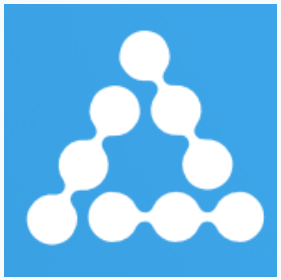


Memory Networks

Jason Weston, Sumit Chopra and Antoine Bordes

Facebook AI Research

Memory Networks

- Class of models that combine large memory with learning component that can read and write to it.
- Most ML has limited memory which is more-or-less all that's needed for "low level" tasks e.g. object detection.
- **Our motivation:** long-term memory is required to read a story (or watch a movie) and then e.g. answer questions about it.
- *We study this by building a simple simulation to generate ``stories''. We also try on some real QA data.*

MCTest comprehension data (Richardson et al.)

James the Turtle was always getting in trouble. Sometimes he'd reach into the freezer and empty out all the food. Other times he'd sled on the deck and get a splinter. His aunt Jane tried as hard as she could to keep him out of trouble, but he was sneaky and got into lots of trouble behind her back.

One day, James thought he would go into town and see what kind of trouble he could get into. He went to the grocery store and pulled all the pudding off the shelves and ate two jars. Then he walked to the fast food restaurant and ordered 15 bags of fries. He didn't pay, and instead headed home.

His aunt was waiting for him in his room. She told James that she loved him, but he would have to start acting like a well-behaved turtle.

After about a month, and after getting into lots of trouble, James finally made up his mind to be a better turtle.

Q: What did James pull off of the shelves in the grocery store?

A) pudding B) fries C) food D) splinters

Q: Where did James go after he went to the grocery store?

A) his deck B) his freezer C) a fast food restaurant D) his room

...

MCTest comprehension data (Richardson et al.)

James the Turtle was always getting in trouble. Sometimes he'd reach into the freezer and empty out all the food. Other times he'd sled on the deck and get a splinter. His aunt Jane tried as hard as she could to keep him out of trouble, but he was sneaky and got into lots of trouble behind her back.

Problems: **…** *it's hard for this data to lead us to design good ML models …*

1) Not enough data to train on (660 stories total). What ends up happening is that shallow methods perform similarly to fancy ones.

2) If we get something wrong we don't really understand why: every question potentially involves a different kind of reasoning, our model has to do a lot of different things.

Our solution: *focus on simpler (toy) subtasks where we can generate data to check what the models we design can and cannot do.*

A) pudding B) fries C) food D) splinters

Q: Where did James go after he went to the grocery store?

A) his deck B) his freezer C) a fast food restaurant D) his room

…

Simulation: Basic Commands

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

Commands only for "gods" (superusers):

- **create <object>**
- **set <object1> <relation> <object2>** *- change the graph that represents the world*

Example QA in bAbI world.

Dataset in simulation command format.	**Dataset after adding a simple grammar.**
antoine go kitchen	Antoine went to the kitchen.
antoine get milk	Antoine picked up the milk.
antoine go office	Antoine travelled to the office.
antoine drop milk	Antoine left the milk there.
antoine go bathroom	Antoine went to the bathroom.
where is milk ? (A: office)	Where is the milk now? (A: office)
where is antoine ? (A: bathroom)	Where is Antoine? (A: bathroom)

Task (1) Factoid QA with Single Supporting Fact ("where is actor")

Our first task consists of questions where a single supporting fact, previously given, provides the answer.

We test the simplest case of this, by asking for the location of a person.

A small sample of the task is thus:

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

It can be considered the simplest case of some real world QA datasets such as in Fader et al., '13.

