have been propagated to the network, we query by showing a random item, and we ask the network to produce the next item. In our experiments, each item consisted of three six-bit binary vectors (giving a total of 18 bits per item). During training, we used a minimum of 2 items and a maximum of 6 items in a single episode.

Figure 10 shows that NTM learns this task significantly faster than LSTM, terminating at near zero cost within approximately 30,000 episodes, whereas LSTM does not reach zero cost after a million episodes. Additionally, NTM with a feedforward controller learns faster than NTM with an LSTM controller. These two results suggest that NTM's external memory is a more effective way of maintaining the data structure than LSTM's internal state. NTM also generalises much better to longer sequences than LSTM, as can be seen in Figure 11. NTM with a feedforward controller is nearly perfect for sequences of up to 12 items (twice the maximum length used in training), and still has an average cost below 1 bit per sequence for sequences of 15 items.

In Figure 12, we show the operation of the NTM memory, controlled by an LSTM with one head, on a single test episode. In "Inputs," we see that the input denotes item delimiters as single bits in row 7. After the sequence of items has been propagated, a
NTM: Experiments

Repeat Copy Experiment

Read the input sequence and re-generate it after finishing reading it N number of times

Figure 9: NTM Memory Use During the Repeat Copy Task.

As with the copy task, the network first writes the input vectors to memory using iterative shifts. It then reads through the sequence to replicate the input as many times as necessary (six in this case). The white dot at the bottom of the read weightings seems to correspond to an intermediate location used to redirect the head to the start of the sequence (the NTM equivalent of a goto statement).

Figure 10: Associative Recall Learning Curves for NTM and LSTM.
NTM: Experiments

Sorting Experiment

Sort a collection of vectors according to their given priority

**Figure 16:** Example Input and Target Sequence for the Priority Sort Task. The input sequence contains random binary vectors and random scalar priorities. The target sequence is a subset of the input vectors sorted by the priorities.

**Figure 17:** NTM Memory Use During the Priority Sort Task. Left: Write locations returned by fitting a linear function of the priorities to the observed write locations. Middle: Observed write locations. Right: Read locations.

4.6 Experimental Details

For all experiments, the RMSProp algorithm was used for training in the form described in (Graves, 2013) with momentum of 0.9. Tables 1 to 3 give details about the network configurations and learning rates used in the experiments. All LSTM networks had three stacked hidden layers. Note that the number of LSTM parameters grows quadratically with
NTM: Experiments

Sorting Experiment

Sort a collection of vectors according to their given priority

![Sorting Experiment Graph]

Figure 18: Priority Sort Learning Curves.

<table>
<thead>
<tr>
<th>Task</th>
<th>Heads</th>
<th>Controller Size</th>
<th>Memory Size</th>
<th>Learning Rate</th>
<th>#Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy</td>
<td>1</td>
<td>100</td>
<td>128</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Repeat Copy</td>
<td>1</td>
<td>100</td>
<td>128</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Associative</td>
<td>4</td>
<td>256</td>
<td>128</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N-Grams</td>
<td>1</td>
<td>100</td>
<td>128</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Priority Sort</td>
<td>8</td>
<td>512</td>
<td>128</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: NTM with Feedforward Controller Experimental Settings

The number of hidden units (due to the recurrent connections in the hidden layers). This contrasts with NTM, where the number of parameters does not increase with the number of memory locations. During the training backward pass, all gradient components are clipped elementwise to the range (-10, 10).

5 Conclusion

We have introduced the Neural Turing Machine, a neural network architecture that takes inspiration from both models of biological working memory and the design of digital computers. Like conventional neural networks, the architecture is differentiable end-to-end and can be trained with gradient descent. Our experiments demonstrate that it is capable of learning simple algorithms from example data and of using these algorithms to generalise well outside its training regime.
NTM: Summary

Another way to augment external memory with a standard deep network

The writer is general enough that it can erase the previous contents of the memory and write new content

Addressing mechanism is more sophisticated than MemNNs

As yet, shown only to work on toy problems which require only small amounts of memory.*
NTM: Summary

Another way to augment external memory with a standard deep network

The writer is general enough that it can erase the previous contents of the memory and write new content

