NEW YORK UNIVERSITY

Facebook AI Research

End-To-End Memory Networks

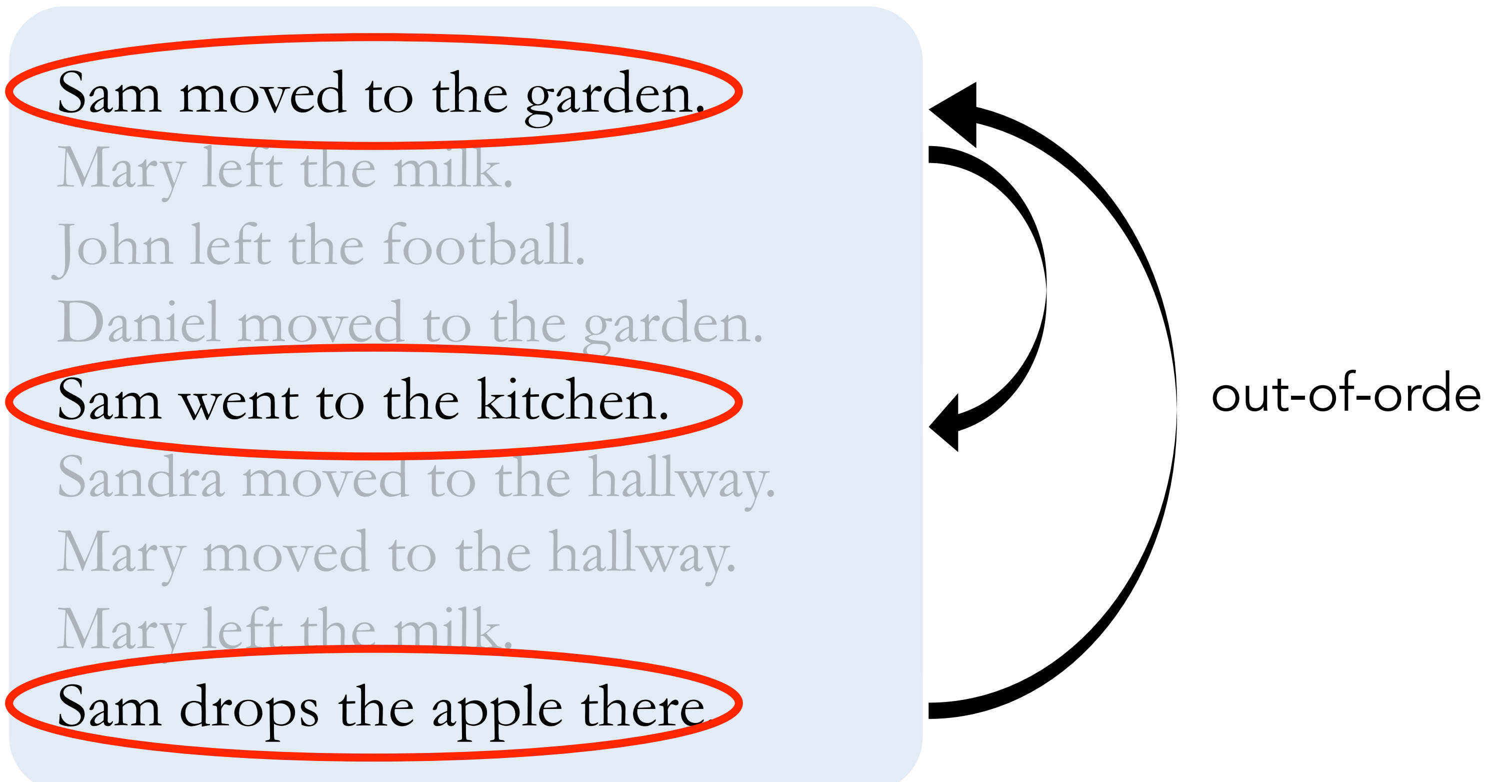
Sainbayar Sukhbaatar[1], Arthur Szlam[2], Jason Weston[2] and Rob Fergus[2]

[1]New York University [2]Facebook AI Research

Motivation

- Good models exist for some data structures
  - RNN for temporal structure
  - ConvNet for spatial structure
- But we still struggle with some type of dependencies
  - out-of-order access
  - long-term dependency
  - unordered set

Ex) Question & Answering on story



Q: Where was the apple after the garden?