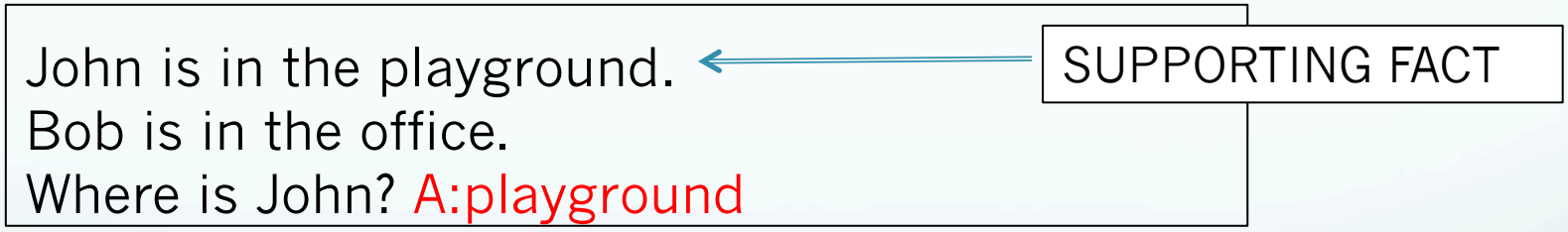
We can control the difficulty by the distance in the past of the supporting fact, and how many other irrelevant facts there are.

Task (1) Factoid QA with Single Supporting Fact ("where is actor")

Our first task consists of questions where a single supporting fact, previously given, provides the answer.

We test simplest case of this, by asking for the location of a person.

A small sample of the task is thus:



We use supporting facts for supervision at training time, but are not known at test time (we call this "strong supervision")

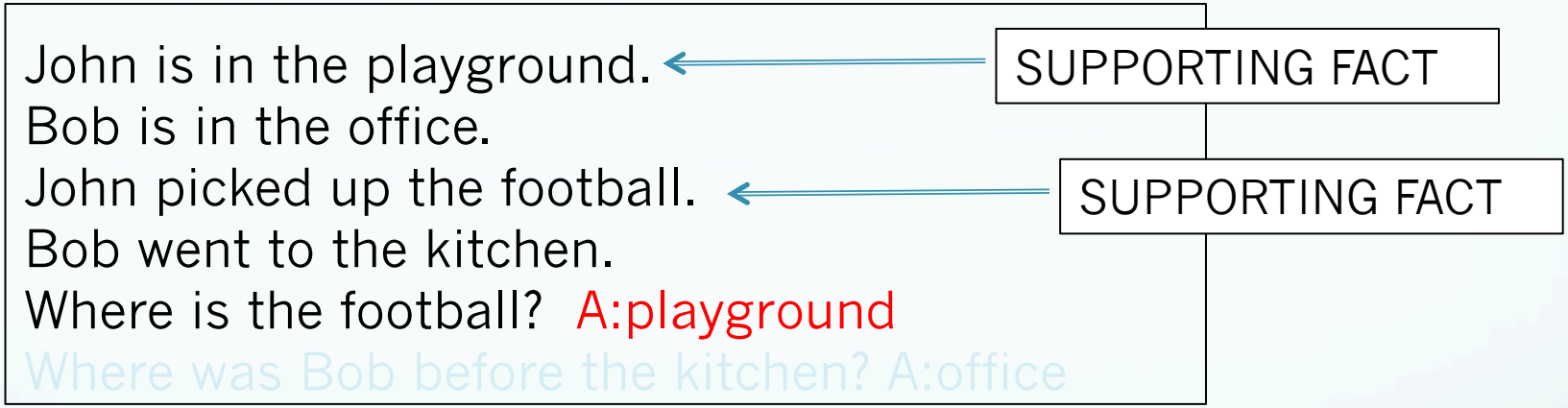
(2) Factoid QA with Two Supporting Facts ("where is actor+object")

A harder task is to answer questions where two supporting statements have to be chained to answer the question:

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
Where was Bob before the kitchen? A:office

(2) Factoid QA with Two Supporting Facts ("where is actor+object")

A harder task is to answer questions where two supporting statements have to be chained to answer the question:



To answer the first question *Where is the football?* both *John picked up the football* and *John is in the playground* are supporting facts.

.

