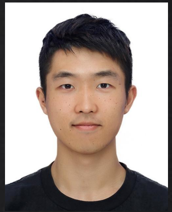
DL Chatbot seminar
Day 04

QA with External Memory

HELLO!

I am Jaemin Cho

- Vision & Learning Lab @ SNU
- NLP / ML / Generative Model
- Looking for Ph.D. / Research programs



You can find me at:

- heythisischo@gmail.com
- j-min
- J-min Cho
- Jaemin Cho

Today We will cover

- ✘ External Memory
  - PyTorch Tutorial
- ✘ Advanced External Memory Architecture
- ✘ Advanced Dialogue model
- ✘ Wrap Up this Seminar!

1.

I found this slide very helpful!

External Memory

Memory Networks / End-to-End Memory Networks
Key-Value Memory Networks
Dynamic Memory Networks
Neural Turing Machine

RNN

x_t h_t

- 외부 자극은 내 상태에 어떻게 영향을 미칠까? $\Rightarrow h_{t+1}$

$$h_{t+1} = \underbrace{W_{x \to h}}_{\text{how much to update}} x_t + \underbrace{W_{h \to h}}_{\text{how much to forget}} h_{t+1}$$

LSTM

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_{t-1}$$

$$h_t = O_t * \tanh(C_t)$$

- i_{input} : 외부 자극을 얼마나 수용할지

  $$i_t = \mathrm{sigmoid}(W_{hi} h_{t-1} + W_{xi} x_t + b_i)$$

- f_{forget} : 현재 기억을 얼마나 잊어버릴지

  $$f_t = \mathrm{sigmoid}(W_{hf} h_{t-1} + W_{xf} x_t + b_f)$$

- $\tilde{C}_{cell}$: 외부 자극을 받아 변화한 기억

  $$\tilde{C}_t = \tanh(W_{hc} h_{t-1} + W_{xc} x_t + b_c)$$

- O_{out} : 내면 상태를 얼마나 외부로 표출할지

  $$O_t = \mathrm{sigmoid}(W_{hc} h_{t-1} + W_{xc} x_t + b_o)$$

bAbI Tasks

- ✘ 당장 사람같이 말하는 인공지능을 만들 순 없습니다..
  - 일단 쉬운 문제를 먼저 풀고, 차근차근 발전시켜 나가야죠

- ✘ 그래서 페이스북 연구진들이 만든 **20**가지 **Toy tasks**
  - 이것도 못 풀면 인공지능이라고 할 수 없다!

"Towards AI Complete Question Answering: A Set of Prerequisite Toy Tasks" (2015)

bAbI Tasks

Story (1: 1 supporting fact)	Support	Hop 1	Hop 2	Hop 3
Daniel went to the bathroom.		0.00	0.00	0.03
Mary travelled to the hallway.		0.00	0.00	0.00
John went to the bedroom.		0.37	0.02	0.00
John travelled to the bathroom.	yes	0.60	0.98	0.96
Mary went to the office.		0.01	0.00	0.00
Where is John? Answer: bathroom Prediction: bathroom				

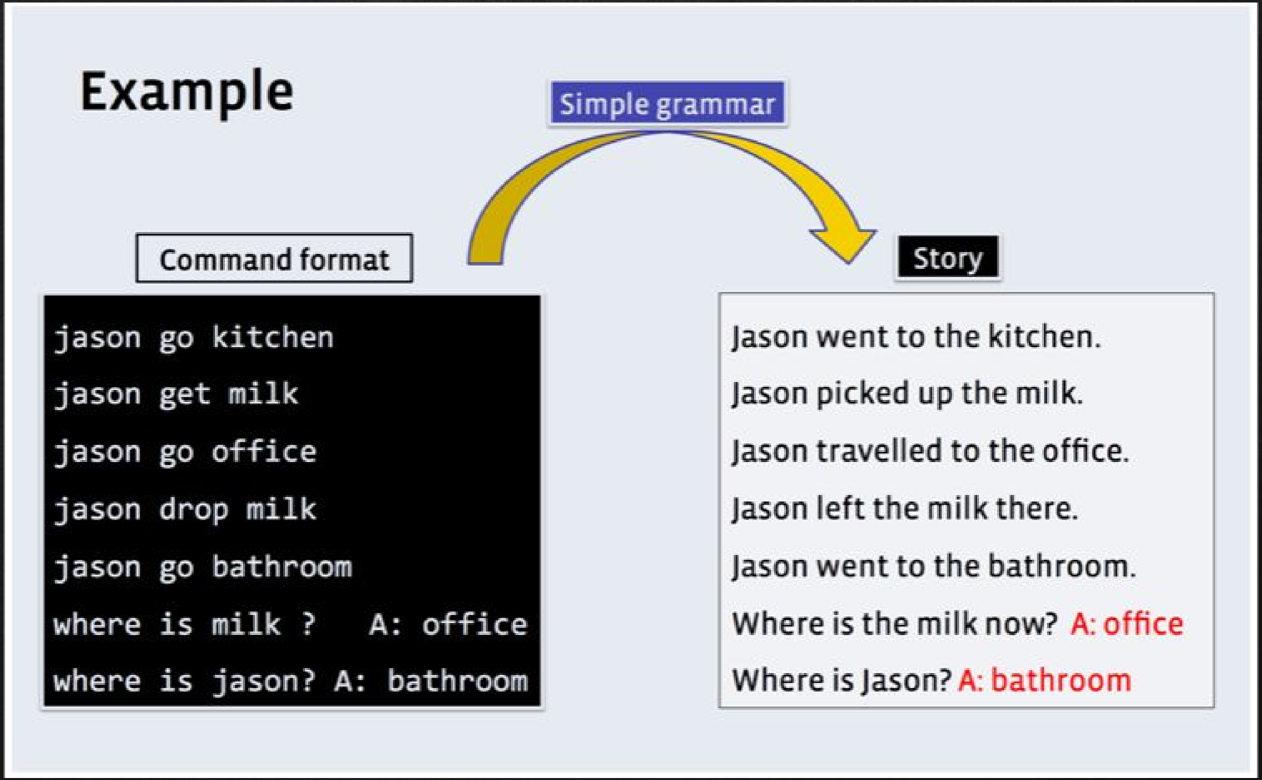
Story (2: 2 supporting facts)	Support	Hop 1	Hop 2	Hop 3
John dropped the milk.		0.06	0.00	0.00
John took the milk there.	yes	0.88	1.00	0.00
Sandra went back to the bathroom.		0.00	0.00	0.00
John moved to the hallway.	yes	0.00	0.00	1.00
Mary went back to the bedroom.		0.00	0.00	0.00
Where is the milk?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

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

Story (18: size reasoning)	Support	Hop 1	Hop 2	Hop 3
The suitcase is bigger than the chest.	yes	0.00	0.88	0.00
The box is bigger than the chocolate.		0.04	0.05	0.10
The chest is bigger than the chocolate.	yes	0.17	0.07	0.90
The chest fits inside the container.		0.00	0.00	0.00
The chest fits inside the box.		0.00	0.00	0.00
Does the suitcase fit in the chocolate? Answer: no Prediction: no				