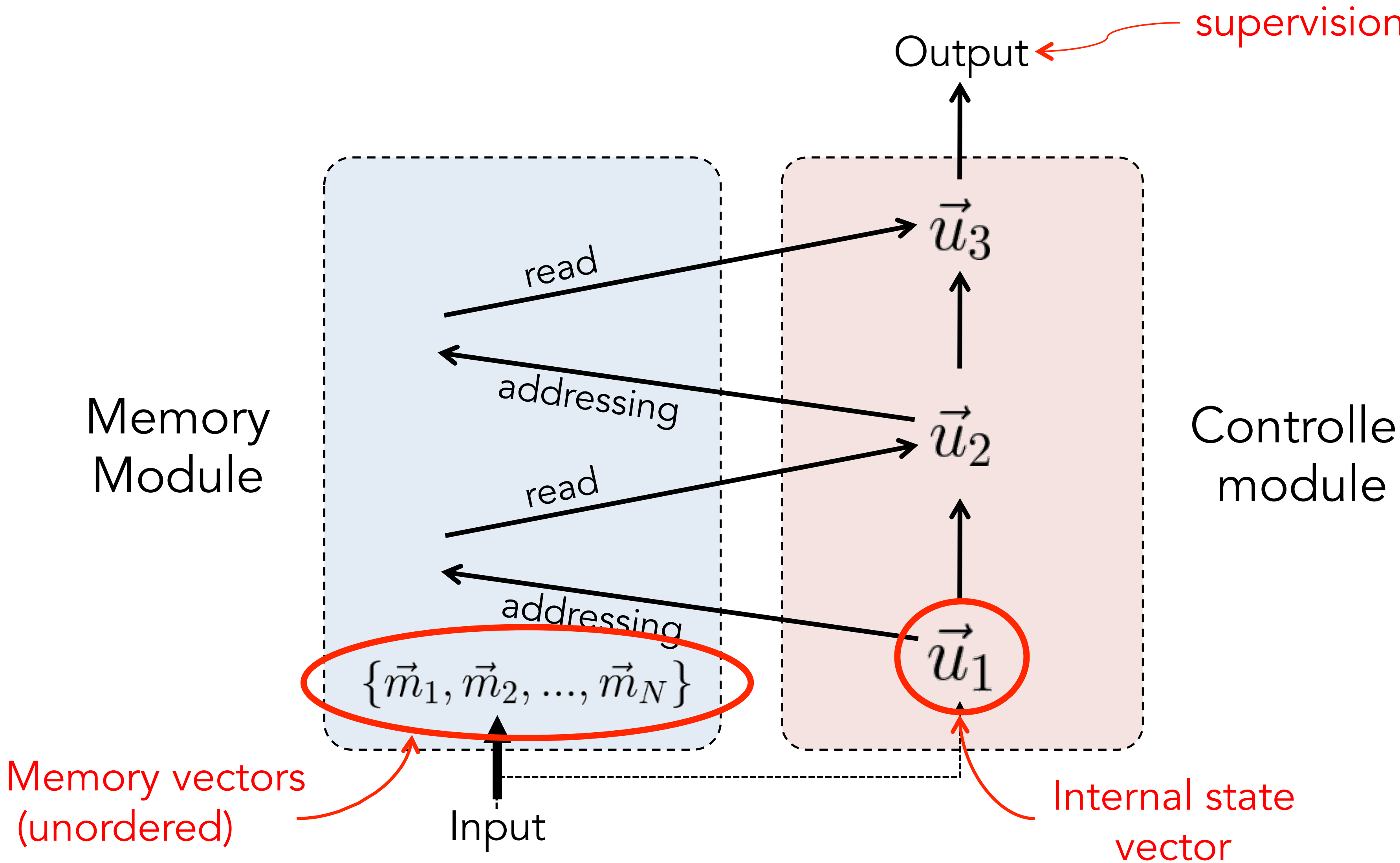
Overview

- We propose a neural network model with external memory
  - Reads from memory with **soft attention**
  - Performs **multiple lookups** (hops) on memory
  - End-to-end training with **backpropagation**

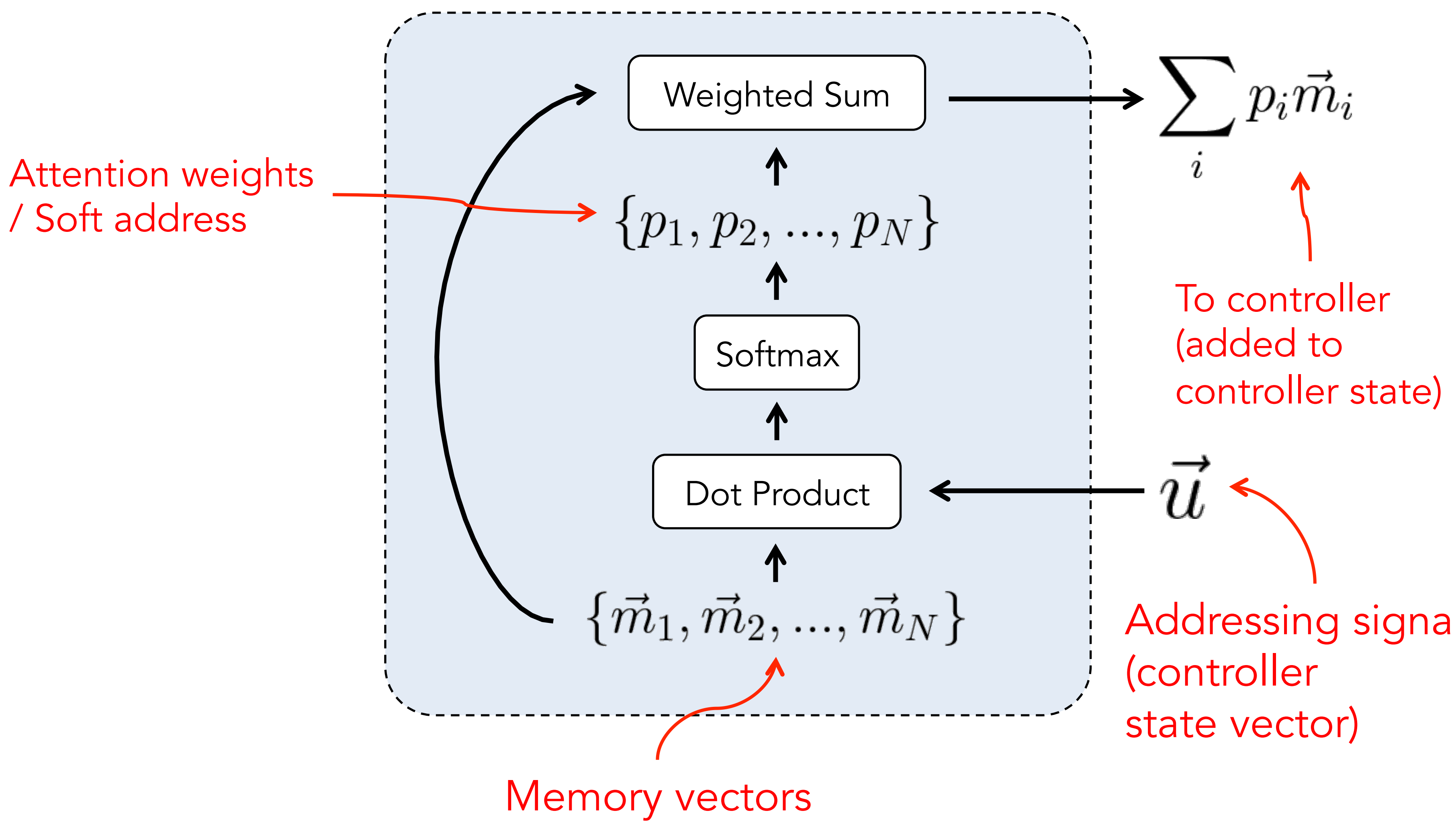
- **End-to-end Memory Network (MemN2N)**

- It is based on "Memory Networks" by [Weston, Chopra & Bordes ICLR 2015]
  - Hard attention
  - requires explicit supervision of attention during training
  - Only feasible for simple tasks
  - Severely limits application of the model
- MemN2N is **soft** attention version
- Only need supervision on the final output

MemN2N architecture



Memory Module



Memory Vectors

E.g.) constructing memory vectors with Bag-of-Words (BoW)