Large QA: Reverb Dataset in (Fader et al., 13)

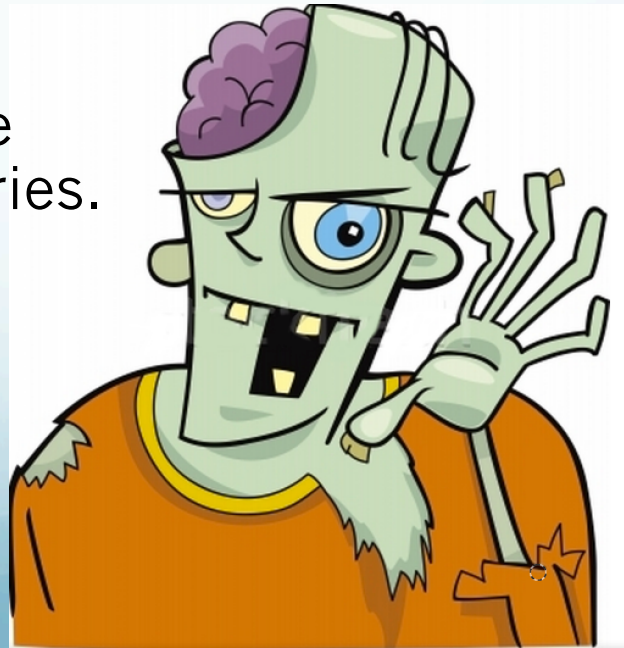
- 14M statements, stored as (subject, relation, object) triples. Triples are REVERB extractions mined from ClueWeb09.
- Statements cover diverse topics:
  - (milne, authored, winnie-the-pooh)
  - (sheep, be-afraid-of, wolf), *etc...*
- Weakly labeled QA pairs and 35M paraphrased questions from WikiAnswers:
  - ``Who wrote the Winnie the Pooh books?
  - ``Who is poohs creator?''

We've also worked on the WebQuestions dataset *(not featured in this talk)*

Memory Networks

MemNNs have four component networks (which may or may not have shared parameters):

- **I:** (input feature map) this converts incoming data to the internal feature representation.
- **G:** (generalization) this updates memories given new input.
- **O:** this produces new output (in feature representation space) given the memories.
- **R:** (response) converts output O into a response seen by the outside world.



Related Memory Models

- RNNSearch (Bahdanau et al.) for Machine Translation
  - Can be seen as a MemNN where memory goes back only one sentence (writes embedding for each word).
  - At prediction time, reads memory and performs a soft max to find best alignment (most useful words).
- Generating Sequences With RNNs (Graves, '13)
  - Also does alignment with previous sentence to generate handwriting (so RNN knows what letter it's currently on).
- Neural Turing Machines (Graves et al., 14)
  [on arxiv just 5 days after MemNNs!]
  - Has read and write operations over memory to perform tasks (e.g. copy, sort, associative recall).
  - 128 memory slots in experiments; content addressing computes a score for each slot ➔ slow for large memory?
- Earlier work by (Das '92), (Schmidhuber et al., 93), DISCERN (Miikkulainen, '90) and others...

Our First **MemNN** Implemention

- I (input): no conversion, keep original text x.
- G (generalization): stores I(x) in next available slot m_N.
- O (output): Loops over all memories k=1 or 2 times:
  - 1st loop max: finds best match m_i with x.
  - 2nd loop max: finds best match m_J with (x, m_i).
  - The output o is represented with (x, m_i, m_J).
- R (response): ranks all words in the dictionary given o and returns best single word. *(OR: use a full RNN here)*

Some Extensions

Some options and extensions:

- **If the input is at the character or word level** one could group inputs (i.e. learn to segment the input into chunks) and put each chunk in a memory slot.
- **Use an RNN for module R** to give true responses.
- **If the memory is huge** (e.g. Wikipedia) we need to organize the memories. Solution: S can hash the memories to store in buckets (topics). Then, G and O don't operate on *all* memories.
- **If the memory is full**, there could be a way of removing one it thinks is most useless; *i.e.* it ``forgets'' somehow. That would require a scoring function of the utility of each memory..

