Language Understanding and Reasoning with Memory Augmented Neural Nets

Tsendsuren Munkhdalai

joint work with Hong Yu

tsendsuren.munkhdalai@umassmed.edu

www.tsendeemts.com

Overview

- Neural Semantic Encoders
- Language Comprehension with Neural Semantic Encoders
- Discussion

Neural Semantic Encoders

What is an Encoder in NLP?

- Most NLP problems involve language/text encoding
  - **Essential topic/operation in neural NLP:** Symbols → vector
- Sequential neural encoders:
  - RNN/LSTM (+attention) reads text word by word
  - Don't get to see the future words in sentence
  - **Restricted to the sequential order!**
- Recursive neural encoders: Syntax parse tree based
- **Neural Semantic Encoders: memory enhanced neural encoder!**
  - **Sees whole input text (stored in memory)**
  - **Models multi-scale dependency and composition**
  - **Sequential and Recursive!**
- **Neural Tree Indexer: N-ary tree – fast, portable**

What is an Encoder in NLP?

- Most NLP problems involve language/text encoding
  - **Essential topic/operation in neural NLP:** Symbols ➔ vector
- Sequential neural encoders:
  - RNN/LSTM (+attention) reads text word by word
  - Don't get to see the future words in sentence
  - **Restricted to the sequential order!**
- Recursive neural encoders: Syntax parse tree based
- **Neural Semantic Encoders: memory enhanced neural encoder!**
  - **Sees whole input text (stored in memory)**
  - **Models multi-scale dependency**
  - **Sequential and Recursive!**
- **Neural Tree Indexer: N-ary tree – fast, portable**

Sequential and Recursive!
End-to-end!

Memory Augmented Neural Nets (MANNs)

- Human brain has different types of memory
  - Long/short term
  - Active/associative
- External memories in neural network
  - Provide with additional storage
  - Act as fast or slow weights
  - Encode/share declarative knowledge/repsentations and support procedural knowledge acquisition
- **Neural external memories are not coupled with neural network parameters**

Related Work

- RNNSearch NMT model (Bahdanau et al. 2014)
  - Stores source sentence states in memory
  - Reads the memory with soft-attention
- Memory Networks (Weston et al. 2014) and End-to-end Memory Networks (Sainbayar et al. 2015)
  - **Read only memory/no memory update**
    - **Is read only memory expressive enough?**
    - **Controller is single layer MLP?**
  - **Implements multi-hop read, can work with a bigger memory**
  - Applied to varies NLP tasks: QA, LM etc.
  - Different variations for mem. repsentations such as **key-value mem**.

Note: dates - first appeared on Arxiv

Related Work

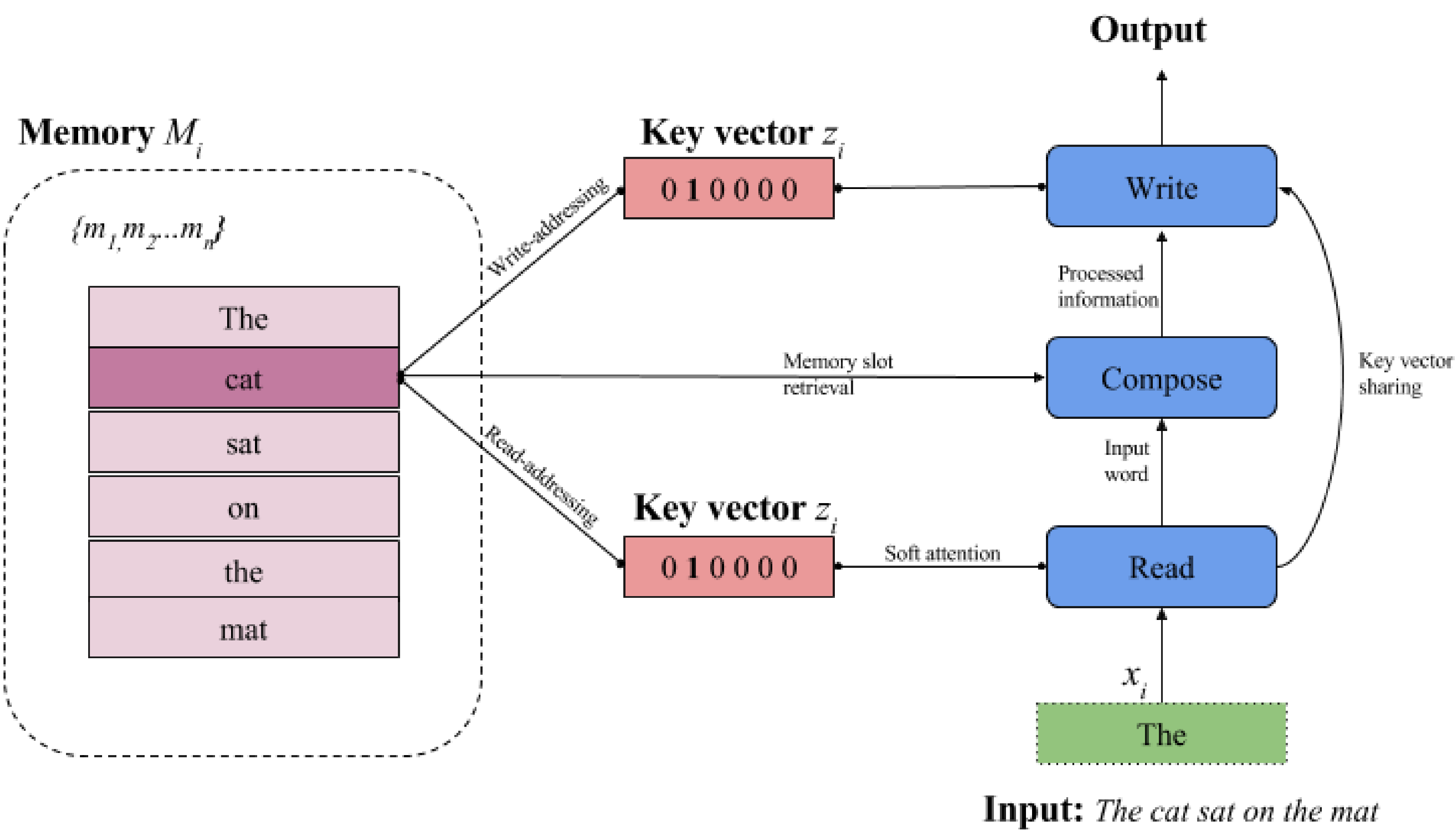
- Neural Turing Machines (Graves et al. 2014)
  - Architecture: Single controller (LSTM or MLP) and fixed memory
  - Memory access (read-write) with soft and hard attention
  - Memory update: read, erase and add weights
    - Memory manipulation overhead?
  - Addresses programming problems: copy, sort etc.
  - Not trivial to training and scale: **Information collision and memory (de-)allocation?**
    - **Fix: NTM+ (Nature paper)**

Related Work

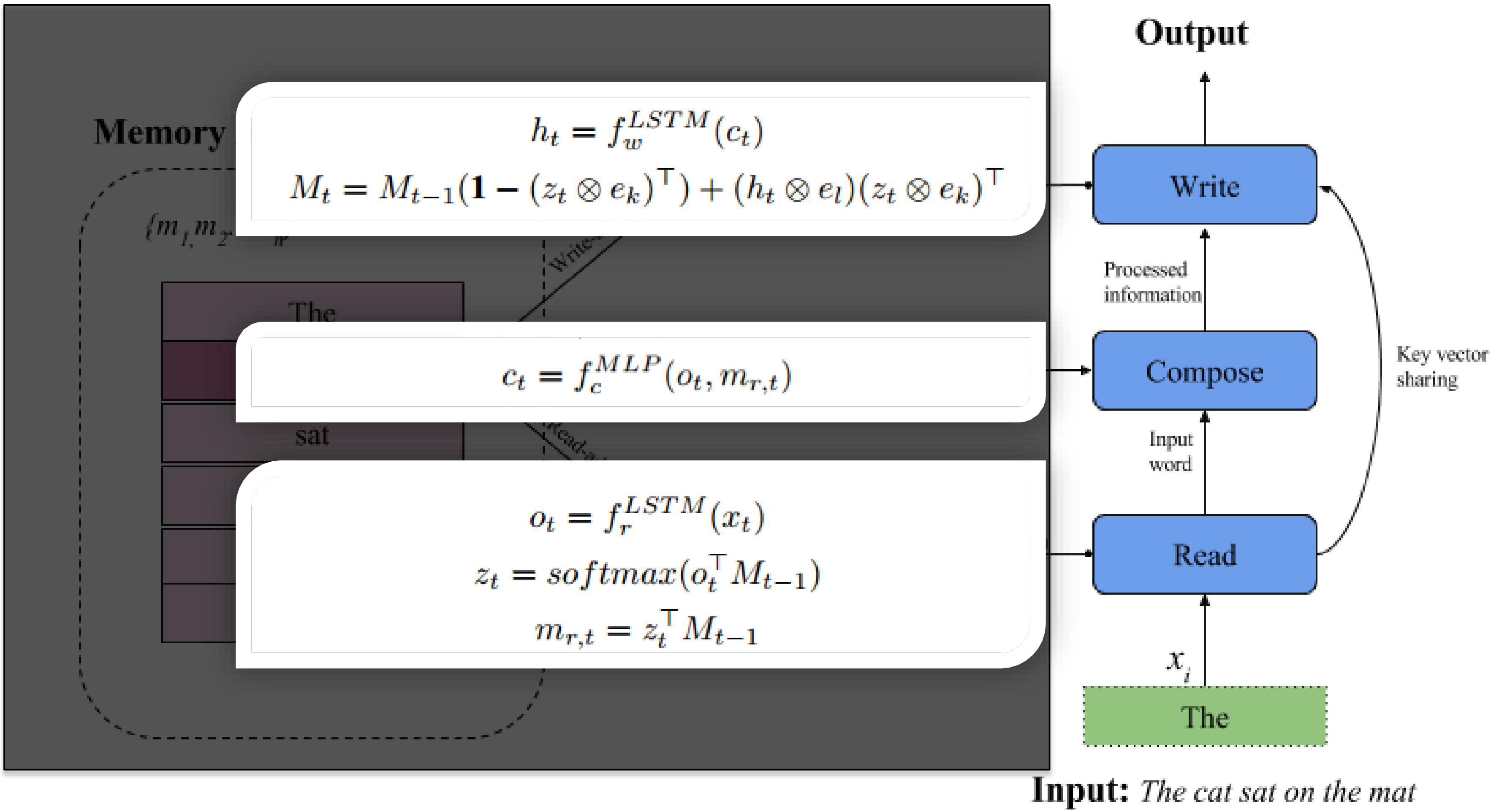
- Dynamic memory networks for NLP (Kumar et al. 2015)
- Memories based on **data structures**:
  - Stack and queue based storage
  - **The memory access is constrained by the data structure used**
  - No random memory access
- **Most previous effort on small programming tasks!**