1. Embed each word
2. Sum embedding vectors

$$\text{“Sam drops apple”} \rightarrow \underbrace{\vec{v}_{\text{Sam}} + \vec{v}_{\text{drops}} + \vec{v}_{\text{apple}}}_{\text{Embedding Vectors}} = \vec{m}_i$$

Memory Vector

E.g.) **temporal structure:** special words for time and include them in BoW

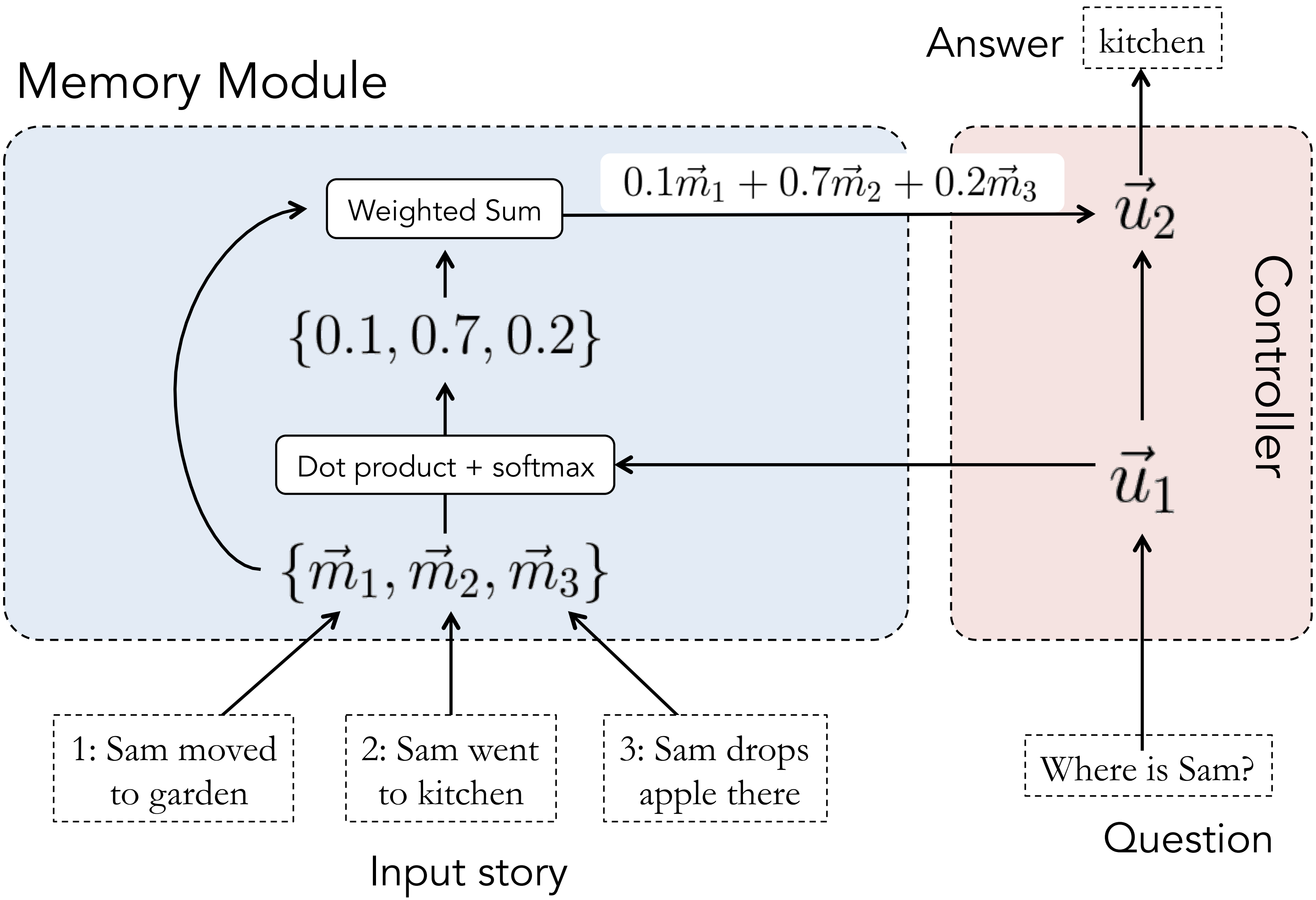
1: “Sam moved to garden”

2: “Sam went to kitchen”

3: “Sam drops apple” $\rightarrow v_{\text{Sam}} + v_{\text{drops}} + v_{\text{apple}} + v_3 = m_3$