"Towards AI Complete Question Answering: A Set of Prerequisite Toy Tasks" (2015)

bAbI Tasks

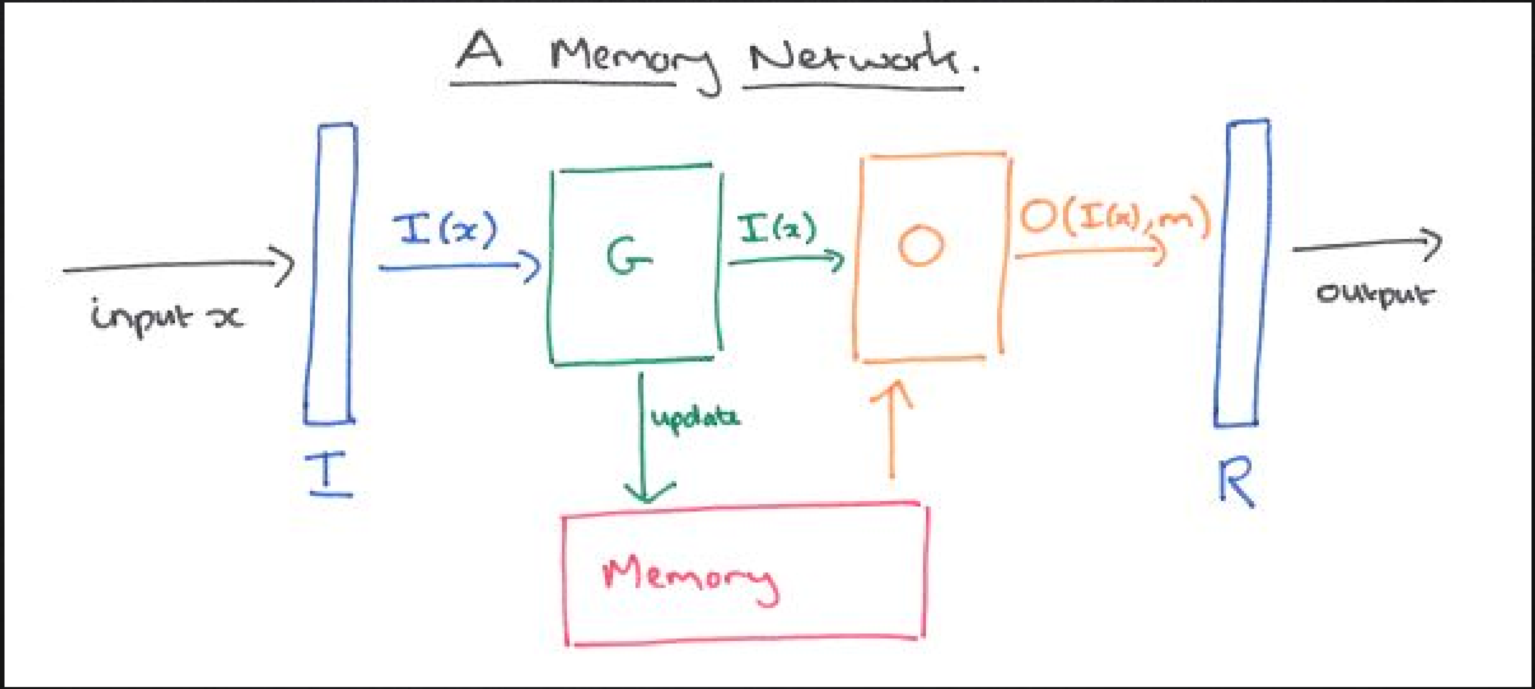


"Towards AI Complete Question Answering: A Set of Prerequisite Toy Tasks" (2015)

External Memory

- 뉴럴넷의 저장공간은 weight parameters
- 매 입력마다 loss에 따라서 갱신됨
- 따라서 이전에 입력받은 정보를 어렴풋이 기억함
  - 정보를 받은 그대로 선명하게 기억하지 못함
- 아예 ~~외장하드~~ 외부 메모리를 만들자!
  - External Memory

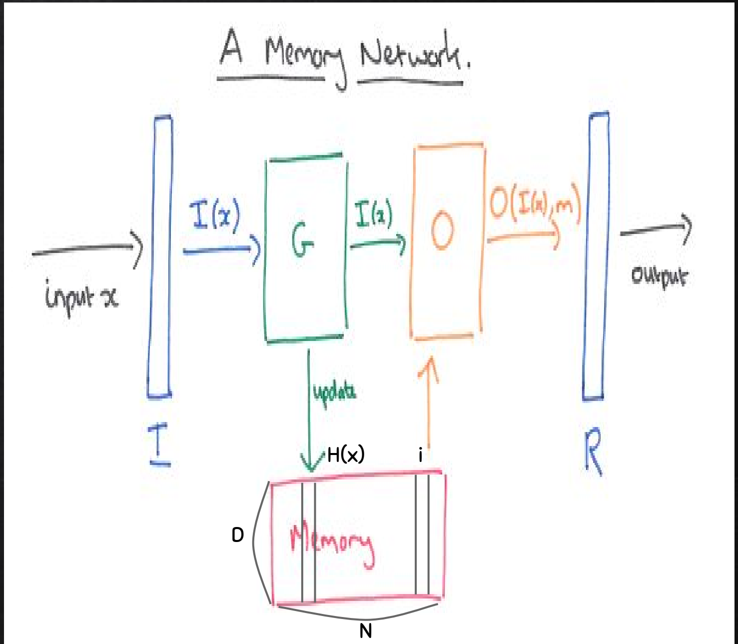
Memory Networks



"Memory Networks" (2014)

Memory Networks

- I (Input feature map)
  - Query => Sparse / Dense feature vector
  - x => I(x)
- G (Generalization)
  - Store given input feature I(X) in index H(x)
  - $H(X)_t = H(X)_{t-1} + 1$
  - $m_{H(x)} = I(x)$
  - (Implementation) m [: , H(x)] = I(x)
- O (Output)
  - Produce output feature from memories with score function
- R (Response)
  - Response sentence = RNN(Output feature)

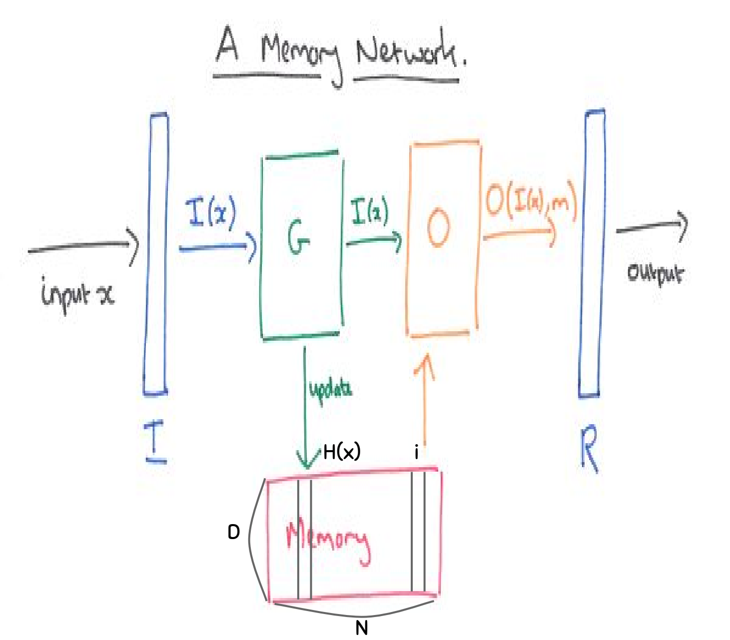


"Memory Networks" (2014)

Memory Networks

Check out more details! => [ICML 2016 Tutorial]

- D: vector dimension / N: # of memory slots
- Vectorization
  - Input sentence (list of integer index) => feature vector
- Memory matrix
  - [D x N]
- Scoring function
  - Relationship between i-th memory <-> query
  - Dot product variant $s(x, y) = \Phi_x(x)^\top U^\top U \Phi_y(y).$
- Take memory with best score
  - Output memory index $i = \text{argmax}_i\ s(x, m_i)$
- Generate Response
  - $h_0 = m_i$
  - Next word = RNN(current word, h)