Is Language Understanding programmable?

Neural Semantic Encoders



Neural Semantic Encoders



$$h_t = f_w^{LSTM}(c_t)$$

$$M_t = M_{t-1}(\mathbf{1} - (z_t \otimes e_k)^\top) + (h_t \otimes e_l)(z_t \otimes e_k)^\top$$

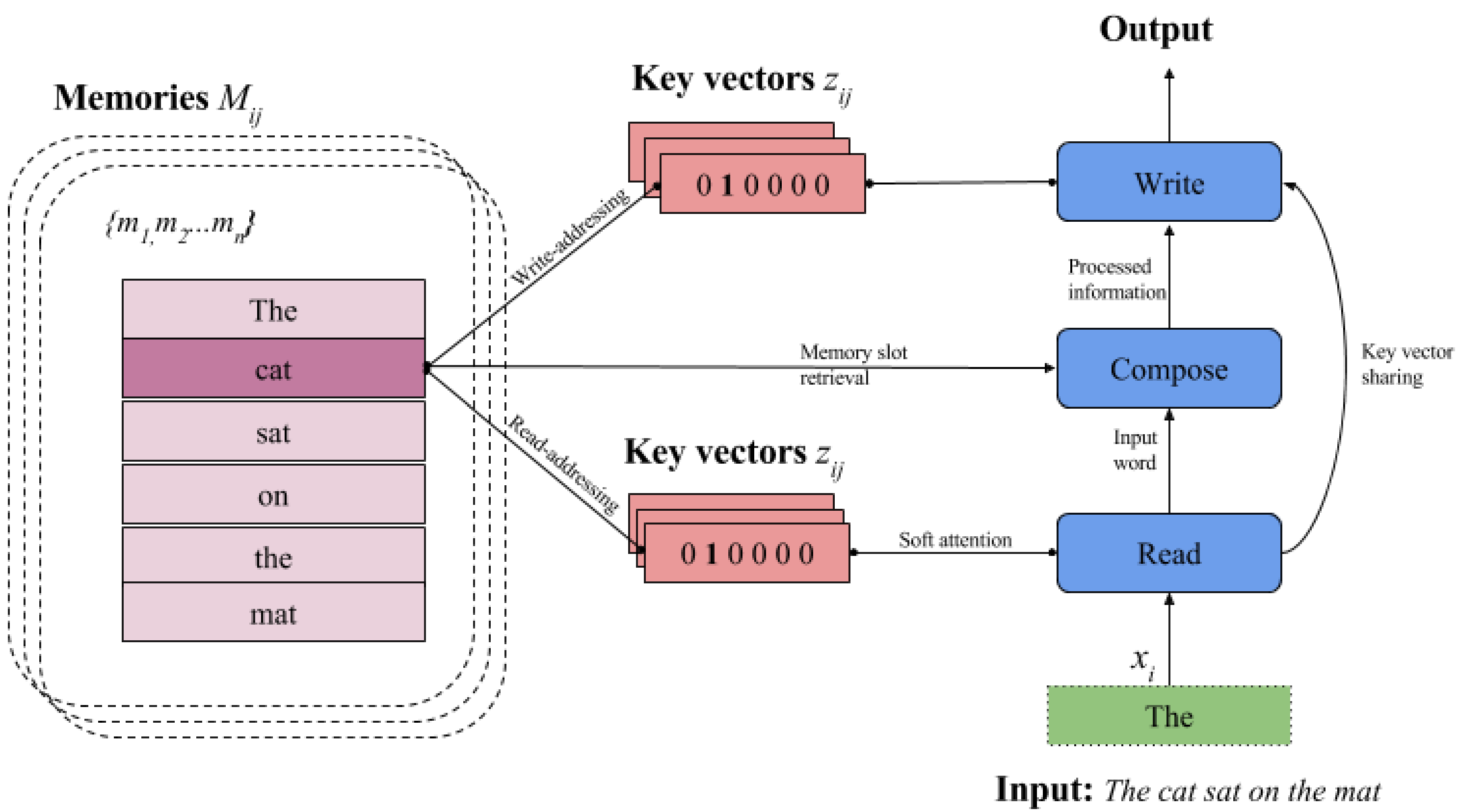
$$c_t = f_c^{MLP}(o_t, m_{r,t})$$

$$o_t = f_r^{LSTM}(x_t)$$

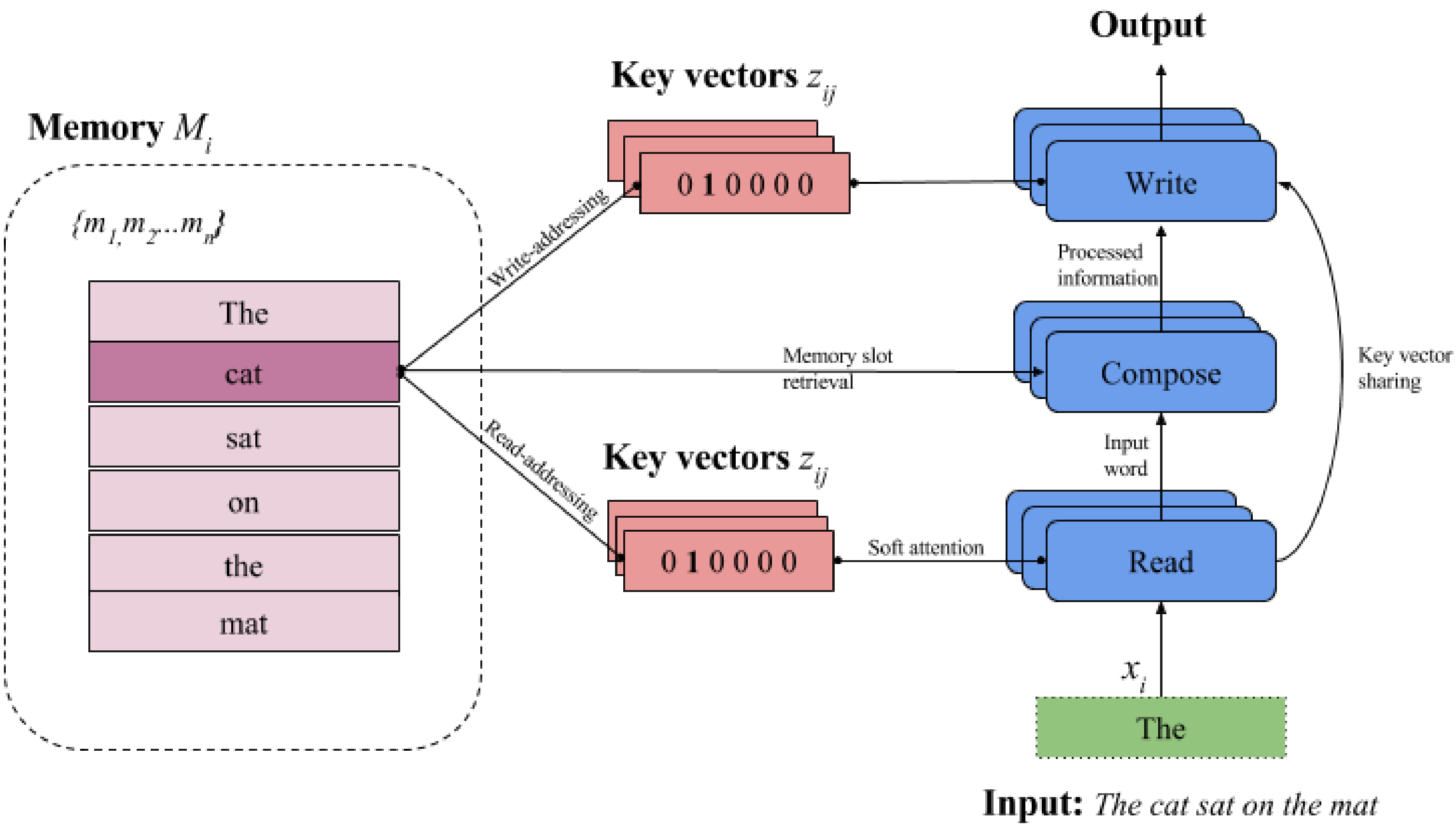
$$z_t = softmax(o_t^\top M_{t-1})$$

$$m_{r,t} = z_t^\top M_{t-1}$$

NSE Variation:
Multiple memory access

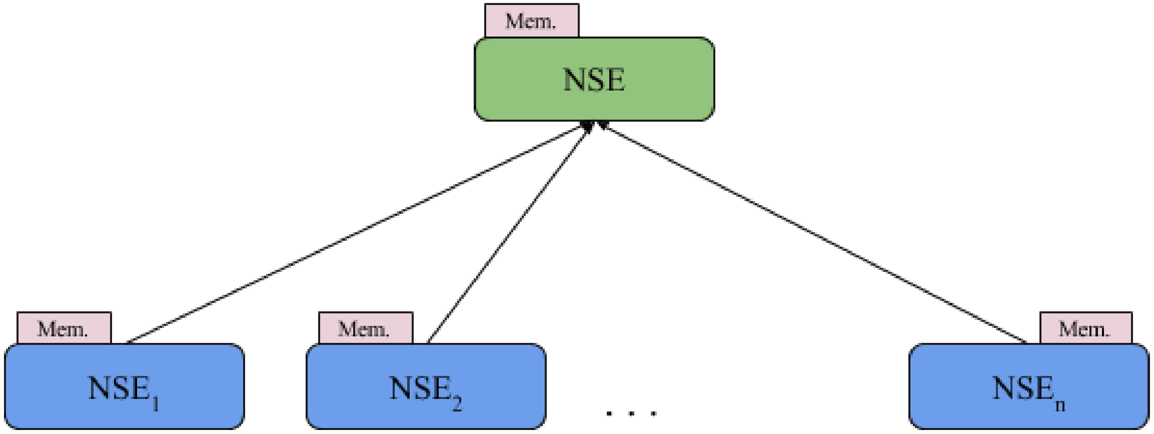


NSE Variation:
Shared memory accesses



NSE Variation: Hierarchical/Stacked NSE

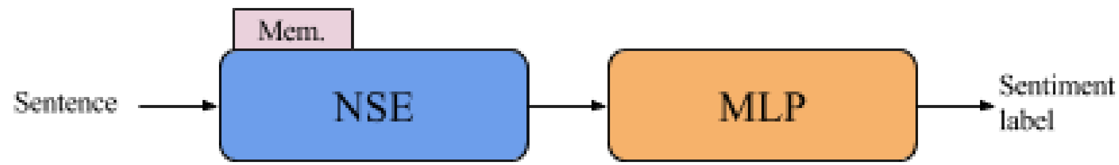
- Hierarchical/Stacked NSE is for document modeling, character level language processing etc.
  - **Lower level NSEs run in parallel, fast!**



Results

- **We applied NSE to five different NLP tasks + Language comprehension**
  - Sentence classification
  - Answer sentence selection/Non-factoid QA
  - Natural language inference
  - Document modelling
  - Neural machine translation

Results:
Sentence classification

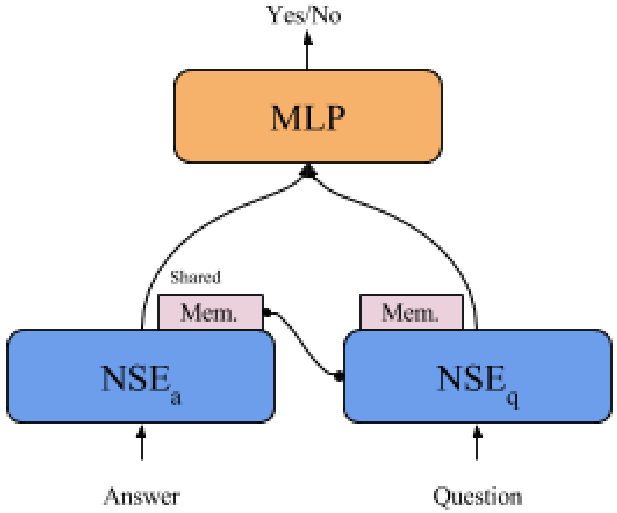
- Architecture: 

- Dataset: Stanford Sentiment Treebank (SST)