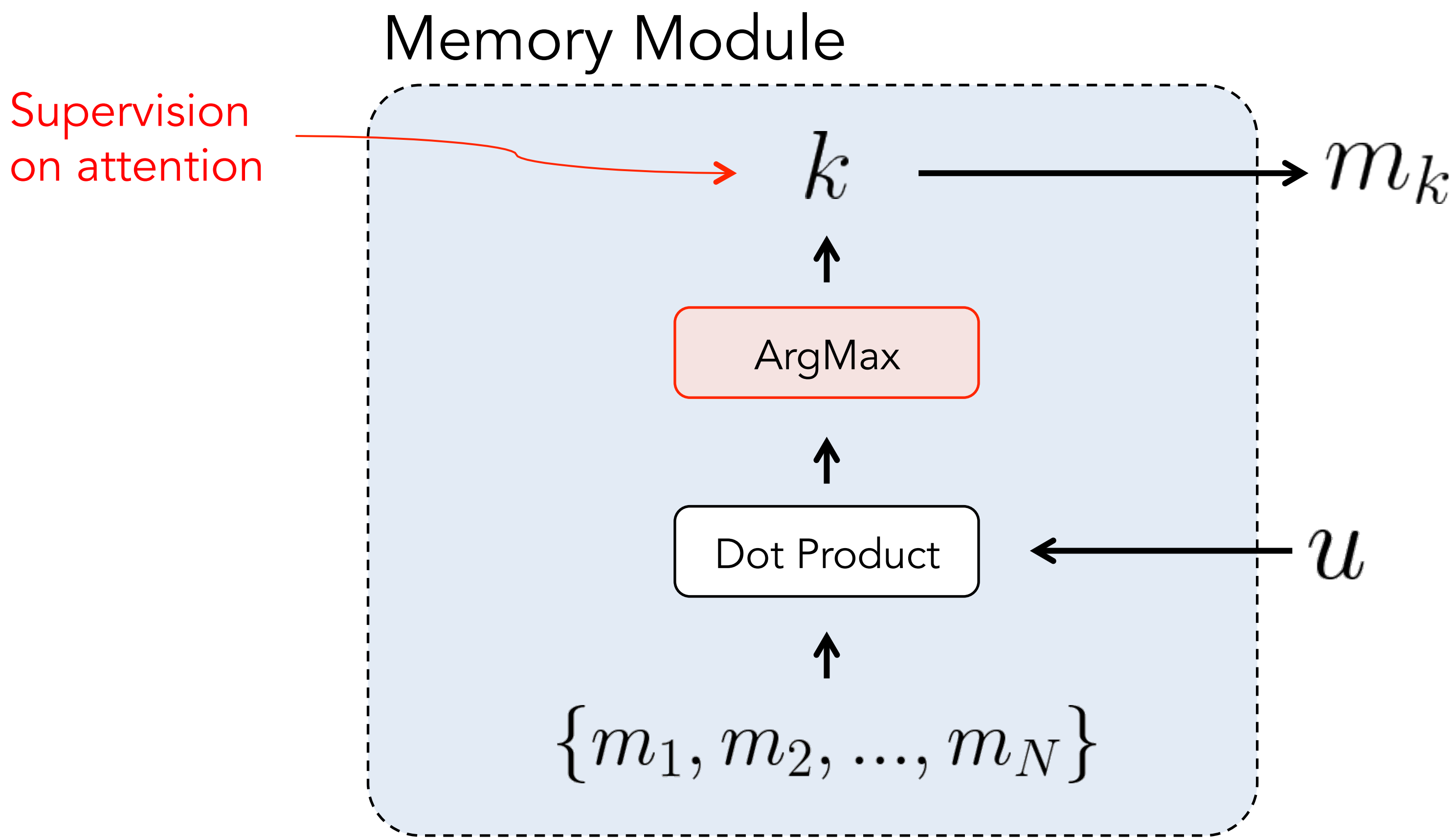
Time embedding

Question & Answering



Related Work (I)

Hard attention Memory Network [Weston et al. ICLR 2015]



Related Work (II)

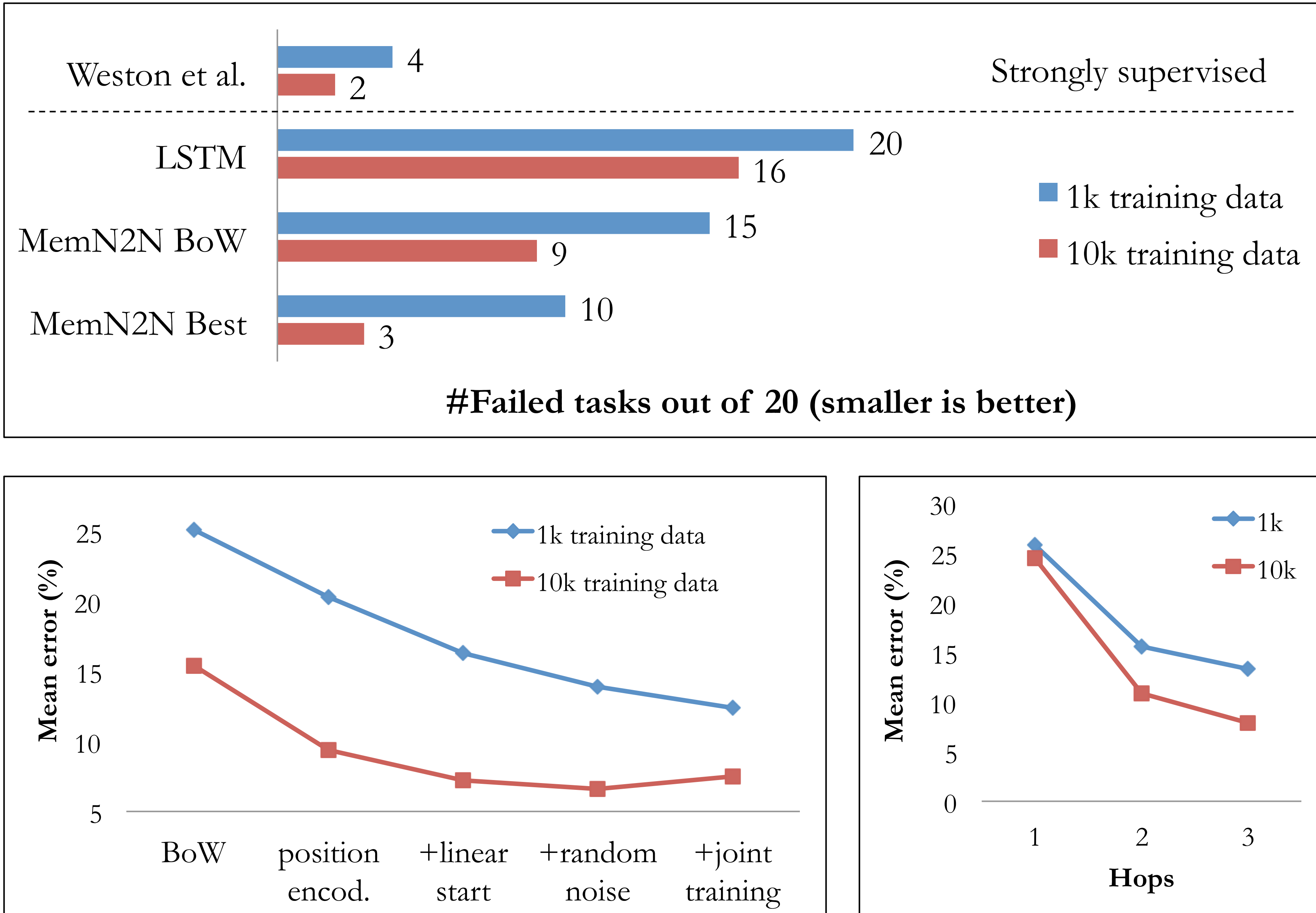
- RNNsearch [Bahdanau et al. 2015]
  - Encoder-decoder RNN with attention
  - Our model can be considered as an attention model with multiple hops
- Recent works on external memory
  - Stack memory for RNNs [Joulin & Mikolov. 2015]
  - Neural Turing Machine [Graves et al. 2014]
- Early works on neural network and memory
  - [Steinbuch & Piske. 1963]; [Taylor. 1959]
  - [Das et al. 1992]; [Mozer et al. 1993]
- Concurrent works
  - Dynamic Memory Networks [Kumar et al. 2015]
  - Attentive reader [Hermann et al. 2015]
  - Stack, Queue [Grefenstette et al. 2015]