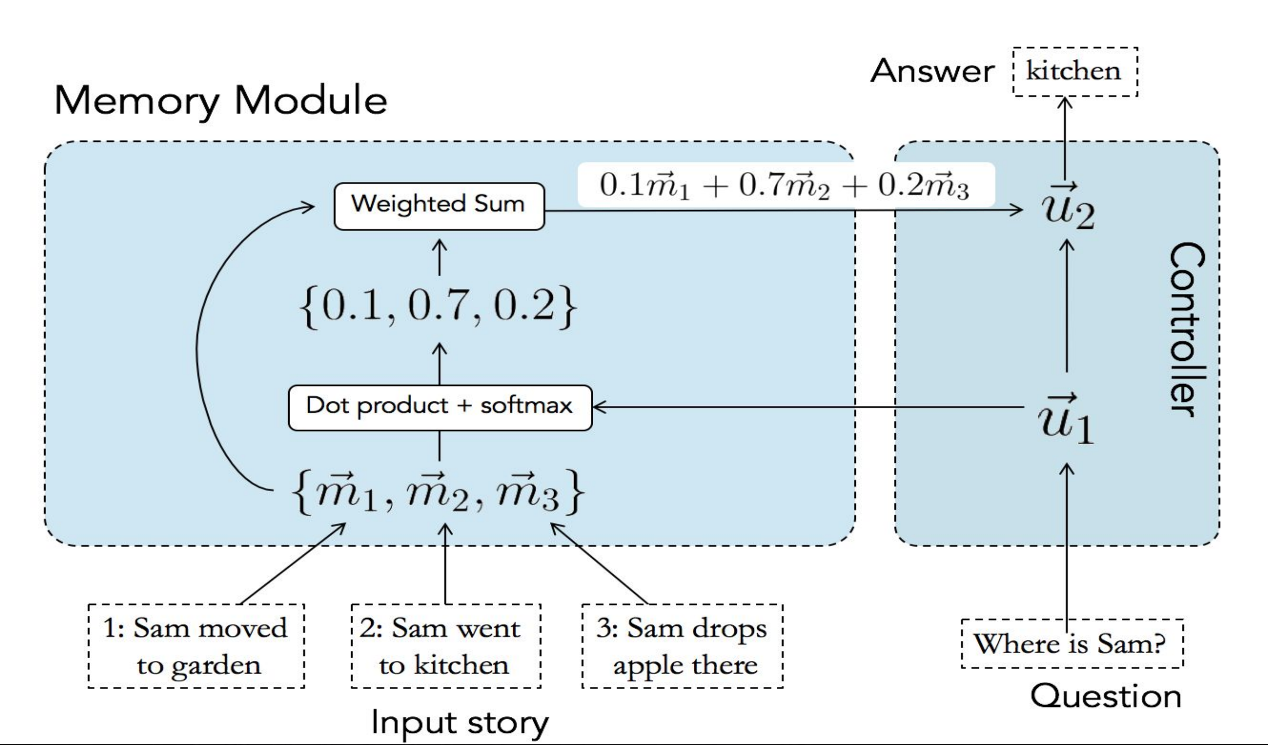
"Memory Networks" (2014)

Memory Networks 의 문제점

- ✘ 학습 과정이 복잡함
  - Question에 답하기 위해 memory에서 어떤 문장에 접근하는지에 대해서도 감독 요구
  - 모든 question 에 대한 '근거 문장 (Supporting facts)' 도 트레이닝 해야 함 => 레이블링 필요
- ✘ 대다수의 데이터는 Question – Answer 쌍으로만 이루어져 있음
  - Question – Answer 쌍만 주어지면 end-to-end 방법으로 학습이 되는 보다 general 한 모델 필요

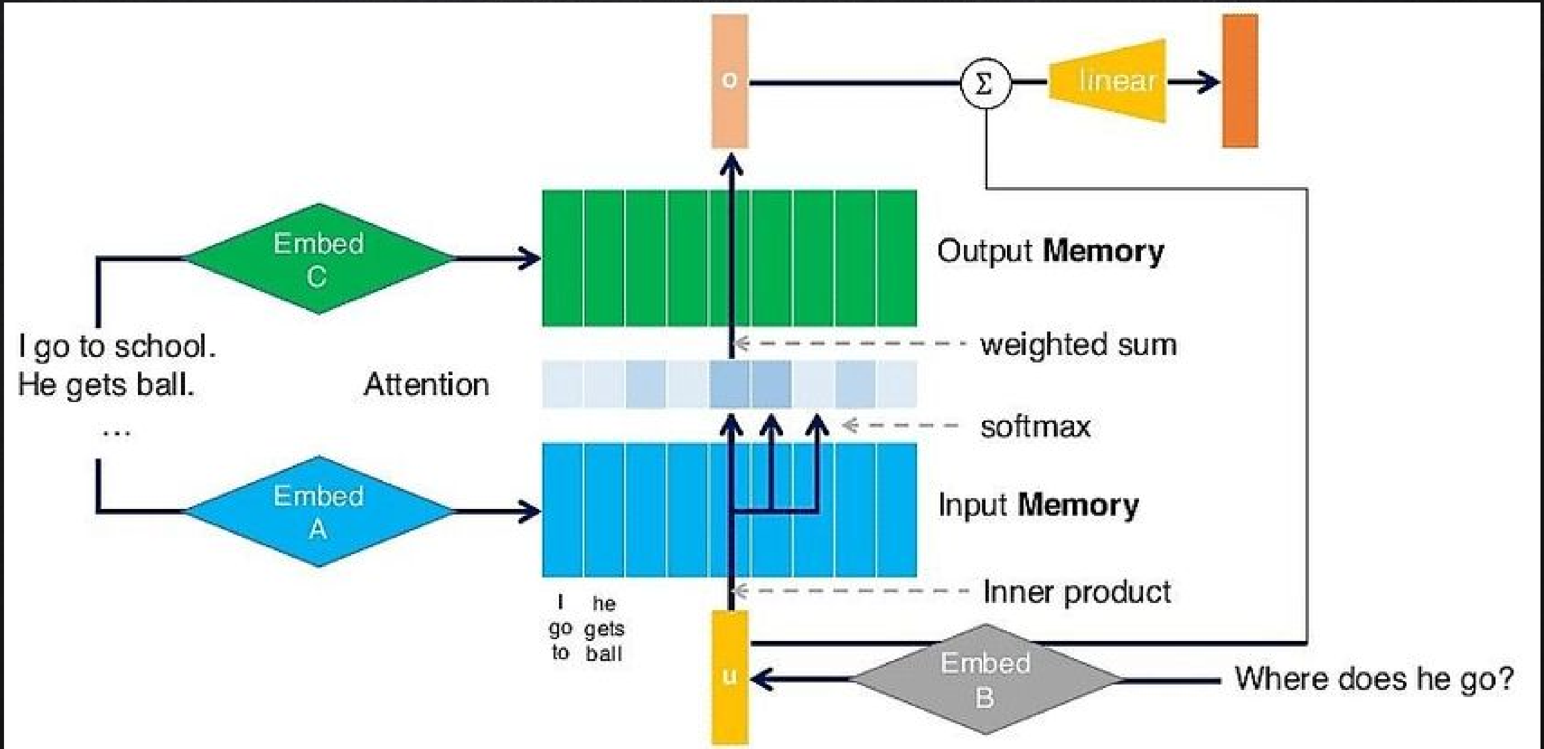
"End-To-End Memory Networks" (2015)