  - Train/dev/test standard splits
  - Binary and 5-label classification

Model	Bin	FG
RNTN [28]	85.4	45.7
Paragraph Vector [23]	87.8	48.7
CNN-MC [29]	88.1	47.4
DRNN [30]	86.6	49.8
2-layer LSTM[31]	86.3	46.0
Bi-LSTM[31]	87.5	49.1
CT-LSTM[31]	88.0	51.0
DMN [10]	88.6	52.1
NSE	**89.7**	**52.8**

Results:
Answer sentence selection

- **Task:** select correct answer sentence from a candidate set to answer a question
- Dataset: WikiQA
  - Train/dev/test: 20,360/2,733/6,165 QA pairs



Model	MAP	MRR
Classifier with features [22]	0.5993	0.6068
Paragraph Vector [23]	0.5110	0.5160
Bigram-CNN [24]	0.6190	0.6281
3-layer LSTM [25]	0.6552	0.6747
3-layer LSTM attention [25]	0.6639	0.6828
NASM [25]	0.6705	0.6914
MMA-NSE attention	**0.6811**	**0.6993**

Results:
Natural language inference

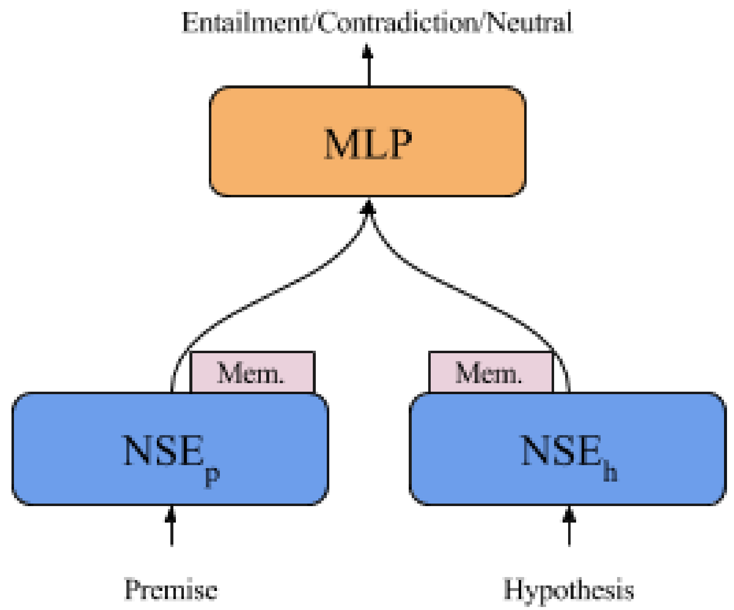
- **Task:**

Premise	Hypothesis	Relationship
A person on a horse jumps over a broken down airplane	A person is outdoors, on a horse	**Entailment**
Kids are smiling and waving at camera	The kids are frowning	**Contradiction**
A boy is jumping on skateboard	The boy is wearing safety equipment	**Neutral**