Explorations of the first 3 points can be found in the paper.

Matching function

- For a given Q, we want a good match to the relevant memory slot(s) containing the answer, e.g.:

Match(Where is the football ?, John picked up the football)

- We use a $q^T U^T U d$ embedding model with word embedding features:
  - *LHS features:* Q:Where Q:is Q:the Q:football Q:?
  - *RHS features:* D:John D:picked D:up D:the D:football QDMatch:the QDMatch:football

  (QDMatch:football is a feature to say there's a Q&A word match, which can help.)

The parameters U are trained with a margin ranking loss: supporting facts should score higher than non-supporting facts.

Matching function: 2nd hop

- On the 2nd hop we match question & 1st hop to new fact:

Match([Where is the football ?, John picked up the football],
J John is in the playground)

- We use the same $q^T U^T U d$ embedding model:
  - *LHS features:* Q:Where Q:is Q:the Q:football Q:? Q2: John Q2:picked Q2:up Q2:the Q2:football
  - *RHS features:* D:John D:is D:in D:the D:playground QDMatch:the QDMatch:is .. Q2DMatch:John

- We also need time information for bAbI simulation. We tried adding absolute time differences (between two memories) as a feature: tricky to get to work.

Comparing triples

- Seems to work better if we compare triples:
- Match(Q,D,D') returns < 0 if D is better than D'

  returns > 0 if D' is better than D

We can loop through memories, keep best m_i at each step.

Now the features include *relative* time features:

L.H.S: same as before

R.H.S: features(D) DbeforeQ:*0-or-1*

-features(D') D'beforeQ:*0-or-1* DbeforeD':*0-or-1*

Results: QA on Reverb data from (Fader et al.)

- 14M statements stored in the memNN memory.
- *k=1 loops* MemNN, *128-dim* embedding.
- R response simply outputs top scoring statement.
- Time features are not necessary, hence not used.
- We also tried adding bag of words (BoW) features.

Method	F1
(Fader et al., 2013)	0.54
(Bordes et al., 2014)	0.73
MemNN	0.72
MemNN (with BoW features)	0.82

Fast QA on Reverb data

Scoring all 14M candidates in the memory is slow.

We consider speedups using hashing in S and O as mentioned earlier:

- Hashing via words (*essentially: inverted index*)
- Hashing via k-means in embedding space (k=1000)

Method	Embedding	Embed+BoW	candidates
MemNN (no hashing)	0.72	0.82	14M
MemNN (word hash)	0.63	0.68	13k (1000x)
MemNN (clust hash)	0.71	0.80	177k (80x)

bAbI Experiment 1

- 10k sentences. (Actor: only ask questions about actors.)
- Difficulty: how many sentences in the past when entity mentioned.
- Fully supervised (supporting sentences are labeled).
- Compare RNN (no supervision)
  and MemNN hops *k = 1 or 2*, & with/without time features.

	Difficulty 1		Difficulty 5	
Method	actor	actor+object	actor	actor+object
RNN	0%	42%	71%	83%
MemNN $k = 1$	10%	81%	54%	79%
MemNN $k = 1$ (+time)	0%	27%	0%	27%
MemNN $k = 2$ (+time)	0%	0.05%	0%	0.6%

Difficulty 5 -- Max mem. sz. required: 65 Average mem. sz. required: 9

bAbI Experiment 1

- Example test story + predictions:

Antoine went to the kitchen. Antoine got the milk. Antoine travelled to the office. Antoine dropped the milk. Sumit picked up the football. Antoine went to the bathroom. Sumit moved to the kitchen.

- *where is the milk now?* A: office
- *where is the football?* A: kitchen
- *where is Antoine ?* A: bathroom
- *where is Sumit ?* A: kitchen
- *where was Antoine before the bathroom*? A: office

Unsegmented setup; R module is an RNN

Joe went to the garden then Fred picked up the milk; Joe moved to the bathroom and Fred dropped the milk, and then Dan moved to the living room.