End-To-End Memory Networks



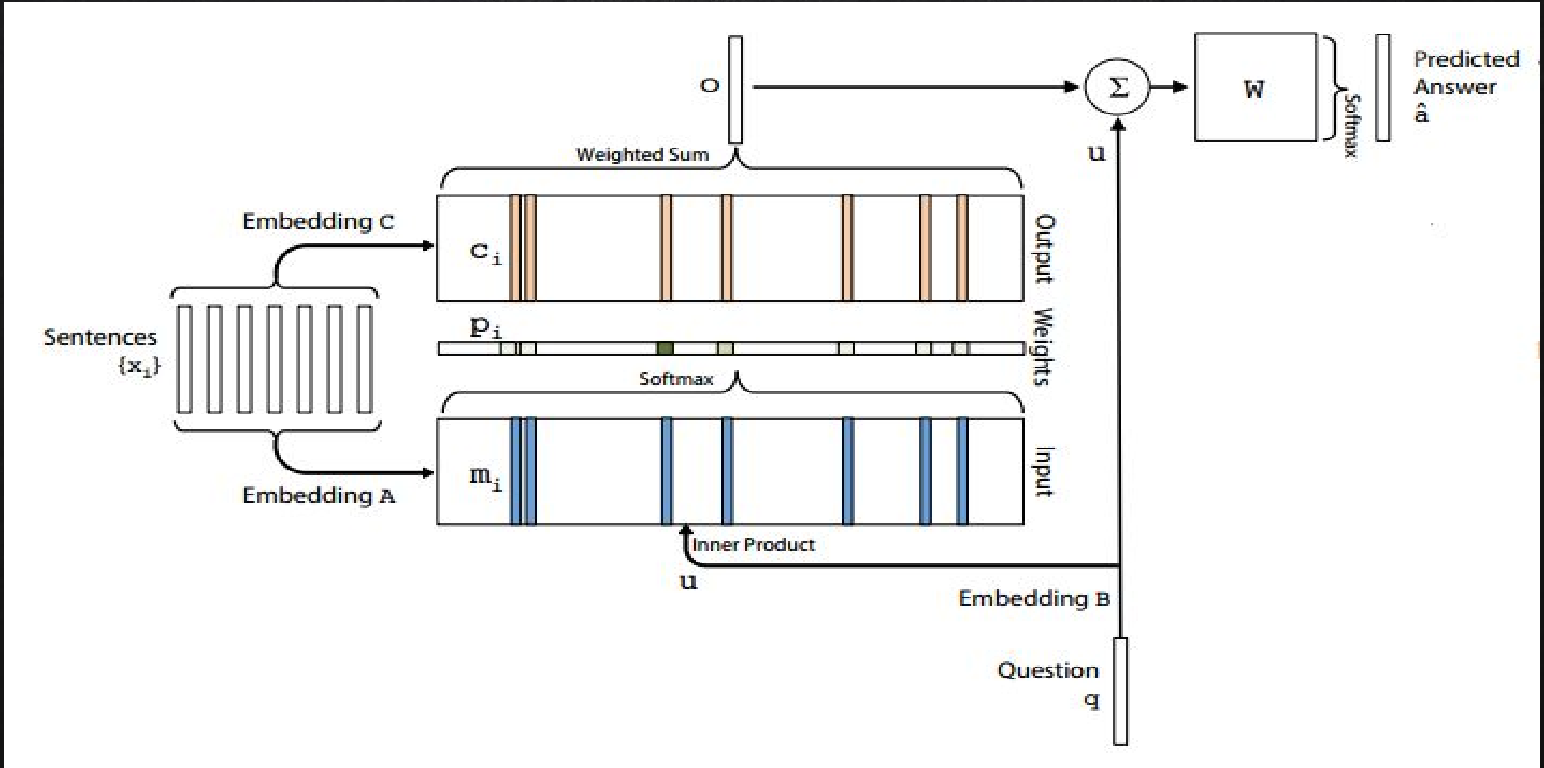
"End-To-End Memory Networks" (2015)

End-To-End Memory Networks



"End-To-End Memory Networks" (2015)

End-To-End Memory Networks



“End-To-End Memory Networks” (2015)

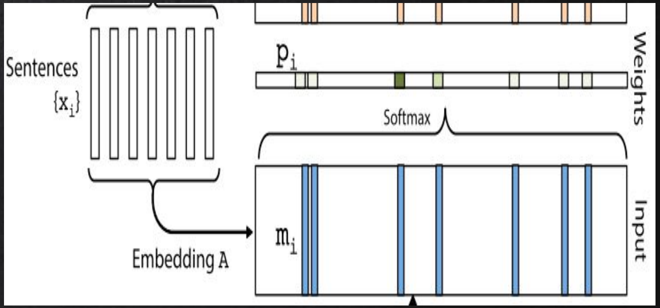
End-To-End Memory Networks

- ✘ Setting
  - Task
    - 지문이 주어지고, 이에 관련된 문제에 답하기
  - 지문: $\{x_i\}$
    - n개의 문장 $x_1 \sim x_n$
    - X_i: i번째 문장
    - 문장은 단어들의 리스트
  - 문제: 문장 q
  - 답: 문장 a
  - Vocabulary
    - 총 단어 갯수: d
    - 모든 '지문', '문제', '답' 들은 Vocabulary 공유
- ✘ Training
  - 모델이 $x_1 \sim x_n$ 의 지문과, 문제 q를 입력받고 출력한 답과 정답 a가 같도록 비교 및 업데이트
  - Word-level Cross Entropy

"End-To-End Memory Networks" (2015)

End-To-End Memory Networks

- Input Memory Representation
  - Embedding matrix A
    - d X V 차원의 행렬
    - 단어 => d-차원 벡터
    - 문장 => d-차원 벡터의 리스트



  - 문장 벡터 m_i
    - Embedding_A(x_i) = m_i
    - Bag-of-Words
      - 워드벡터들을 합한 것이 문장 벡터
    - Positional Encoding (PE)
      - 지금 단어가 문장에서 몇 번째인지에 대한 정보를 추가
      - 워드벡터들을 weighted sum 한 것이 문장 벡터
      - [YerevaNN's slide]

$$m_i = \sum_j Ax_{ij}$$

$$m_i = \sum_j l_j \cdot Ax_{ij}$$

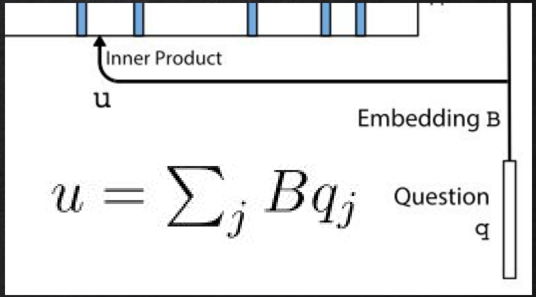
$$l_{kj} = (1 - j/J) - (k/d)(1 - 2j/J)$$

J: 문장을 구성하는 단어의 수

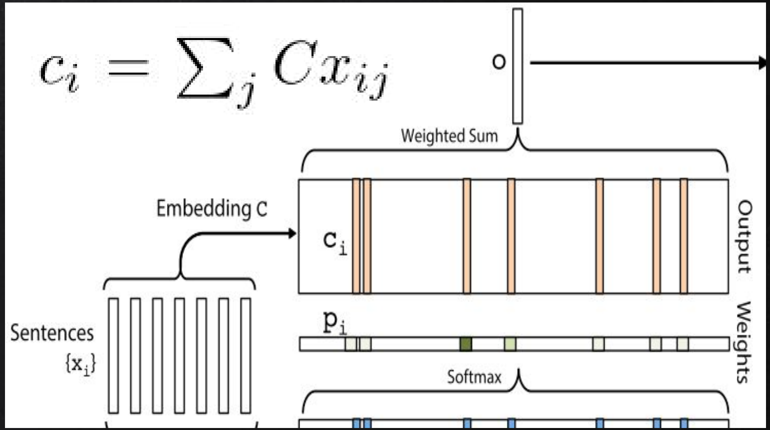
"End-To-End Memory Networks" (2015)