- Dataset: SNLI
  - Train/dev/test: 550K/10K/10K pairs

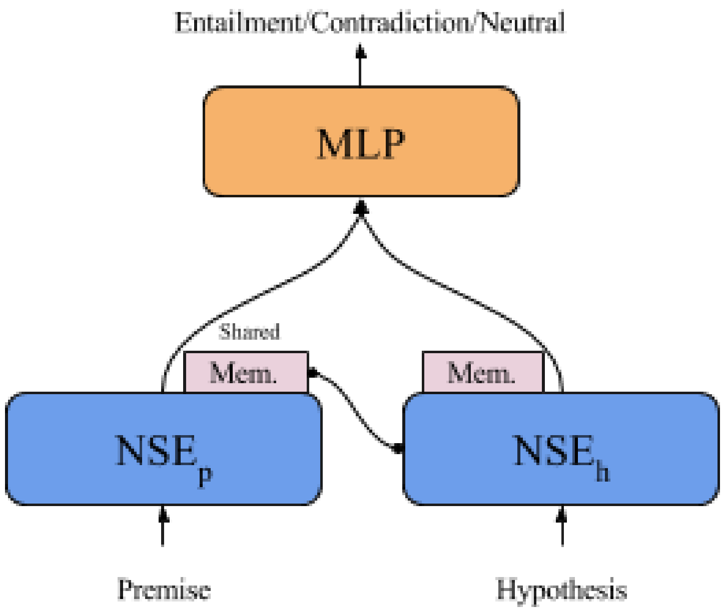
Results:
Natural language inference

- Model variations:
  - **NSE**, MMA-NSE and MMA-NSE + attention



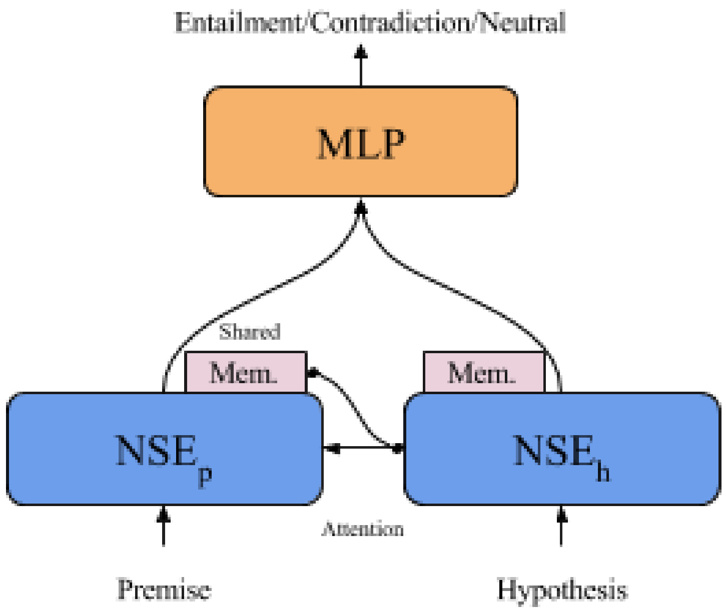
Results:
Natural language inference

- Model variations:
  - NSE, **MMA-NSE** and MMA-NSE + attention



Results:
Natural language inference

- Model variations:
  - NSE, MMA-NSE and **MMA-NSE + attention**



Results:
Natural language inference

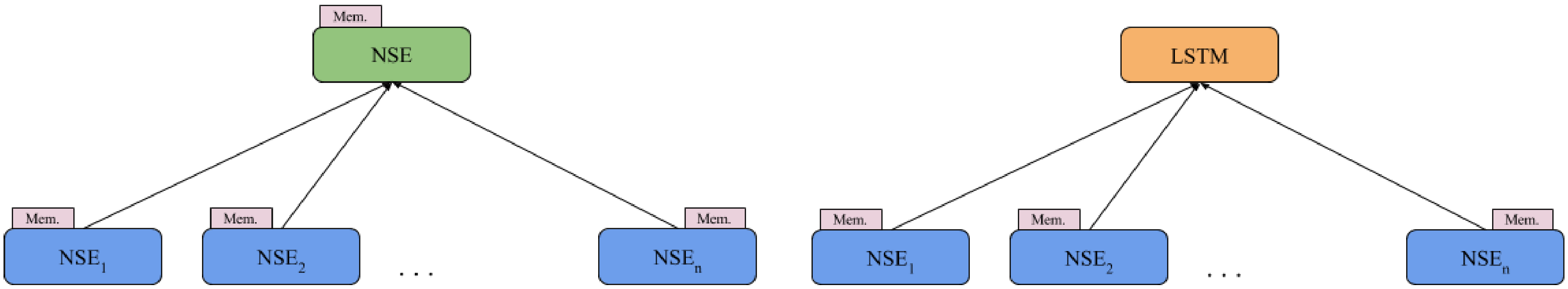
Model	d	$\|\theta\|_M$	Train	Test
Classifier with handcrafted features [12]	-	-	99.7	78.2
LSTM encoders [12]	300	3.0M	83.9	80.6
Dependency Tree CNN encoders [13]	300	3.5M	83.3	82.1
SPINN-PI encoders [14]	300	3.7M	89.2	83.2
NSE	300	3.4M	86.2	84.6
MMA-NSE	300	6.3M	87.1	**84.8**
LSTM attention [15]	100	242K	85.4	82.3
LSTM word-by-word attention [15]	100	252K	85.3	83.5
MMA-NSE attention	300	6.5M	86.9	85.4
mLSTM word-by-word attention [16]	300	1.9M	92.0	86.1
LSTMN with deep attention fusion [17]	450	3.4M	89.5	86.3
Decomposable attention model [18]	200	582K	90.5	86.8
Full tree matching NTI-SLSTM-LSTM global attention [19]	300	3.2M	88.5	**87.3**

Results:
Document modelling

- **Task:** document-level sentiment classification

Corpus	#docs	Avg. #sents	Max. #sents	#classes
Yelp 2013	335,018	8.9	151	5
IMDB	348,415	14.02	143	10

- Evaluated models: NSE-NSE and NSE-LSTM



Results: Document modelling

Model	Yelp 13		IMDB	
	Acc	MSE	Acc	MSE
Classifier [32]	59.8	0.68	40.5	3.56
PV [32]	57.7	0.86	34.1	4.69
CNN [32]	59.7	0.76	37.6	3.30
Conv-GRNN [32]	63.7	0.56	42.5	2.71
LSTM-GRNN [32]	65.1	0.50	45.3	3.00
NSE-NSE	66.6	0.48	**48.3**	**1.94**
NSE-LSTM	**67.0**	**0.47**	48.1	1.98

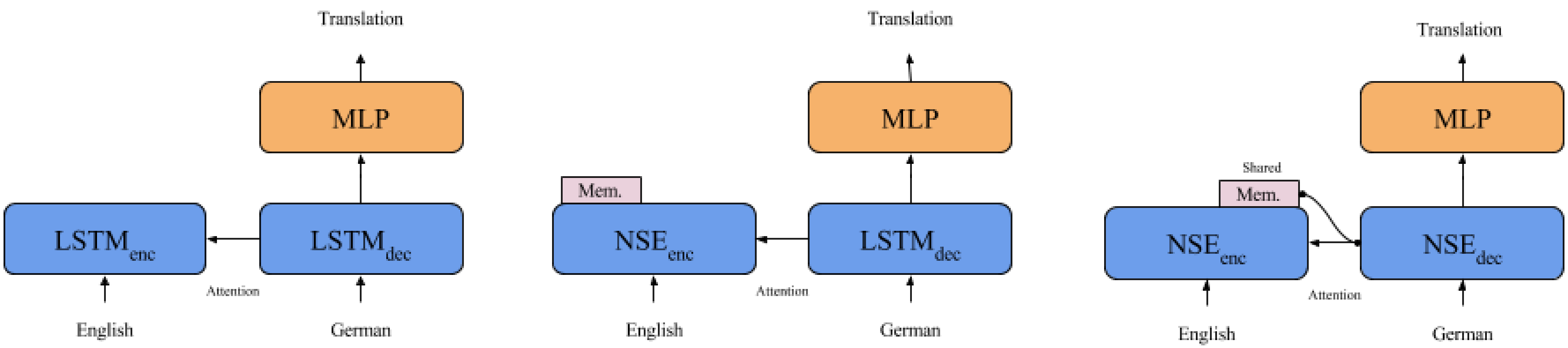
- **IMDB has longer docs with more sentences and 10 different classes**

Results:
Neural machine translation

- **NMT is formulated within encoder-decoder framework**
  - Classic example of seq2seq learning
  - Encoder: source language ➔ vector space
  - Decoder: vector space ➔ target language
- Dataset: IWSLT 2014 English-German corpus
  - train/dev/test: 110,439/4,998/4,793 pairs

Results: Neural machine translation

- Compared models:



Model	Train	Dev	Test
Baseline LSTM-LSTM	28.06	17.96	17.02
NSE-LSTM	28.73	17.67	17.13
NSE-NSE	29.89	**18.53**	**17.93**

Memory visualization

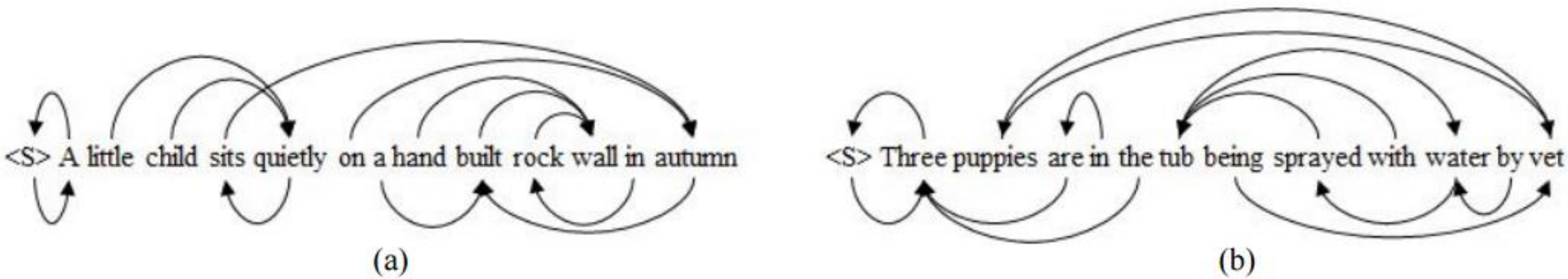


Figure 2: Word association or composition graphs produced by NSE memory access. The directed arcs connect the words that are composed via *compose* module. The source nodes are input words and the destination nodes (pointed by the arrows) correspond to the accessed memory slots. $< S >$ denotes the beginning of sequence.