Experiment on bAbI Q&A data

- Data: 20 bAbI tasks [Weston et al. arXiv: 1502.05698, 2015]
- Answer questions after reading short story
- Small vocabulary, simple language
- Different tasks require different reasoning
- Training data size 1K or 10K for each task

Sam walks into the kitchen. Sam picks up an apple. Sam walks into the bedroom. Sam drops the apple. Q: Where is the apple? A. Bedroom	Brian is a lion. Julius is a lion. Julius is white. Bernhard is green. Q: What color is Brian? A. White

Performance on bAbI test set



Examples of Attention Weights

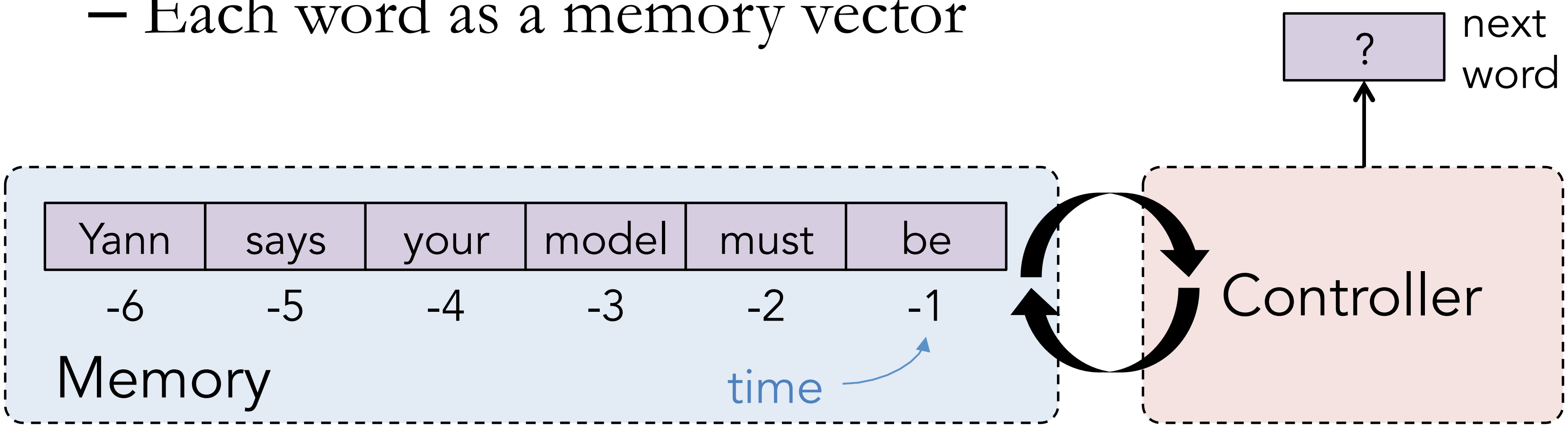
- 2 test cases:

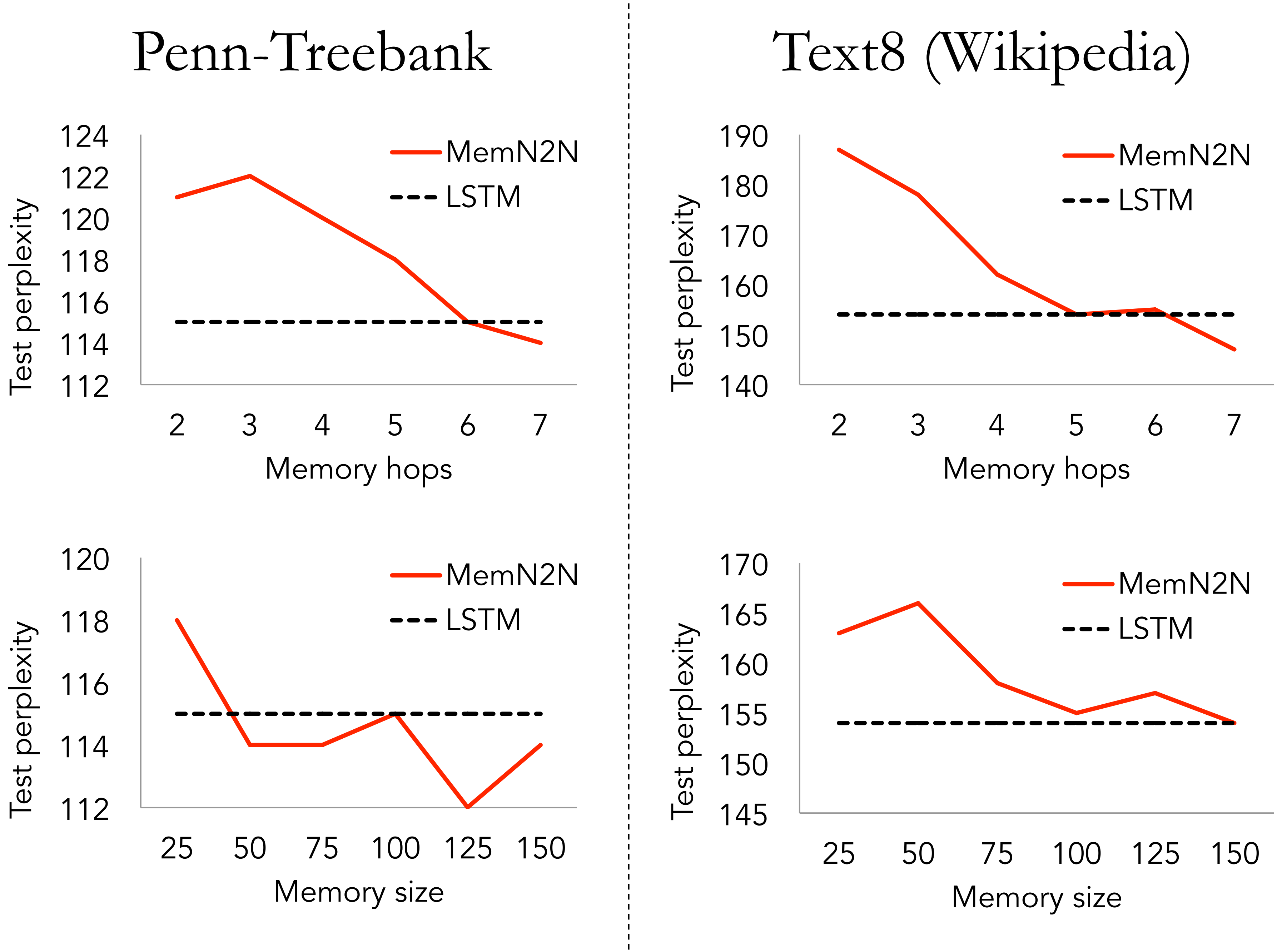
Story (2: 2 supporting facts)	Hop 1	Hop 2	Hop 3
John dropped the milk.	0.06	0.00	0.00
John took the milk there.	0.88	1.00	0.00
Sandra went back to the bathroom.	0.00	0.00	0.00
John moved to the hallway.	0.00	0.00	1.00
Mary went back to the bedroom.	0.00	0.00	0.00
Where is the milk? Answer: hallway Prediction: hallway			