End-To-End Memory Networks

- Query Representation
  - Embedding matrix B
    - d X V 차원의 행렬
    - 문장 => d-차원 벡터의 리스트
  - Embedding_B (q) => u



- Output Memory Representation
  - Embedding matrix C
    - d X V 차원의 행렬
    - 문장 => d-차원 벡터의 리스트
  - Embedding_C (x_i) = c_i



"End-To-End Memory Networks" (2015)

End-To-End Memory Networks

- Input memory m_i – Query representation u
  - 지문 중 어떤 문장이 문제와 가장 연관이 있을까?
  - Scoring function: dot product
  - Normalized weight : p_i

$$p_i = \text{Softmax}(u^T m_i)$$

- Output representation o
  - 출력을 위해 지문 전체를 한 벡터로 압축하기
  - 위에서 구한 p_i를 가중치로 하는 weighted sum

$$o = \sum_i p_i c_i$$

- Final output
  - 출력을 위한 마지막 projection W
  - 차원: V x d (A, B, C 와 같음)
  - a^: V 차원 벡터
  - 이것을 one-hot encoded 정답 단어와 비교
  - Cross-Entropy

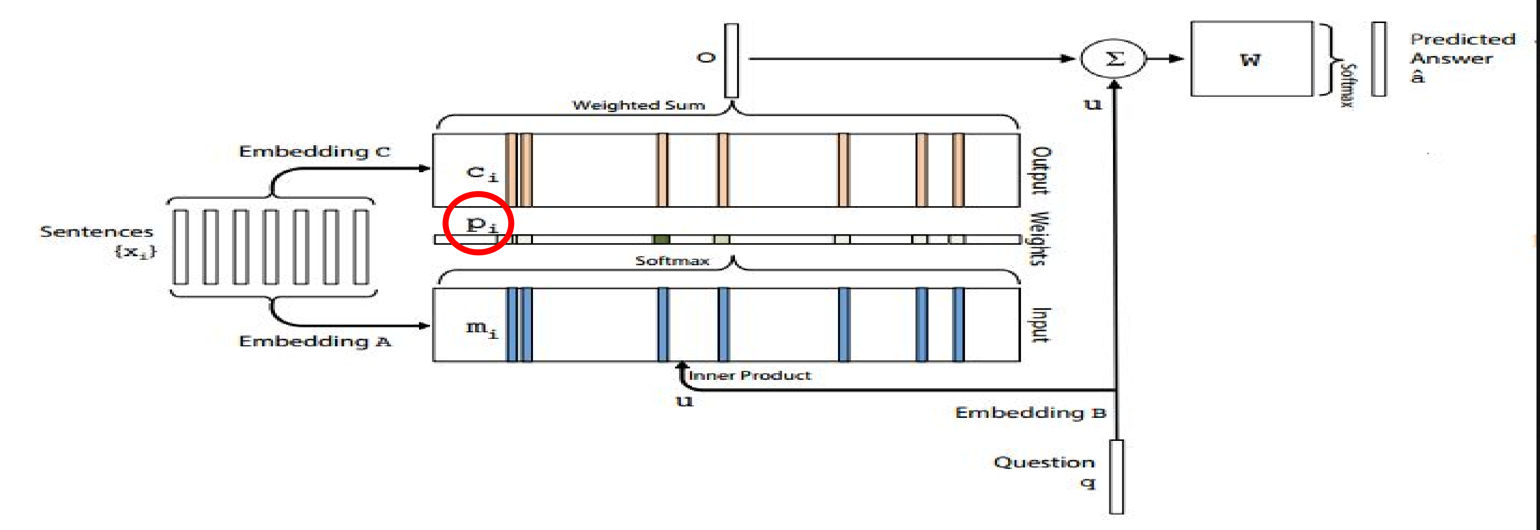
$$\hat{a} = \text{Softmax}(W(o + u))$$

"End-To-End Memory Networks" (2015)

End-To-End Memory Networks

- Input memory m_i – Query representation u
  - 지문 중 어떤 문장이 문제와 가장 연관이 있을까?
  - Scoring function: dot product
  - Normalized weight : p_i

$$p_i = \text{Softmax}(u^T m_i)$$

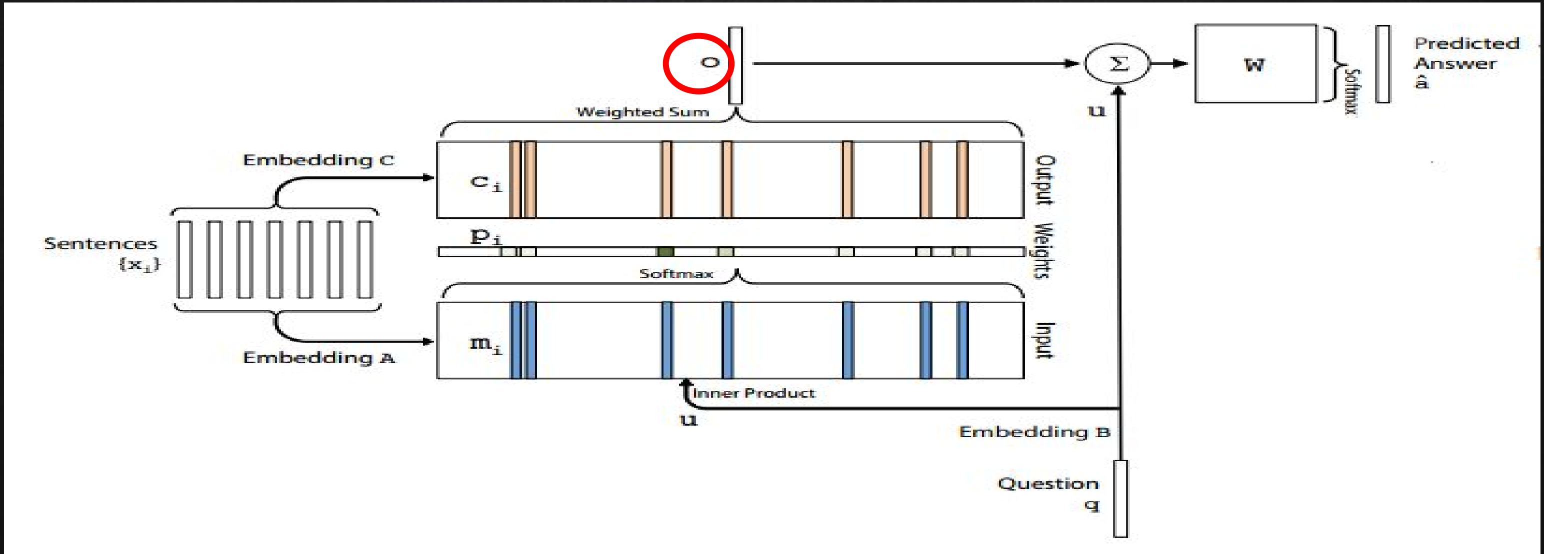


“End-To-End Memory Networks” (2015)