Memory visualization

t=0
input:
<S>
A
little
child
sits
quietly
on
a
hand
built
rock
wall
in
autumn

t=1
input: <S>
<S>
(<S>A)
little
child
sits
quietly
on
a
hand
built
rock
wall
in
autumn

t=2
input: A
(A <S>)
(<S>A)
little
child
sits
quietly
on
a
hand
built
rock
wall
in
autumn

t=3
input: little
(A <S>)
(<S>A)
little
child
sits
(little quietly)
on
a
hand
built
rock
wall
in
autumn

t=4
input: child
(A <S>)
(<S>A)
little
child
sits
(child (little quietly))
on
a
hand
built
rock
wall
in
autumn

t=5
input: sits
(A <S>)
(<S>A)
little
child
sits
(child (little quietly))
on
a
hand
built
rock
wall
in
(sits autumn)

t=6
input: quietly
(A <S>)
(<S>A)
little
child
(quietly sits)
(child (little quietly))
on
a
hand
built
rock
wall
in
(sits autumn)

t=7
input: on
(A <S>)
(<S>A)
little
child
(quietly sits)
(child (little quietly))
on
a
hand
built
rock
wall
in
(on (sits autumn))

t=8
input: a
(A <S>)
(<S>A)
little
child
(quietly sits)
(child (little quietly))
on
a
hand
(a built)
rock
wall
in
(on (sits autumn))

t=9
input: hand
(A <S>)
(<S>A)
little
child
(quietly sits)
(child (little quietly))
on
a
hand
(a built)
rock
(hand wall)
in
(on (sits autumn))

t=10
input: built
(A <S>)
(<S>A)
little
child
(quietly sits)
(child (little quietly))
on
a
hand
(a built)
rock
(built (hand wall))
in
(on (sits autumn))

t=11
input: rock
(A <S>)
(<S>A)
little
child
(quietly sits)
(child (little quietly))
on
a
hand
(a built)
rock
(rock (built (hand wall)))
in
(on (sits autumn))

t=12
input: wall
(A <S>)
(<S>A)
little
child
(wall (quietly sits))
(child (little quietly))
on
a
hand
(a built)
rock
(rock (built (hand wall)))
in
(on (sits autumn))

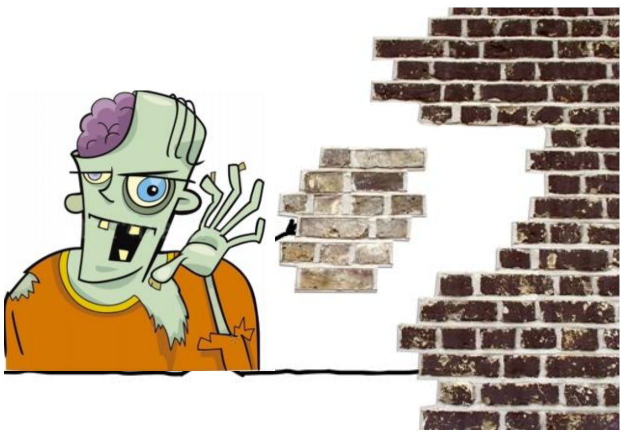
t=13
input: in
(A <S>)
(<S>A)
little
child
(wall (quietly sits))
(child (little quietly))
on
a
hand
(a built)
(in rock)
(rock (built (hand wall)))
in
(on (sits autumn))

t=14
input: autumn
(A <S>)
(<S>A)
little
child
(wall (quietly sits))
(child (little quietly))
on
a
hand
(autumn (a built))
(in rock)
(rock (built (hand wall)))
in
(on (sits autumn))

Language Comprehension with Neural Semantic Encoders

Introduction

- **Task:** given document story, find an answer for query/question related to the document
  - A large dataset can be generated automatically
- Closely related to Question Answering
  - Cloze type QA
- Some benchmark datasets:
  - CNN/Daily news (news domain)
  - CBTest (children book)
  - WDW (new domain)



Related Work

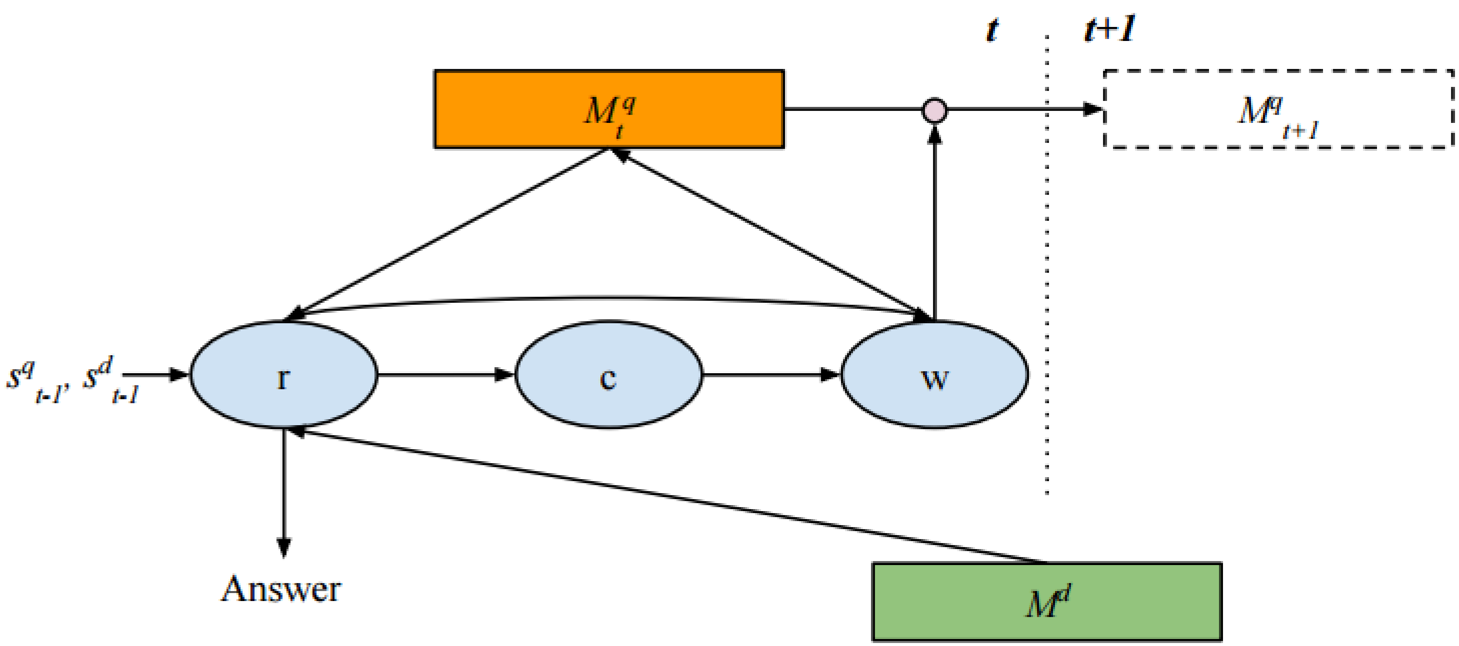
- **Single-step comprehension:** read document once to reach conclusion
  - Context modeling with bi-directional recurrent neural networks (Bi-RNN)
  - Selective focusing with attention mechanism
- **Multi-step comprehension:** read iteratively
  - Use external memory and attention
  - Retrieve query-relevant information
- **When to stop reading?**
- **How to organize and manipulate the memory?**

Hypothesis Testing with NSE

- **Hypothesis-test loop**
  - Formulate/refine (the previous) hypothesis for the correct answer and check it against the document story in each step
  - Dynamically halt the loop – correct answer is found
- Don't summarize the query
  - regress it towards completion
- **Proposed: NSE-Query gating, NSE-Adaptive computation**

Hypothesis Testing with NSE

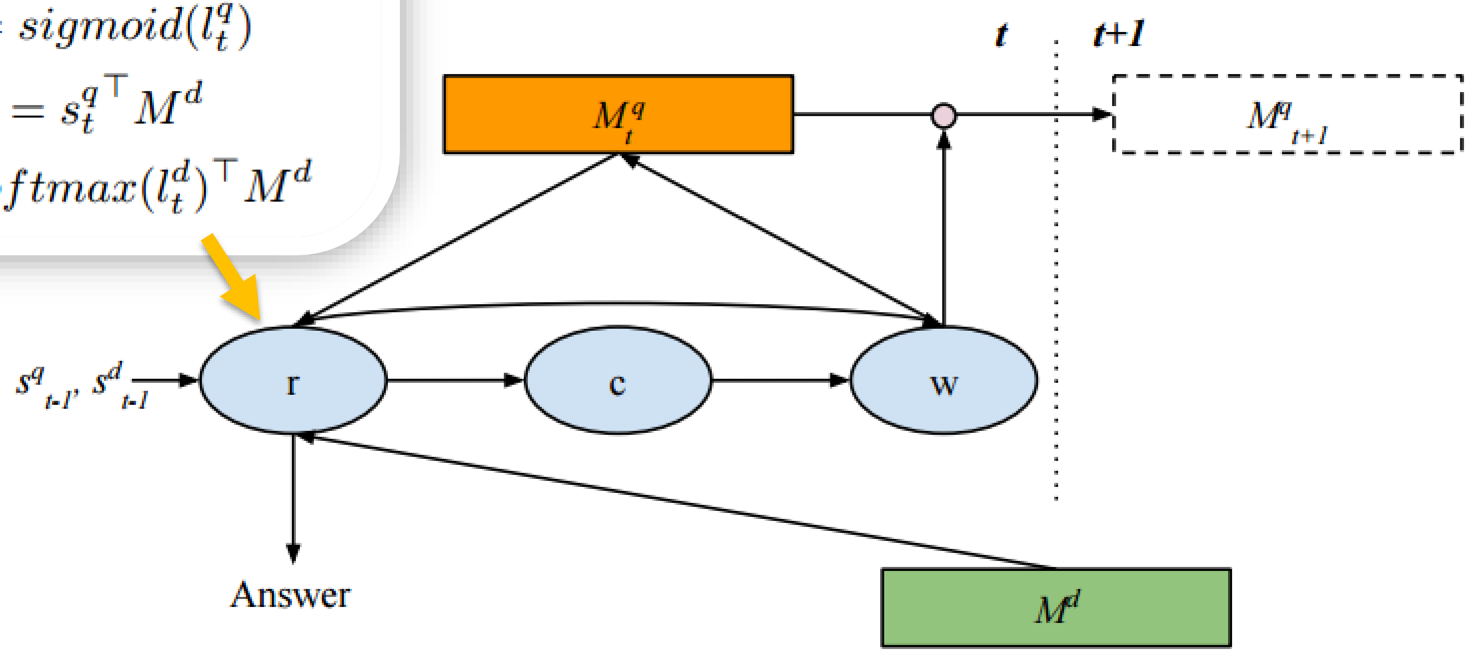
- NSE-Query gating model



Hypothesis Testing with NSE

$$r_t = read^{LSTM}([s^q_{t-1}; s^d_{t-1}])$$
$$l^q_t = r_t^\top M^q_{t-1}$$
$$s^q_t = softmax(l^q_t)^\top M^q_{t-1}$$
$$z^q_t = sigmoid(l^q_t)$$
$$l^d_t = {s^q_t}^\top M^d$$
$$s^d_t = softmax(l^d_t)^\top M^d$$