Story (16: basic induction)	Hop 1	Hop 2	Hop 3
Brian is a frog.	0.00	0.98	0.00
Lily is gray.	0.07	0.00	0.00
Brian is yellow.	0.07	0.00	1.00
Julius is green.	0.06	0.00	0.00
Greg is a frog.	0.76	0.02	0.00
What color is Greg? Answer: yellow Prediction: yellow			

Experiment on Language modeling

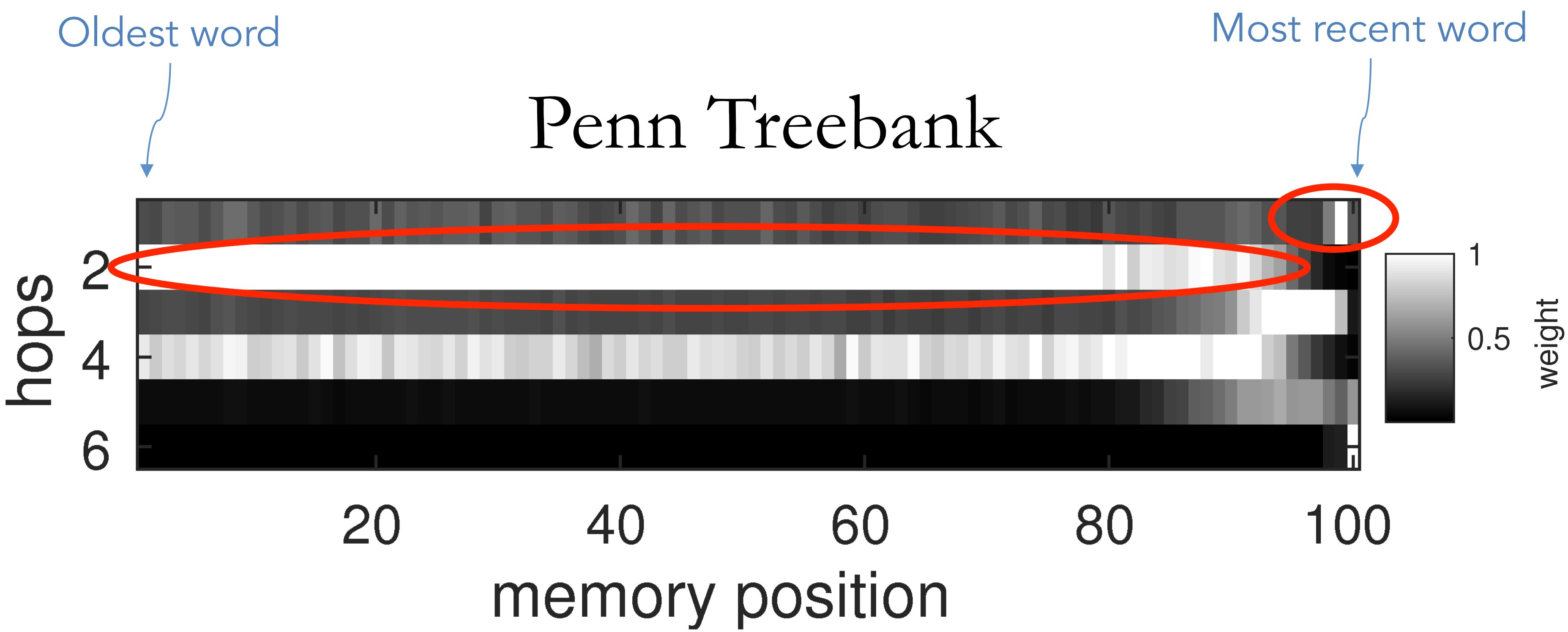
- Data
  - Penn Treebank: 1M words 10K vocab
  - Text8 (Wikipedia): 16M words 40K vocab
- Model
  - Controller module: linear + non-linearity
  - Each word as a memory vector





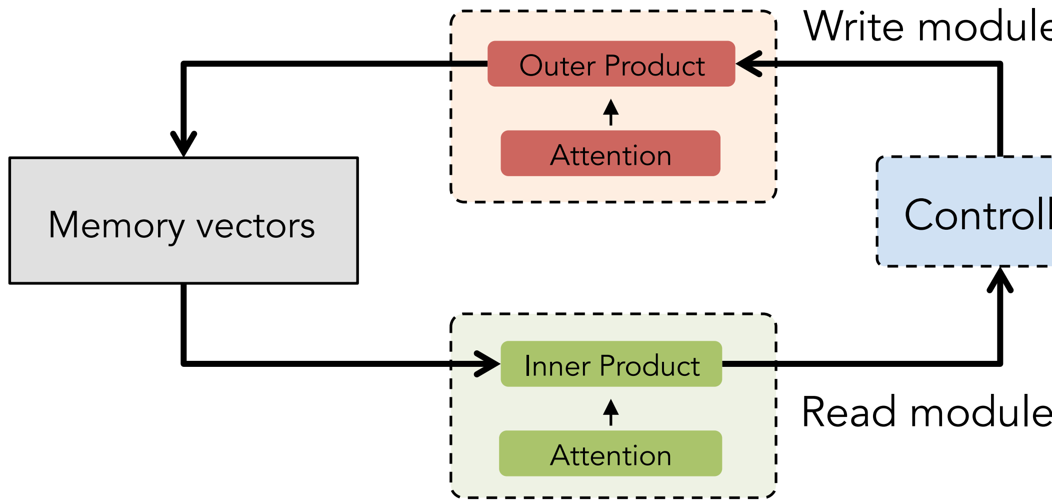
Penn-Treebank
Text8 (Wikipedia)
MemN2N
LSTM
Test perplexity
Memory hops
Memory size
124
122
120
118
116
114
112
190
180
170
160
150
140
170
165
160
155
150
145
2
3
4
5
6
7
25
50
75
100
125
150