End-To-End Memory Networks

- Output representation o
  - 출력을 위해 지문 전체를 한 벡터로 압축하기
  - 위에서 구한 p_i를 가중치로 하는 weighted sum

$$o = \sum_i p_i c_i$$

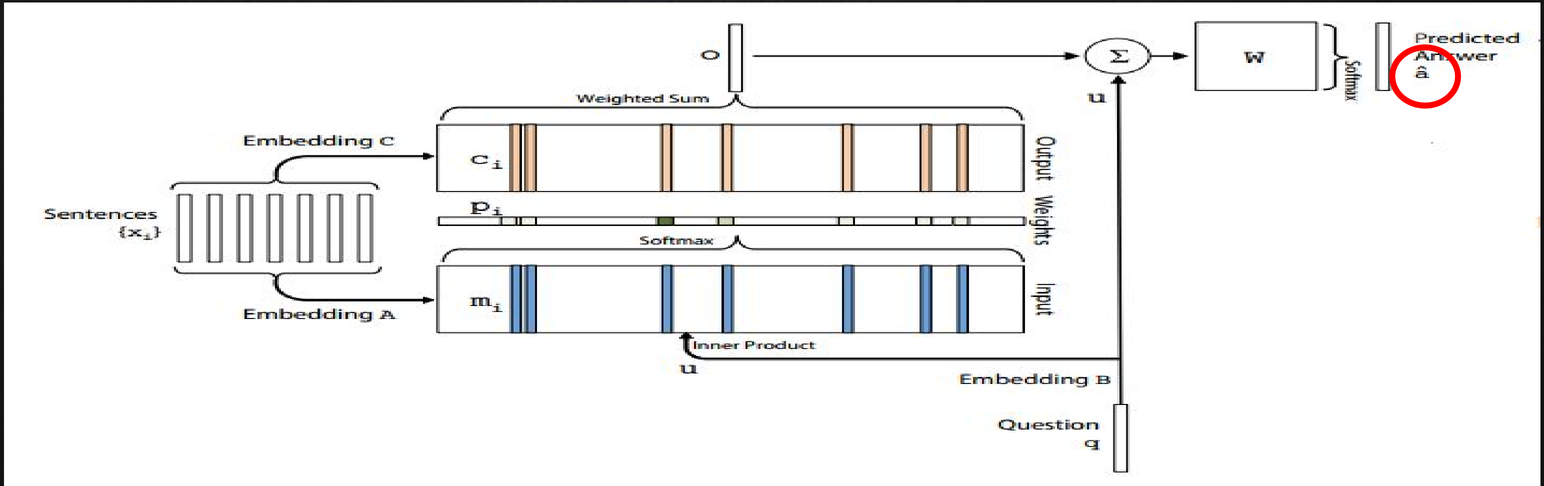


"End-To-End Memory Networks" (2015)

End-To-End Memory Networks

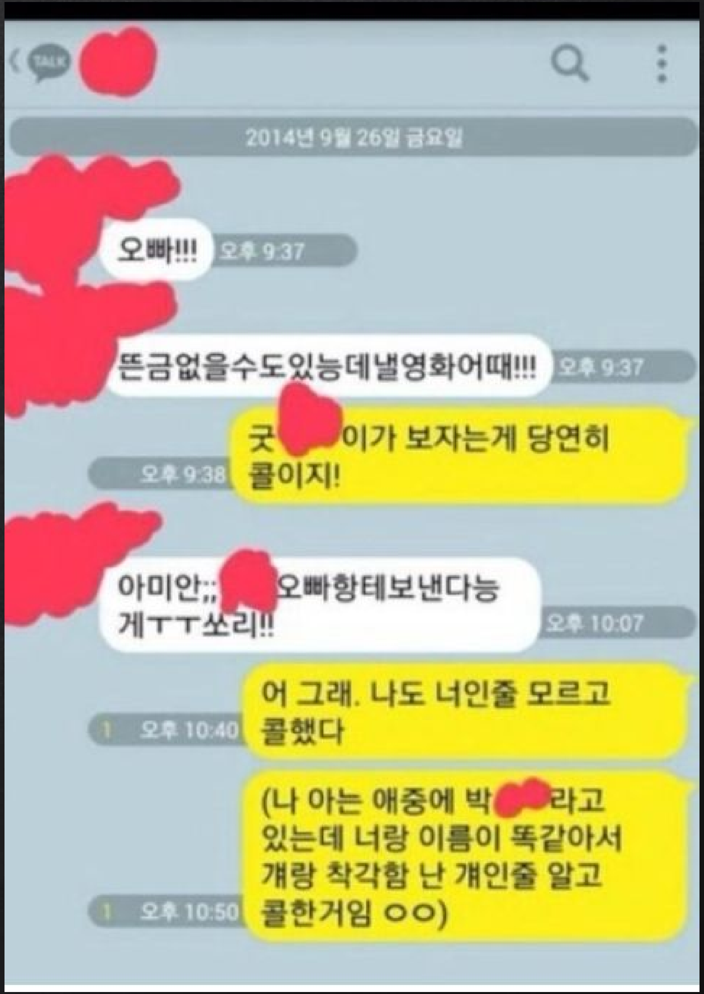
- Final output
  - 출력을 위한 마지막 projection W
  - 차원: V x d (A, B, C 와 같음)
  - a^: V 차원 벡터
  - 이것을 one-hot encoded 정답 단어와 비교
  - Cross-Entropy

$$\hat{a} = \text{Softmax}(W(o + u))$$



"End-To-End Memory Networks" (2015)

✘ 왜 (챗봇은) 말실수를 할까…


2014년 9월 26일 금요일
오빠!!!
오후 9:37
뜬금없을수도있능데낼영화어때!!!
오후 9:37
굿 이가 보자는게 당연히 콜이지!
오후 9:38
아미안;; 오빠항테보낸다능 게ㅜㅜ쏘리!!
오후 10:07
어 그래. 나도 너인줄 모르고 콜했다
1 오후 10:40
(나 아는 애중에 박 라고 있는데 너랑 이름이 똑같아서 걔랑 착각함 난 걔인줄 알고 콜한거임 ㅇㅇ)
1 오후 10:50

- ✘ 생각을 충분히 하지 않아서…

Multi-Hop Attention

- 여러 번 생각하지 않으면 풀 수 없는 문제도 많습니다..

Story (1: 1 supporting fact)	Support	Hop 1	Hop 2	Hop 3
Daniel went to the bathroom.		0.00	0.00	0.03
Mary travelled to the hallway.		0.00	0.00	0.00
John went to the bedroom.		0.37	0.02	0.00
John travelled to the bathroom.	yes	0.60	0.98	0.96
Mary went to the office.		0.01	0.00	0.00
Where is John? Answer: bathroom Prediction: bathroom				