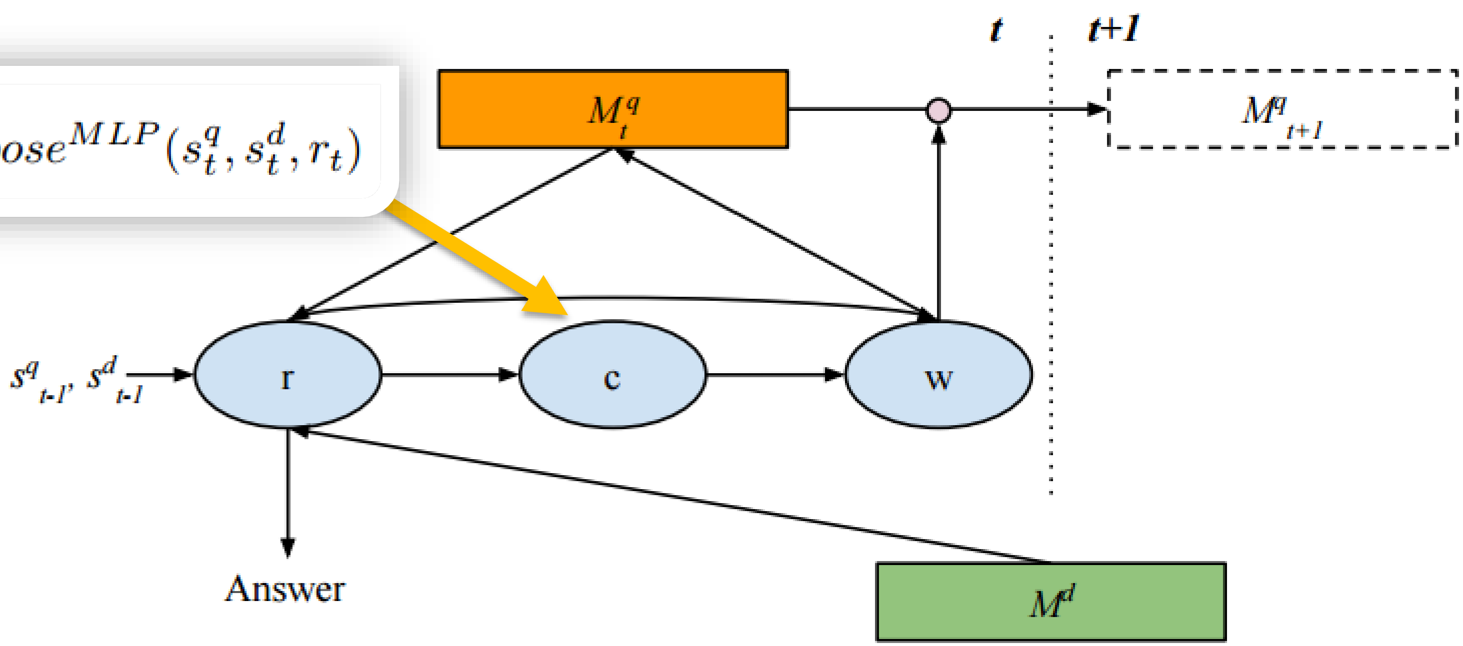
ating model



Hypothesis Testing with NSE

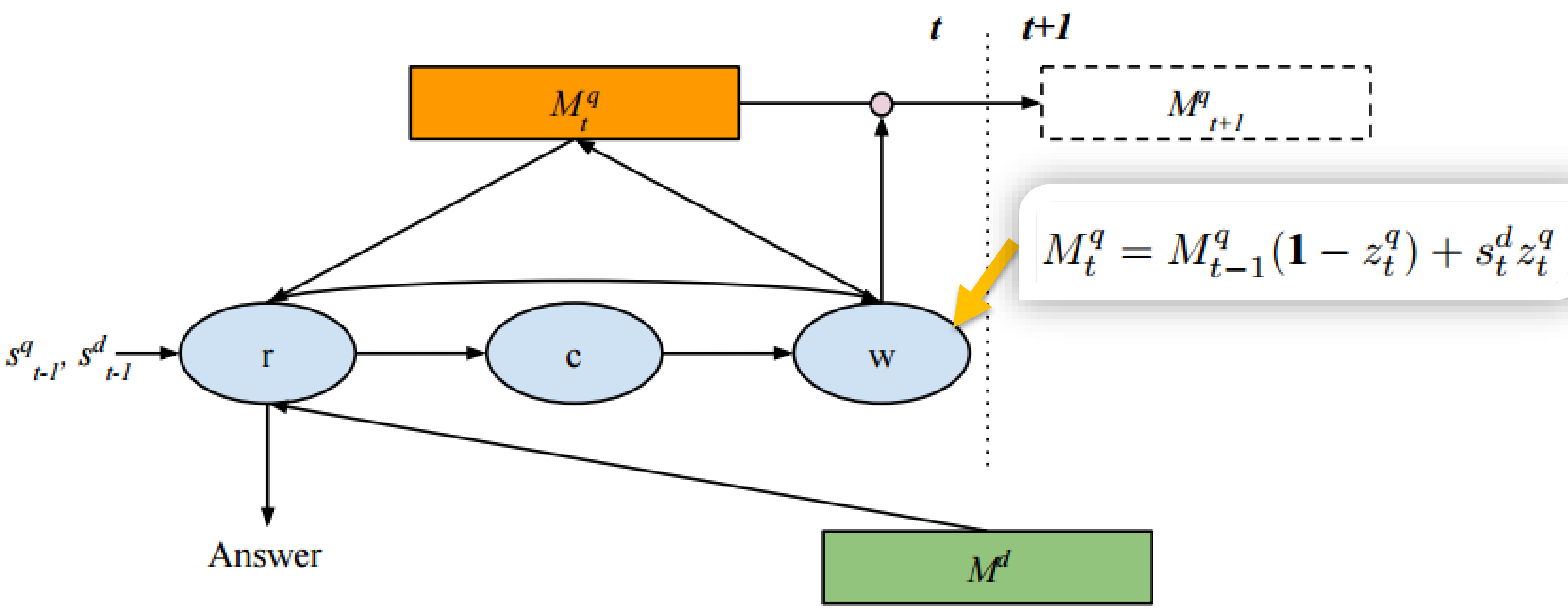
- NSE-Query gating model

$$c_t = compose^{MLP}(s_t^q, s_t^d, r_t)$$



Hypothesis Testing with NSE

- NSE-Query gating model



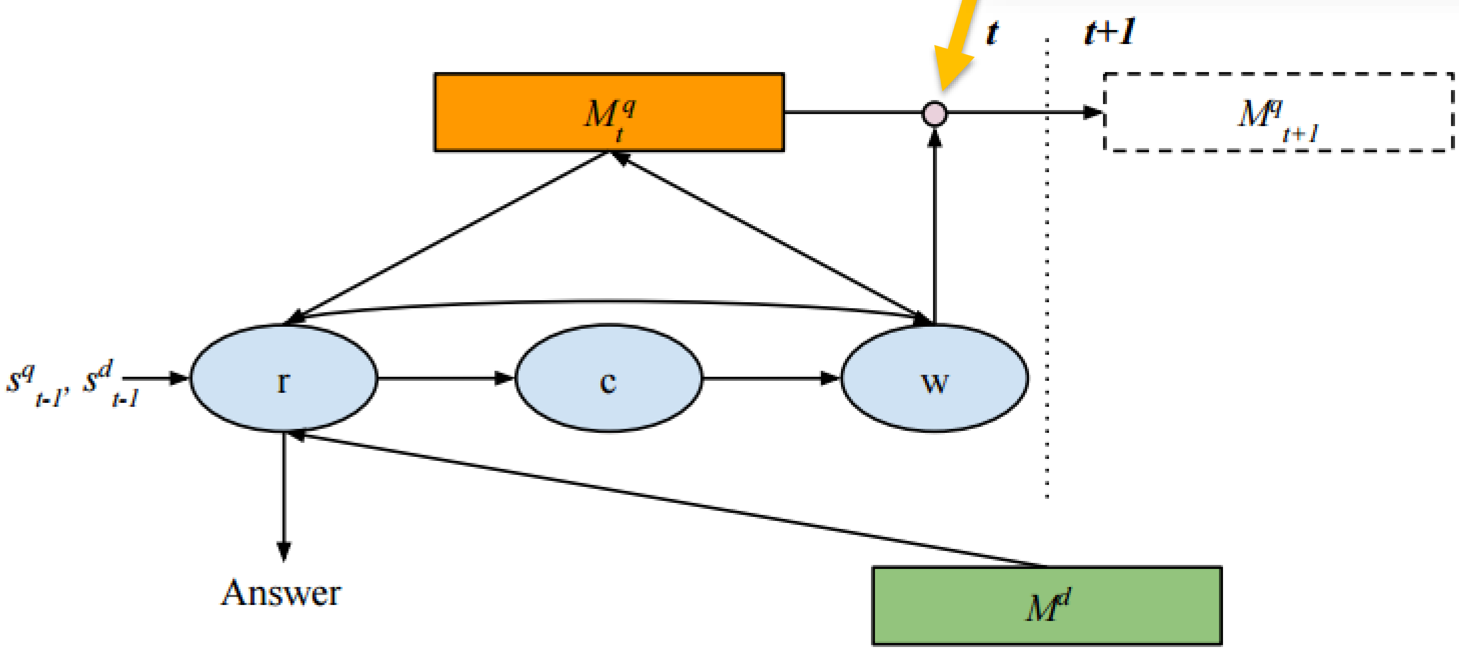
Hypothesis Testing with NSE

- NSE-Query gating model

$$w_t = write^{LSTM}(c_t)$$

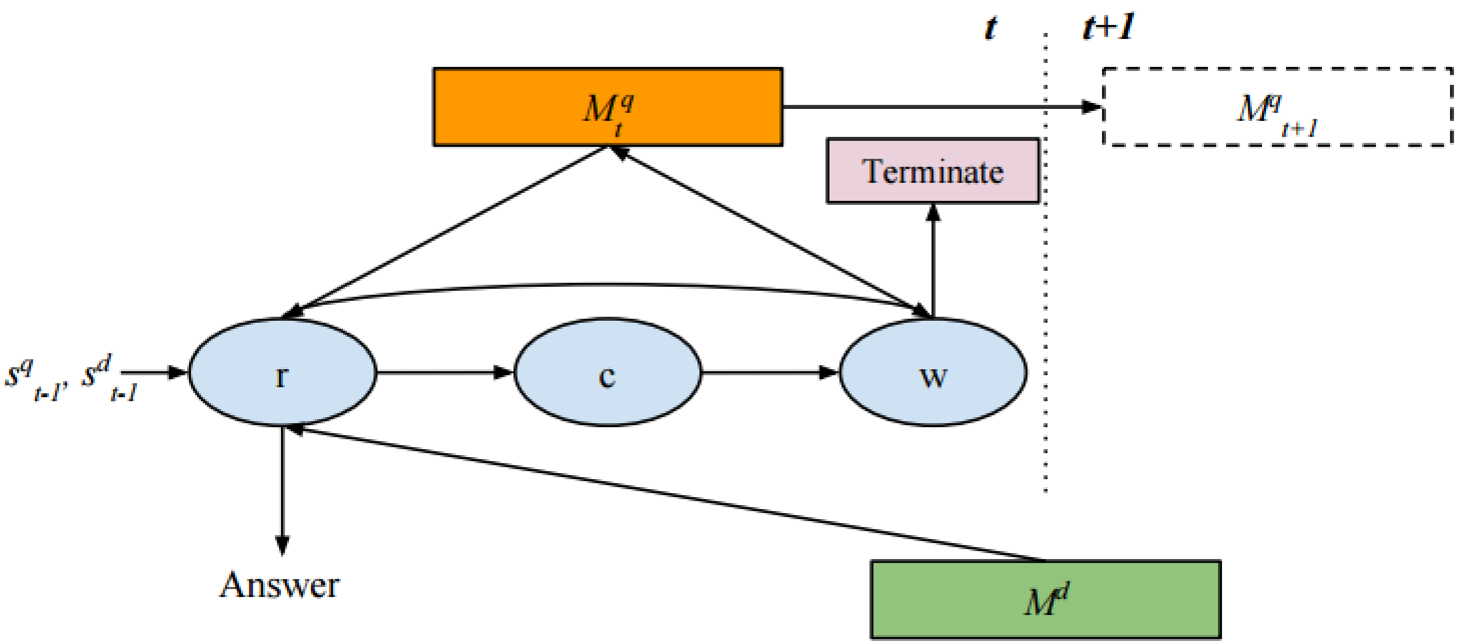
$$g_t^q = sigmoid(w_t^\top M_{t-1}^q)$$

$$M_t^q = M_t^q(\mathbf{1} - g_t^q) + M_{t-1}^q g_t^q$$



Hypothesis Testing with NSE

- NSE-Adaptive computation



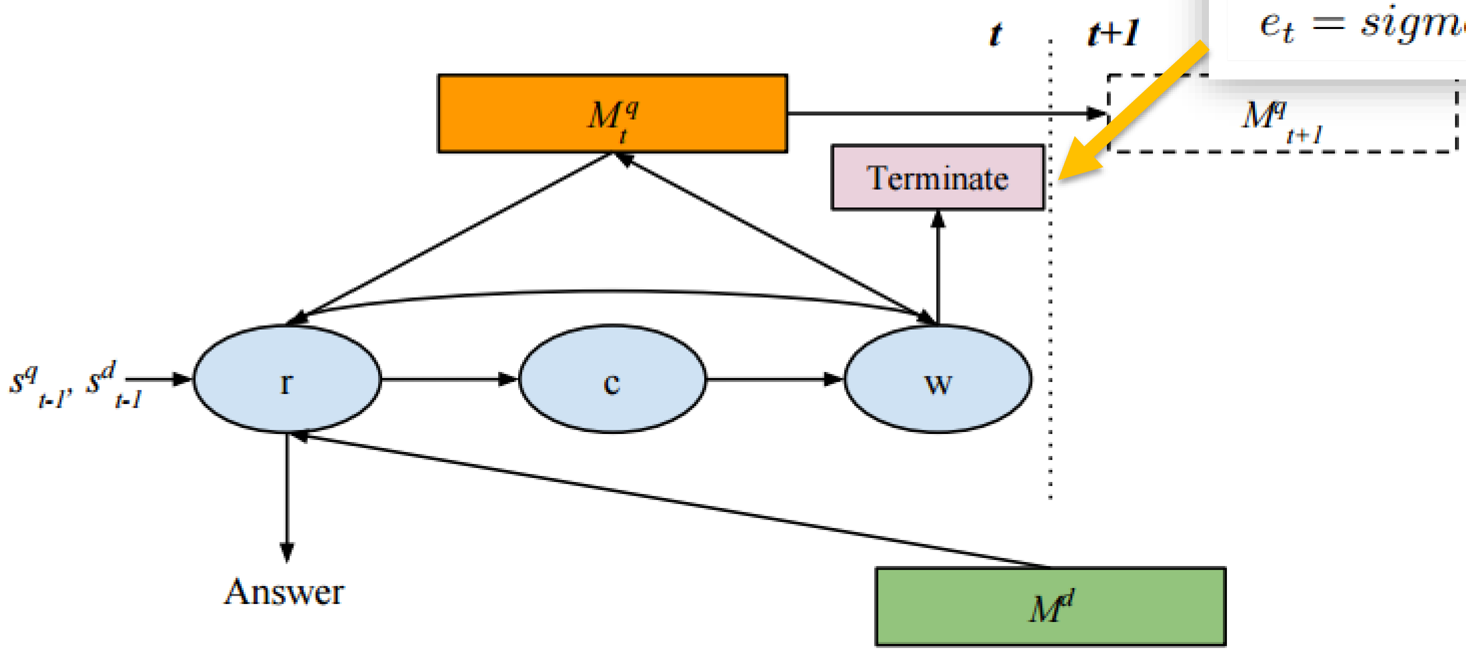
(b)

Hypothesis Testing with NSE

- NSE-Adaptive computation

$$w_t = write^{LSTM}(c_t)$$

$$e_t = sigmoid(o^\top w_t)$$



(b)

Results

- Datasets: CBTest and WDW
- **Sub-tasks**
  - CBT-NE and CBT-CN
  - WDW strict and WDW relaxed

Table 1: Statistics of the datasets. train (s): train strict, train (r): train relaxed and cands: candidates.

	WDW				CBT-NE			CBT-CN		
	train (s)	train (r)	dev	test	train	dev	test	train	dev	test
# queries	127,786	185,978	10,000	10,000	108,719	2,000	2,500	120,769	2,000	2,500
avg. # cands	3.5	3.5	3.4	3.4	10	10	10	10	10	10
avg. # tokens	365	378	325	326	433	412	424	470	448	461
vocab size	308,602		347,406			53,063			53,185	