Attention during memory hops

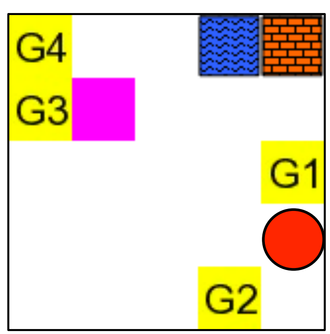


Ongoing Work

1. Writing



2. Playing games



→

Memory

1:	y-3x0	block	nil	nil	nil
2:	y-3x-1	water	nil	nil	nil
3:	y-1x0	goal1	goal	nil	nil
4:	y1x-1	goal2	goal	nil	nil
5:	y-2x-4	goal3	goal	nil	nil
6:	y-3x-4	goal4	goal	nil	nil
7:	y-2x-3	color2	switch	nil	nil
8:	y0x0	agent1	agent	nil	nil
9:	if	color1	goal3	switch	info
10:	if	color2	goal2	switch	info

http://arxiv.org/abs/1511.07401

Conclusion

- Proposed a neural net model with external memory
  - Soft attention over memory locations
  - End-to-end training with backpropagation
- Good results on a toy QA tasks
- Comparable to LSTM on language modeling
- Versatile model: also apply to writing and games

Code http://github.com/facebook/MemNN Poster #7

Thank you!

Code http://github.com/facebook/MemNN Poster #7

References (I)

- J. Weston, S. Chopra, and A. Bordes. Memory networks. In International Conference on Learning Representations (ICLR), 2015
- J. Weston, A. Bordes, S. Chopra, and T. Mikolov. Towards AI-complete question answering: a set of prerequisite toy tasks. arXiv preprint: 1502.05698, 2015
- D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. ICLR, 2015
- K. Xu, J. L. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. ICML, 2015
- L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. Describing videos by exploiting temporal structure. ICCV, 2015
- J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio. Attention-based models for speech recognition. NIPS, 2015

References (II)

- O. Vinyals, M. Fortunato, and N. Jaitly. Pointer networks. NIPS, 2015
- M. C. Mozer and S. Das. A connectionist symbol manipulator that discovers the structure of context-free languages. NIPS, 1993
- A. Joulin, and T. Mikolov. Inferring Algorithmic Patterns with Stack-Augmented Recurrent Nets. NIPS, 2015
- A. Graves, G. Wayne, and I. Danihelka. Neural Turing Machines. arXiv preprint: 1410.5401, 2014
- A. Kumar, O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, and R. Socher. Ask Me Anything: Dynamic Memory Networks for Natural Language Processing. arXiv preprint: 1506.07285, 2015
- K. Hermann, T. Kočiský, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom. Teaching Machines to Read and Comprehend, arXiv preprint: 1506.03340, 2015

Results on 1k training data

	Baseline			MemN2N								
Task	Strongly Supervised MemNN [21]	LSTM [21]	MemNN WSH	BoW	PE	PE LS	PE LS RN	1 hop PE LS joint	2 hops PE LS joint	3 hops PE LS joint	PE LS RN joint	PE LS LW joint
1: 1 supporting fact	0.0	50.0	0.1	0.6	0.1	0.2	0.0	0.8	0.0	0.1	0.0	0.1
2: 2 supporting facts	0.0	80.0	42.8	17.6	21.6	12.8	8.3	62.0	15.6	14.0	11.4	18.8
3: 3 supporting facts	0.0	80.0	76.4	71.0	64.2	58.8	40.3	76.9	31.6	33.1	21.9	31.7
4: 2 argument relations	0.0	39.0	40.3	32.0	3.8	11.6	2.8	22.8	2.2	5.7	13.4	17.5
5: 3 argument relations	2.0	30.0	16.3	18.3	14.1	15.7	13.1	11.0	13.4	14.8	14.4	12.9
6: yes/no questions	0.0	52.0	51.0	8.7	7.9	8.7	7.6	7.2	2.3	3.3	2.8	2.0
7: counting	15.0	51.0	36.1	23.5	21.6	20.3	17.3	15.9	25.4	17.9	18.3	10.1
8: lists/sets	9.0	55.0	37.8	11.4	12.6	12.7	10.0	13.2	11.7	10.1	9.3	6.1
9: simple negation	0.0	36.0	35.9	21.1	23.3	17.0	13.2	5.1	2.0	3.1	1.9	1.5
10: indefinite knowledge	2.0	56.0	68.7	22.8	17.4	18.6	15.1	10.6	5.0	6.6	6.5	2.6
11: basic coreference	0.0	38.0	30.0	4.1	4.3	0.0	0.9	8.4	1.2	0.9	0.3	3.3
12: conjunction	0.0	26.0	10.1	0.3	0.3	0.1	0.2	0.4	0.0	0.3	0.1	0.0
13: compound coreference	0.0	6.0	19.7	10.5	9.9	0.3	0.4	6.3	0.2	1.4	0.2	0.5
14: time reasoning	1.0	73.0	18.3	1.3	1.8	2.0	1.7	36.9	8.1	8.2	6.9	2.0
15: basic deduction	0.0	79.0	64.8	24.3	0.0	0.0	0.0	46.4	0.5	0.0	0.0	1.8
16: basic induction	0.0	77.0	50.5	52.0	52.1	1.6	1.3	47.4	51.3	3.5	2.7	51.0
17: positional reasoning	35.0	49.0	50.9	45.4	50.1	49.0	51.0	44.4	41.2	44.5	40.4	42.6
18: size reasoning	5.0	48.0	51.3	48.1	13.6	10.1	11.1	9.6	10.3	9.2	9.4	9.2
19: path finding	64.0	92.0	100.0	89.7	87.4	85.6	82.8	90.7	89.9	90.2	88.0	90.6
20: agent's motivation	0.0	9.0	3.6	0.1	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.2
Mean error (%)	6.7	51.3	40.2	25.1	20.3	16.3	13.9	25.8	15.6	13.3	12.4	15.2
Failed tasks (err. $> 5\%$)	4	20	18	15	13	12	11	17	11	11	11	10
On 10k training data												
Mean error (%)	3.2	36.4	39.2	15.4	9.4	7.2	6.6	24.5	10.9	7.9	7.5	11.0
Failed tasks (err. $> 5\%$)	2	16	17	9	6	4	4	16	7	6	6	6

Table 1: Test error rates (%) on the 20 QA tasks for models using 1k training examples (mean test errors for 10k training examples are shown at the bottom). Key: BoW = bag-of-words representation; PE = position encoding representation; LS = linear start training; RN = random injection of time index noise; LW = RNN-style layer-wise weight tying (if not stated, adjacent weight tying is used); joint = joint training on all tasks (as opposed to per-task training).