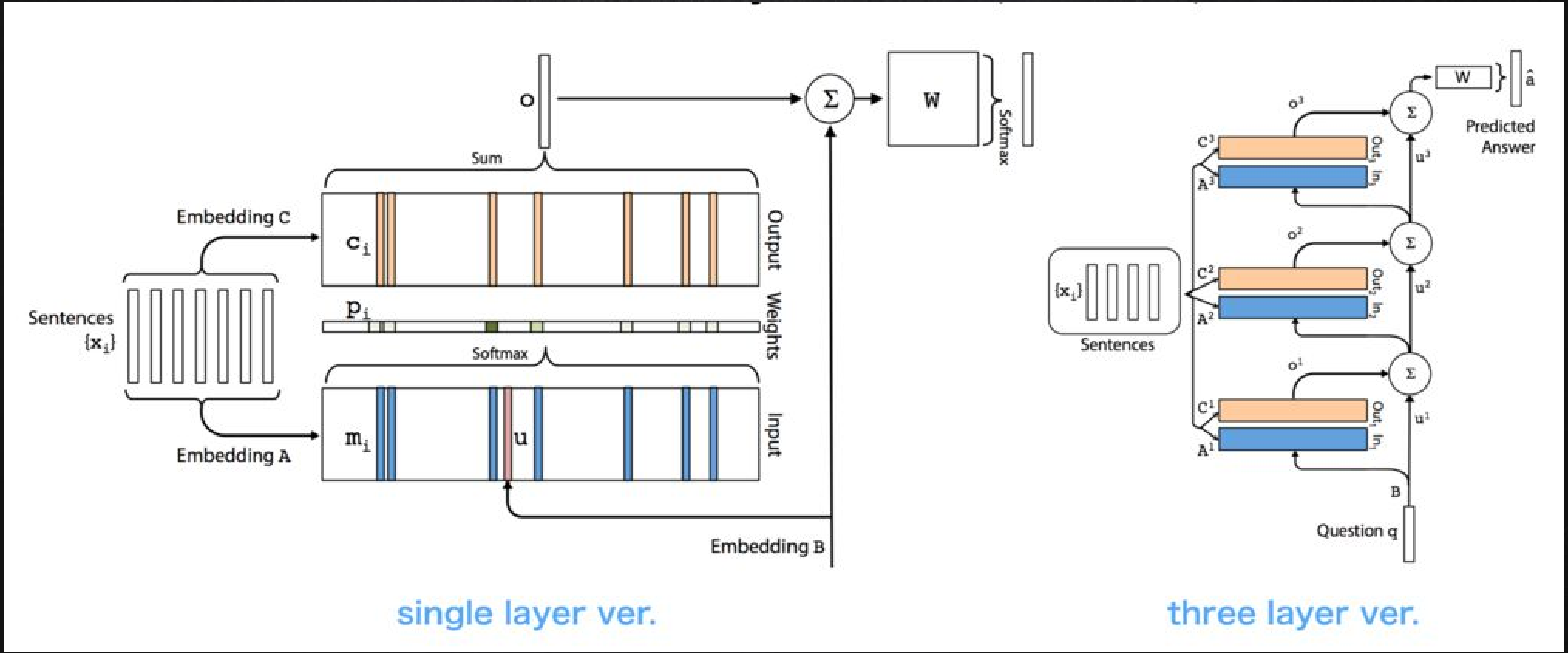
Story (2: 2 supporting facts)	Support	Hop 1	Hop 2	Hop 3
John dropped the milk.		0.06	0.00	0.00
John took the milk there.	yes	0.88	1.00	0.00
Sandra went back to the bathroom.		0.00	0.00	0.00
John moved to the hallway.	yes	0.00	0.00	1.00
Mary went back to the bedroom.		0.00	0.00	0.00
Where is the milk?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

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

Story (18: size reasoning)	Support	Hop 1	Hop 2	Hop 3
The suitcase is bigger than the chest.	yes	0.00	0.88	0.00
The box is bigger than the chocolate.		0.04	0.05	0.10
The chest is bigger than the chocolate.	yes	0.17	0.07	0.90
The chest fits inside the container.		0.00	0.00	0.00
The chest fits inside the box.		0.00	0.00	0.00
Does the suitcase fit in the chocolate? Answer: no Prediction: no				

"End-To-End Memory Networks" (2015)

Multi-Hop Attention



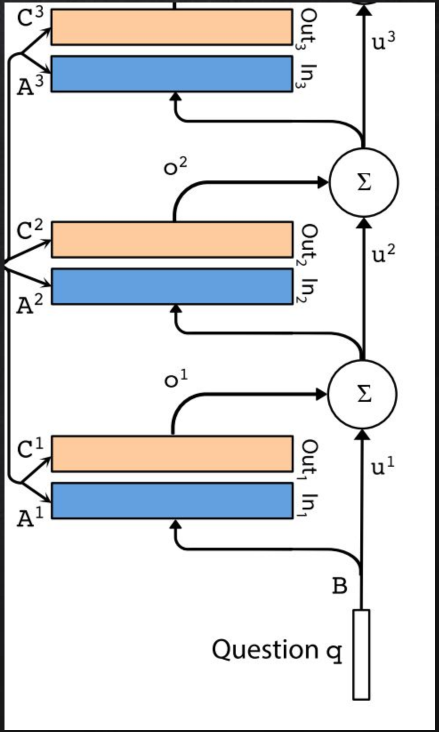
"End-To-End Memory Networks" (2015)

Multi-Hop Attention

- ✘ 딥러닝은 역시 깊이 쌓아야 제맛!
- ✘ Residual Connection
  - Next query = previous query + output

$$u^{k+1} = u^k + o^k$$

- ✘ 그런데 매 Layer 마다 V x d 차원 행렬이 3개씩… ㅠㅠ



“End-To-End Memory Networks” (2015)

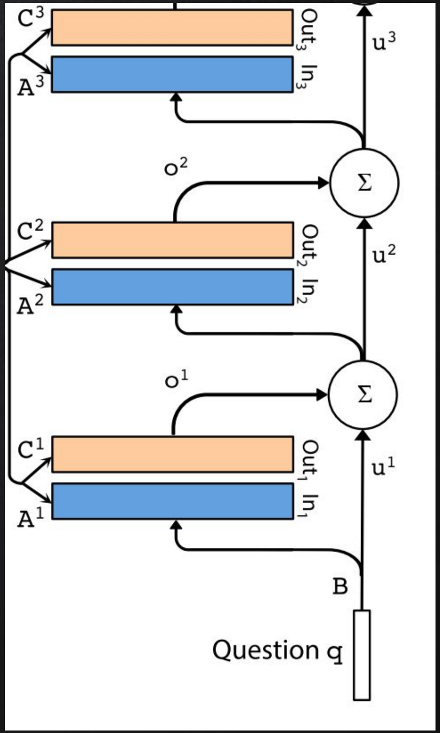
Multi-Hop Attention

- Tying embedding weight
  - Adjacent
    - 이전 레이어의 C를 현재 A와 공유
      - $A^{k+1} = C^k$
    - 출력 Weight는 마지막 C를 한번 더 사용
      - $W^T = C^K$
  - Layer-wise (RNN처럼)
    - Input embedding, Output embedding 각각 모든 레이어에서 공유

$$A^1 = A^2 = \ldots = A^K \text{ and } C^1 = C^2 = \ldots = C^K$$

    - Extra linear mapping H
      - d x d 차원
      - 실험 결과 성능 향상

$$u^{k+1} = Hu^k + o^k$$



"End-To-End Memory Networks" (2015)

End-To-End Memory Networks

- Temporal Encoding
  - 사건의 순서를 알아야 대답할 수 있는 질문들이 있음
  - Sam 이 Kitchen 에 간 “이후” bedroom 으로 이동
  - 만약 이 두 문장의 순서가 뒤바뀌면 답도 달라짐
  - 문장들의 순서도 인코딩

$$m_i = \sum_j Ax_{ij} + T_A(i)$$

$$c_i = \sum_j Cx_{ij} + T_C(i)$$

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

  - T_A, T_C는 학습 대상
  - Learning time variance by injecting Random Noise (RN)
    - Regularization 을 위해 Training 시 T_A 에 10% 의 empty memory 추가

- Linear Start (LS)
  - 초기 loss가 감소할 때까지 마지막 Softmax를 제외한 Softmax를 모두 제거하고 학습

“End-To-End Memory Networks” (2015)

End-To-End Memory Networks

- Results
  - Memory Network 에 근접
  - PE 가 Bag-of-Words 보다 나음
  - Joint training 효과 있음
  - Hop 많을수록 향상
  - LS 가 local minima 피하게 함
    - Task 16

	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

"End-To-End Memory Networks" (2015)

Key-Value Memory Networks

- ✘ Large Scale QA
  - ○ 모든 지식을 책으로 읽기보다는 미리 잘 정리된 표를 참고하자!
  - ○ Question Answering 문제를 풀 때
    - ■ Raw Text 보다는
    - ■ 미리 잘 정리된 Knowledge Base (KB) 의 도움을 받자!