Results

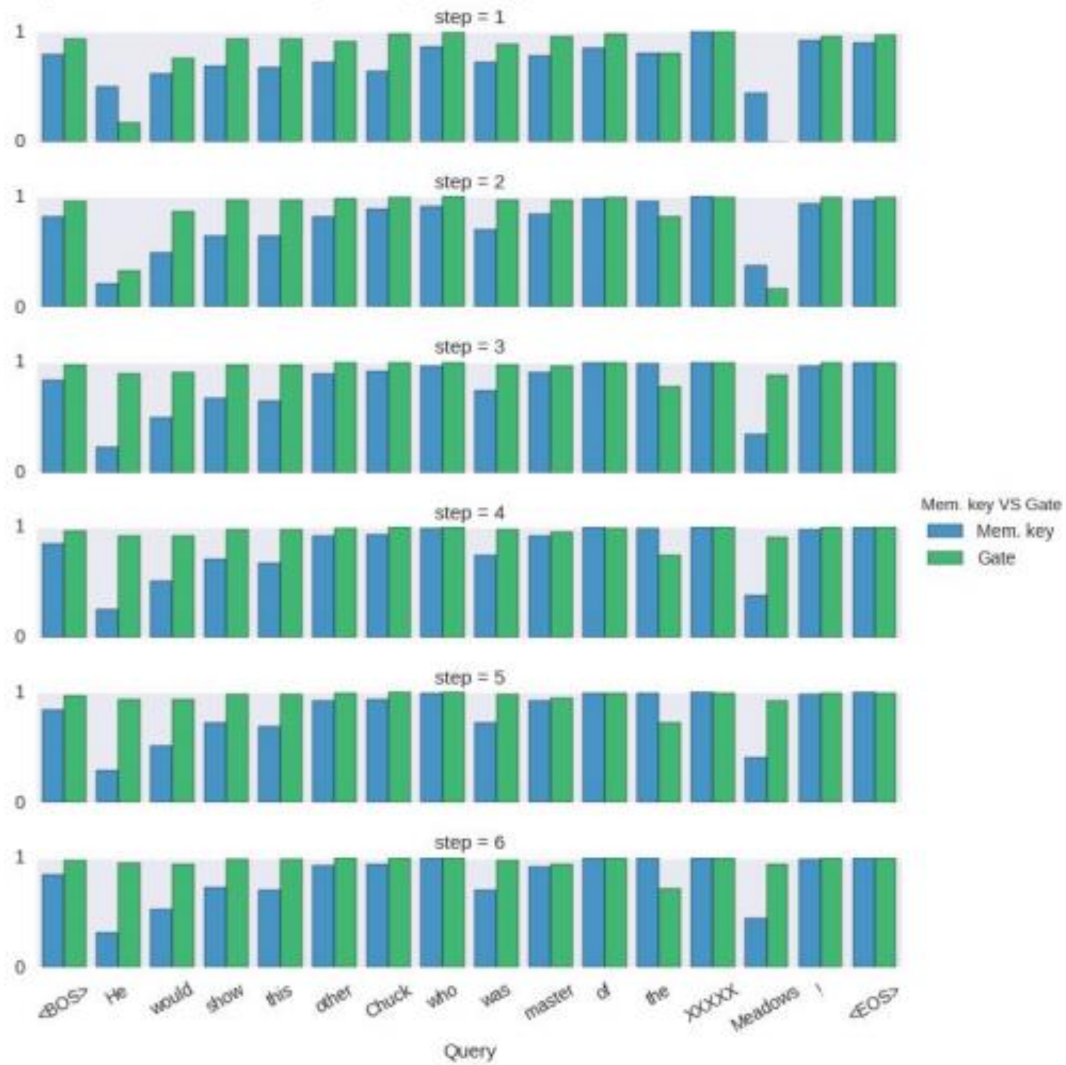
Model	CBT-NE		CBT-CN	
	dev	test	dev	test
Human (context + query) (Hill et al., 2015)	-	81.6	-	81.6
LSTMs (context + query) (Hill et al., 2015)	51.2	41.8	62.6	56.0
MemNNs (window mem. + self-sup.) (Hill et al., 2015)	70.4	66.6	64.2	63.0
AS Reader (Kadlec et al., 2016)	73.8	68.6	68.8	63.4
GA Reader (Dhingra et al., 2016)	74.9	69.0	69.0	63.9
EpiReader (Trischler et al., 2016)	75.3	69.7	71.5	67.4
IAA Reader (Sordoni et al., 2016)	75.2	68.6	72.1	69.2
AoA Reader (Cui et al., 2016)	77.8	72.0	72.2	69.4
MemNN (window mem. + self-sup. + ensemble) (Hill et al., 2015)	70.4	66.6	64.2	63.0
AS Reader (ensemble) (Kadlec et al., 2016)	74.5	70.6	71.1	68.9
EpiReader (ensemble) (Trischler et al., 2016)	76.6	71.8	73.6	70.6
IAA Reader (ensemble) (Sordoni et al., 2016)	76.9	72.0	74.1	71.0
NSE ($T = 1$)	76.2	71.1	72.8	69.7
NSE Query Gating ($T = 2$)	76.6	71.5	72.3	70.7
NSE Query Gating ($T = 6$)	77.0	71.4	73.0	**72.0**
NSE Query Gating ($T = 9$)	78.0	72.6	73.5	71.2
NSE Query Gating ($T = 12$)	77.7	72.2	**74.3**	71.9
NSE Adaptive Computation ($T = 2$)	77.1	72.1	72.8	71.2
NSE Adaptive Computation ($T = 12$)	**78.2**	**73.2**	74.2	71.4

Results

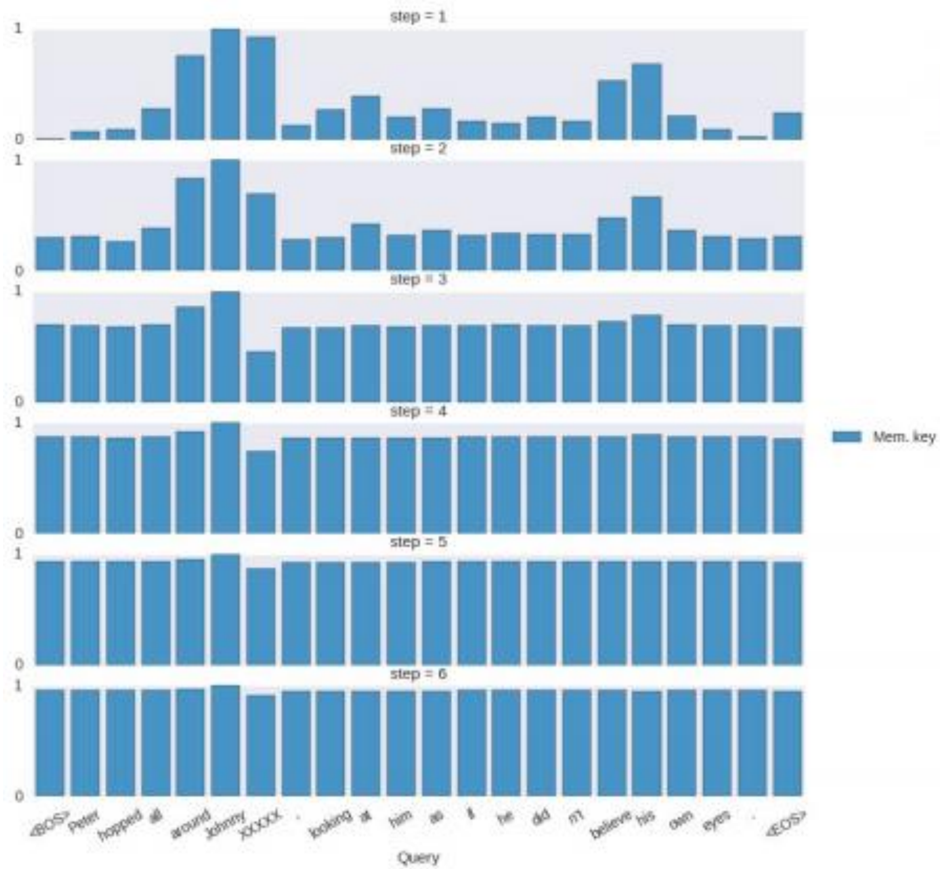
- WDW dataset

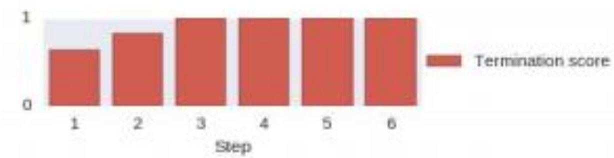
Model	Strict		Relaxed	
	dev	test	dev	test
Human (Onishi et al., 2016)	-	84.0	-	-
Attentive Reader (Hermann et al., 2015)	-	53.0	-	55.0
AS Reader (Kadlec et al., 2016)	-	57.0	-	59.0
GA Reader (Dhingra et al., 2016)	-	57.0	-	60.0
Stanford Attentive Reader (Chen et al., 2016)	-	64.0	-	65.0
NSE ($T = 1$)	65.1	65.5	66.4	65.3
NSE Query Gating ($T = 2$)	65.4	65.1	65.7	65.5
NSE Query Gating ($T = 6$)	65.5	65.7	65.6	65.8
NSE Query Gating ($T = 9$)	65.8	65.8	65.8	65.9
NSE Query Gating ($T = 12$)	65.2	65.5	65.7	65.4
NSE Adaptive Computation ($T = 2$)	65.3	65.4	66.2	66.0
NSE Adaptive Computation ($T = 12$)	**66.5**	**66.2**	**67.0**	**66.7**

Query Regression Visualization: NSE-Query gating



Query Regression Visualization: NSE-Adaptive computation





Discussion

- Memory and attention can be useful tool for efficient NLP
- Questions to ask:
  - How to organize the memory?
  - How to manipulate the memory?
    - What is the update rule?
    - Avoid the curse of memory - memory manipulation overhead
  - What would be the controller architecture?
  - Is your MANN scalable, flexible etc.?

Thank you!

Publications

- Munkhdalai, Tsendsuren, and Hong Yu. "Neural Semantic Encoders." (EACL 2017)
- Munkhdalai, Tsendsuren, and Hong Yu. "Reasoning with memory augmented neural networks for language comprehension." (ICLR 2017)
- Munkhdalai, Tsendsuren, and Hong Yu. "Neural Tree Indexers." (EACL 2017)