Addressing mechanism is more sophisticated than MemNNs

As yet, shown only to work on toy problems which require only small amounts of memory.*

Very recently there has been some new developments in this area

Dynamic Neural Turing Machine with Soft and Hard Addressing Schemes: Gulcehre et. al., 2016

One-Shot Learning with Memory Augmented Neural Networks: Santoro et. al., 2016
Stack Augmented RNNs

So far we’ve dealt with memories which are like tapes

For MemNNs the tapes are write-once read-multiple

For NTM tapes are write-multiple read multiple

Natural to think of other forms of memory data structures - stacks, lists, queues, de-queues and more
Stack Augmented RNNs

A number of people have worked on such architectures

Learning Context-Free Grammars: Capabilities and Limitations of a Recurrent Neural Network with External Stack Memory: Das et. al., 1992

A Connectionist Symbol Manipulator that Discovers the Structure of Context Free Languages: Mozer and Das, 1993

The Induction of Dynamical Recognizers: Pollack, 1991

Discrete Recurrent Neural Networks for Grammatical Inference: Zeng et. al., 1994

Inferring Algorithmic Patterns with Stack-Augmented Recurrent Nets: Joulin and Mikolov, 2015

Learning to Transduce with Unbounded Memory: Grefenstette et. al., 2015
Stack Augmented RNNs

Inferring Algorithmic Patterns with Stack-Augmented Recurrent Nets, Joulin and Mikolov, 2015

Standard Recurrent Net

\[ h_t = \sigma(Ux_t + Rh_{t-1}) \]

\[ y_t = g(Vh_t) \]
Stack Augmented RNNs

Inferring Algorithmic Patterns with Stack-Augmented Recurrent Nets, Joulin and Mikolov, 2015

Stack Augmented Recurrent Net

\[ a_t = f(Ah_t) \]

\[ s_t[0] = a_t[\text{PUSH}]\sigma(Dh_t) + a_t[\text{POP}]s_{t-1}[1], \]

\[ s_t[i] = a_t[\text{PUSH}]s_{t-1}[i - 1] + a_t[\text{POP}]s_{t-1}[i + 1]. \]

\[ h_t = \sigma(Ux_t + Rh_{t-1} + Ps_{t-1}^k), \]
Stack Augmented RNNs

Inferring Algorithmic Patterns with Stack-Augmented Recurrent Nets, Joulin and Mikolov, 2015

<table>
<thead>
<tr>
<th>method</th>
<th>$a^nb^n$</th>
<th>$a^nb^n c^n$</th>
<th>$a^nb^n c^nd^n$</th>
<th>$a^nb^{2n}$</th>
<th>$a^nb^m c^{n+m}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>RNN</td>
<td>25%</td>
<td>23.3%</td>
<td>13.3%</td>
<td>23.3%</td>
<td>33.3%</td>
</tr>
<tr>
<td>LSTM</td>
<td>100%</td>
<td>100%</td>
<td>68.3%</td>
<td>75%</td>
<td>100%</td>
</tr>
<tr>
<td>List RNN 40+5</td>
<td>100%</td>
<td>33.3%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Stack RNN 40+10</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>43.3%</td>
</tr>
<tr>
<td>Stack RNN 40+10 + rounding</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
</tbody>
</table>

Table 2: Comparison with RNN and LSTM on sequences generated by counting algorithms. The sequences seen during training are such that $n < 20$ (and $n + m < 20$), and we test on sequences up to $n = 60$. We report the percent of $n$ for which the model was able to correctly predict the sequences. Performance above 33.3% means it is able to generalize to never seen sequence lengths.
Wrapping Up

We discussed the importance of having a persistent memory in models for a number of problems.

Memory Networks — Neural Turing Machines — Stack Augmenting RNNs

Attention Mechanism (soft/hard) seems to be one fundamental way of implementing things.

Quite a bit lacking still.
Wrapping Up

How to decide what to write and what not to write

How to decide which type of memory to use and when?

How to represent knowledge stored in memory

How to incorporate forgetting/compression of information

How to build hierarchical memories: multi scale attention?

How to build hierarchical reasoning: composition of functions?
Thank You!