- *Where is Dan?* A: living room I believe
- *Where is Joe?* A: the bathroom

Fred moved to the bedroom and Joe went to the kitchen then Joe took the milk there and Dan journeyed to the bedroom; Joe discarded the milk.

- *Where is the milk now?* A: the milk is in the kitchen
- *Where is Dan now?* A: I think he is in the bedroom

Joe took the milk there, after that Mike travelled to the office, then Joe went to the living room, next Dan went back to the kitchen and Joe travelled to the office.

- *Where is Joe now?* A: I think Joe is in the office

Dealing with new words

Sometimes words come along you've never seen before, e.g. a new name:

- *`Here,' said Elrond, turning to Gandalf, `is* **Boromir**, *a man from the South. He arrived in the grey morning, and seeks for counsel"*

Our approach: incorporate missing word prediction in **same model.**

Boromir is represented as **unk**nown with left and right context words, features shown in bold*:*

- Then Boromir goes to the office ➔
  D:UNK-L:Then D:UNK-R:goes D:UNK-R:to D:goes D:to D:the D:office

We can learn this with the original training data using a kind of "dropout": *sometimes we pretend we don't have a word's embedding, and represent it as above instead***.**

A MemNN that has never seen Frodo, Bilbo or the ring before..

The "story" told to the model *after* **training:**

"Bilbo travelled to the cave. Gollum dropped the ring there. Bilbo took the ring. Bilbo went back to the Shire. Bilbo left the ring there. Frodo got the ring. Frodo journeyed to Mount-Doom. Frodo dropped the ring there. Sauron died. Frodo went back to the Shire. Bilbo travelled to the Grey-havens. The End."

MemNN's answers to some questions:

- *Where is the ring?* A: Mount-Doom
- *Where is Bilbo now?* A: Grey-havens
- *Where is Frodo now?* A: Shire

A MemNN multitasked on bAbI data and Reverb QA data

The "story" told to the model *after* **training:**

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

MemNN's answers to some questions:

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

What's next?

- Make it harder/add more stuff, e.g. "he went..", "Frodo and Sam", **etc.!!!**
- MemNNs that reason with *more than* 2 supporting memories.
- Weakly supervised?
- Ask questions? Say statements? Perform actions?
- MCTest reading comprehension data (Richardson et al.)
- *Do MemNN ideas extend to other ML tasks and model variants...? [A: probably, yes!].*

What's next?

- Make it harder/add more stuff, e.g. "he went..", "Frodo and Sam", **etc.!!!**
- MemNNs th
memories.

J. Weston, A. Bordes, S. Chopra, T. Mikolov. Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks. arXiv:1502.05698.

- Weakly supervised?
- Ask questions? Say statements? Perform actions?
- MCTest comprehension data (Richardson et al.)
- *Do MemNN ideas extend to other ML tasks and model variants...? [A: probably, yes!].*

What else is next?

- ~~Make it harder/add more stuff, e.g. "he went..", "Frodo and Sam", **etc.!!!**~~
- ~~MemNNs that reason with *more than* 2 supporting memories.~~
- Weakly supervised?
- Ask que
- MCTest comprehension data (Richardson et al.)
- *Do MemNN ideas extend to other ML tasks and model variants...? [A: probably, yes!].*

S. Sukhbaatar, A. Szlam, J. Weston, R. Fergus.
Weakly Supervised Memory Networks.
arXiv:1503.08895.

What's next?

- Make it harder/add more stuff, e.g. "he went..", "Frodo and Sam", **etc.!!!**
- MemNNs that reason with *more than* 2 supporting memories.
- Weakly supervised?
- Ask questions? Say statements? Perform actions?
- MCTest reading comprehension data (Richardson et al.)
- *Do MemNN ideas extend to other ML tasks and model variants...? [A: probably, yes!].*

Thanks!