Results on 10k training data

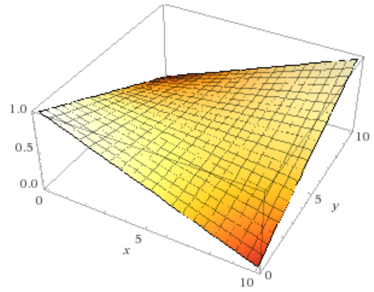
	Baseline			MemN2N									
Task	Strongly Supervised MemNN	LSTM	MemNN WSH	BoW	PE	PE LS	PE LS RN	PE LS LW RN*	1 hop PE LS joint	2 hops PE LS joint	3 hops PE LS joint	PE LS RN joint	PE LS LW joint
1: 1 supporting fact	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2: 2 supporting facts	0.0	81.9	39.6	0.6	0.4	0.5	0.3	0.3	62.0	1.3	2.3	1.0	0.8
3: 3 supporting facts	0.0	83.1	79.5	17.8	12.6	15.0	9.3	2.1	80.0	15.8	14.0	6.8	18.3
4: 2 argument relations	0.0	0.2	36.6	31.8	0.0	0.0	0.0	0.0	21.4	0.0	0.0	0.0	0.0
5: 3 argument relations	0.3	1.2	21.1	14.2	0.8	0.6	0.8	0.8	8.7	7.2	7.5	6.1	0.8
6: yes/no questions	0.0	51.8	49.9	0.1	0.2	0.1	0.0	0.1	6.1	0.7	0.2	0.1	0.1
7: counting	3.3	24.9	35.1	10.7	5.7	3.2	3.7	2.0	14.8	10.5	6.1	6.6	8.4
8: lists/sets	1.0	34.1	42.7	1.4	2.4	2.2	0.8	0.9	8.9	4.7	4.0	2.7	1.4
9: simple negation	0.0	20.2	36.4	1.8	1.3	2.0	0.8	0.3	3.7	0.4	0.0	0.0	0.2
10: indefinite knowledge	0.0	30.1	76.0	1.9	1.7	3.3	2.4	0.0	10.3	0.6	0.4	0.5	0.0
11: basic coreference	0.0	10.3	25.3	0.0	0.0	0.0	0.0	0.1	8.3	0.0	0.0	0.0	0.4
12: conjunction	0.0	23.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0
13: compound coreference	0.0	6.1	12.3	0.0	0.1	0.0	0.0	0.0	5.6	0.0	0.0	0.0	0.0
14: time reasoning	0.0	81.0	8.7	0.0	0.2	0.0	0.0	0.1	30.9	0.2	0.2	0.0	1.7
15: basic deduction	0.0	78.7	68.8	12.5	0.0	0.0	0.0	0.0	42.6	0.0	0.0	0.2	0.0
16: basic induction	0.0	51.9	50.9	50.9	48.6	0.1	0.4	51.8	47.3	46.4	0.4	0.2	49.2
17: positional reasoning	24.6	50.1	51.1	47.4	40.3	41.1	40.7	18,6	40.0	39.7	41.7	41.8	40.0
18: size reasoning	2.1	6.8	45.8	41.3	7.4	8.6	6.7	5.3	9.2	10.1	8.6	8.0	8.4
19: path finding	31.9	90.3	100.0	75.4	66.6	66.7	66.5	2.3	91.0	80.8	73.3	75.7	89.5
20: agent's motivation	0.0	2.1	4.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Mean error (%)	3.2	36.4	39.2	15.4	9.4	7.2	6.6	4.2	24.5	10.9	7.9	7.5	11.0
Failed tasks (err. $> 5\%$)	2	16	17	9	6	4	4	3	16	7	6	6	6

Table 1: Test error rates (%) on the 20 bAbI QA tasks for models using 10k training examples. Key: BoW = bag-of-words representation; PE = position encoding representation; LS = linear start training; RN = random injection of time index noise; LW = RNN-style layer-wise weight tying (if not stated, adjacent weight tying is used); joint = joint training on all tasks (as opposed to per-task training); $*$ = this is a larger model with non-linearity (embedding dimension is $d = 100$ and ReLU applied to the internal state after each hop. This was inspired by [1] and crucual for getting better performance on tasks 17 and 19).

Sentence Representation

- Bag-of-Words
  - Embed each word into vectors and add them
- Position Encoding
  - Apply simple order dependent transformation before adding

$$l_{kj} = (1 - j/J) - (k/d)(1 - 2j/J)$$



Results on language modeling

Model	Penn Treebank					Text8				
	# of hidden	# of hops	memory size	Valid. perp.	Test perp.	# of hidden	# of hops	memory size	Valid. perp.	Test perp.
RNN [15]	300	-	-	133	129	500	-	-	-	184
LSTM [15]	100	-	-	120	115	500	-	-	122	154
SCRN [15]	100	-	-	120	115	500	-	-	-	161
MemN2N	150	2	100	128	121	500	2	100	152	187
	150	3	100	129	122	500	3	100	142	178
	150	4	100	127	120	500	4	100	129	162
	150	5	100	127	118	500	5	100	123	154
	150	6	100	122	115	500	6	100	124	155
	150	7	100	120	114	500	7	100	118	**147**
	150	6	25	125	118	500	6	25	131	163
	150	6	50	121	114	500	6	50	132	166
	150	6	75	122	114	500	6	75	126	158
	150	6	100	122	115	500	6	100	124	155
	150	6	125	120	112	500	6	125	125	157
	150	6	150	121	114	500	6	150	123	154
	150	7	200	118	**111**	-	-	-	-	-

Table 2: The perplexity on the test sets of Penn Treebank and Text8 corpora. Note that increasing the number of memory hops improves performance.