- ✘ 하지만 Knowledge Base 도 방대하다..
  - ○ 중요한 문서만 골라 읽자!
    - ■ Key hashing
      - ● 질문과 겹치는 단어가 있는 문서들만 자세히 살펴보자
  - ○ 어떻게?
    - ■ End-To-End Memory Networks

Doc: Wikipedia Article for Blade Runner (partially shown)

Blade Runner is a 1982 American neo-noir dystopian science fiction film directed by Ridley Scott and starring Harrison Ford, Rutger Hauer, Sean Young, and Edward James Olmos. The screenplay, written by Hampton Fancher and David Peoples, is a modified film adaptation of the 1968 novel "Do Androids Dream of Electric Sheep?" by Philip K. Dick. The film depicts a dystopian Los Angeles in November 2019 in which genetically engineered replicants, which are visually indistinguishable from adult humans, are manufactured by the powerful Tyrell Corporation as well as by other "mega-corporations" around the world. Their use on Earth is banned and replicants are exclusively used for dangerous, menial, or leisure work on off-world colonies. Replicants who defy the ban and return to Earth are hunted down and "retired" by special police operatives known as "Blade Runners". . . .

KB entries for Blade Runner (subset)

Blade Runner *directed_by* Ridley Scott
Blade Runner *written_by* Philip K. Dick, Hampton Fancher
Blade Runner *starred_actors* Harrison Ford, Sean Young, . . .
Blade Runner *release_year* 1982
Blade Runner *has_tags* dystopian, noir, police, androids, . . .

IE entries for Blade Runner (subset)

Blade Runner, Ridley Scott *directed* dystopian, science fiction, film
Hampton Fancher *written* Blade Runner
Blade Runner *starred* Harrison Ford, Rutger Hauer, Sean Young. . .
Blade Runner *labelled* 1982 neo noir
special police, Blade *retired* Blade Runner
Blade Runner, special police *known* Blade

"Key-Value Memory Networks for Directly Reading Documents" (2016)

WIKIMOVIES

Questions for Blade Runner (subset)

Ridley Scott directed which films?
What year was the movie Blade Runner released?
Who is the writer of the film Blade Runner?
Which films can be described by dystopian?
Which movies was Philip K. Dick the writer of?
Can you describe movie Blade Runner in a few words?

Doc: Wikipedia Article for Blade Runner (partially shown)

Blade Runner is a 1982 American neo-noir dystopian science fiction film directed by Ridley Scott and starring Harrison Ford, Rutger Hauer, Sean Young, and Edward James Olmos. The screenplay, written by Hampton Fancher and David Peoples, is a modified film adaptation of the 1968 novel "Do Androids Dream of Electric Sheep?" by Philip K. Dick. The film depicts a dystopian Los Angeles in November 2019 in which genetically engineered replicants, which are visually indistinguishable from adult humans, are manufactured by the powerful Tyrell Corporation as well as by other "mega-corporations" around the world. Their use on Earth is banned and replicants are exclusively used for dangerous, menial, or leisure work on off-world colonies. Replicants who defy the ban and return to Earth are hunted down and "retired" by special police operatives known as "Blade Runners". . . .

KB entries for Blade Runner (subset)

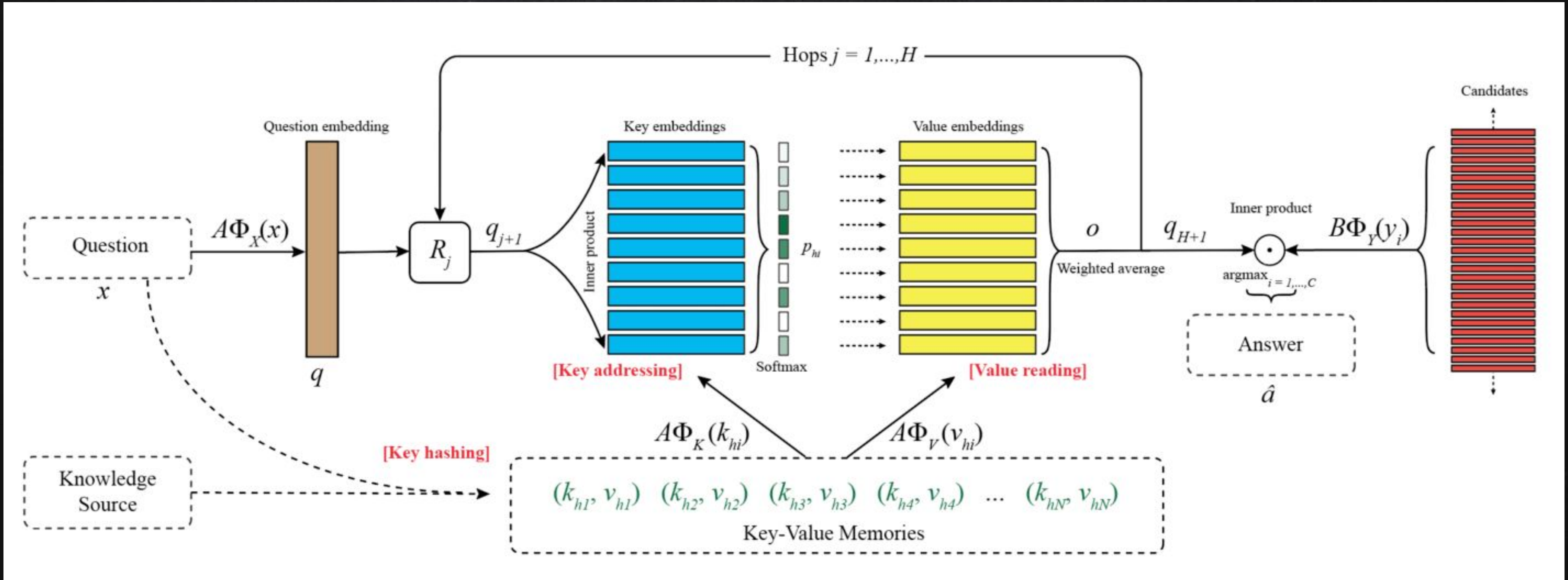
Blade Runner *directed_by* Ridley Scott
Blade Runner *written_by* Philip K. Dick, Hampton Fancher
Blade Runner *starred_actors* Harrison Ford, Sean Young, . . .
Blade Runner *release_year* 1982
Blade Runner *has_tags* dystopian, noir, police, androids, . . .

IE entries for Blade Runner (subset)

Blade Runner, Ridley Scott *directed* dystopian, science fiction, film
Hampton Fancher *written* Blade Runner
Blade Runner *starred* Harrison Ford, Rutger Hauer, Sean Young. . .
Blade Runner *labelled* 1982 neo noir
special police, Blade *retired* Blade Runner
Blade Runner, special police *known* Blade

"Key-Value Memory Networks for Directly Reading Documents" (2016)

Key-Value Memory Networks



"Key-Value Memory Networks for Directly Reading Documents" (2016)

Key-Value Memory Networks

영화 관련 질문
100,00개 이상의 질문

Method	KB	IE	Doc
(Bordes *et al.*, 2014) QA system	93.5	56.5	N/A
Supervised Embeddings	54.4	54.4	54.4
Memory Network	78.5	63.4	69.9
Key-Value Memory Network	**93.9**	**68.3**	**76.2**

Table 2: Test results (% hits@1) on WikiMovies, comparing human-annotated KB (KB), information extraction-based KB (IE), and directly reading Wikipedia documents (Doc).

위키피디아 모든 주제
1,000 여개 질문

Method	MAP	MRR
Word Cnt	0.4891	0.4924
Wgt Word Cnt	0.5099	0.5132
2-gram CNN (Yang *et al.*, 2015)	0.6520	0.6652
AP-CNN (Santos *et al.*, 2016)	0.6886	0.6957
Attentive LSTM (Miao *et al.*, 2015)	0.6886	0.7069
Attentive CNN (Yin and Schütze, 2015)	0.6921	0.7108
L.D.C. (Wang *et al.*, 2016)	0.7058	0.7226
Memory Network	0.5170	0.5236
Key-Value Memory Network	**0.7069**	**0.7265**

Table 6: Test results on WikiQA.

"Key-Value Memory Networks for Directly Reading Documents" (2016)

Dynamic Memory Networks

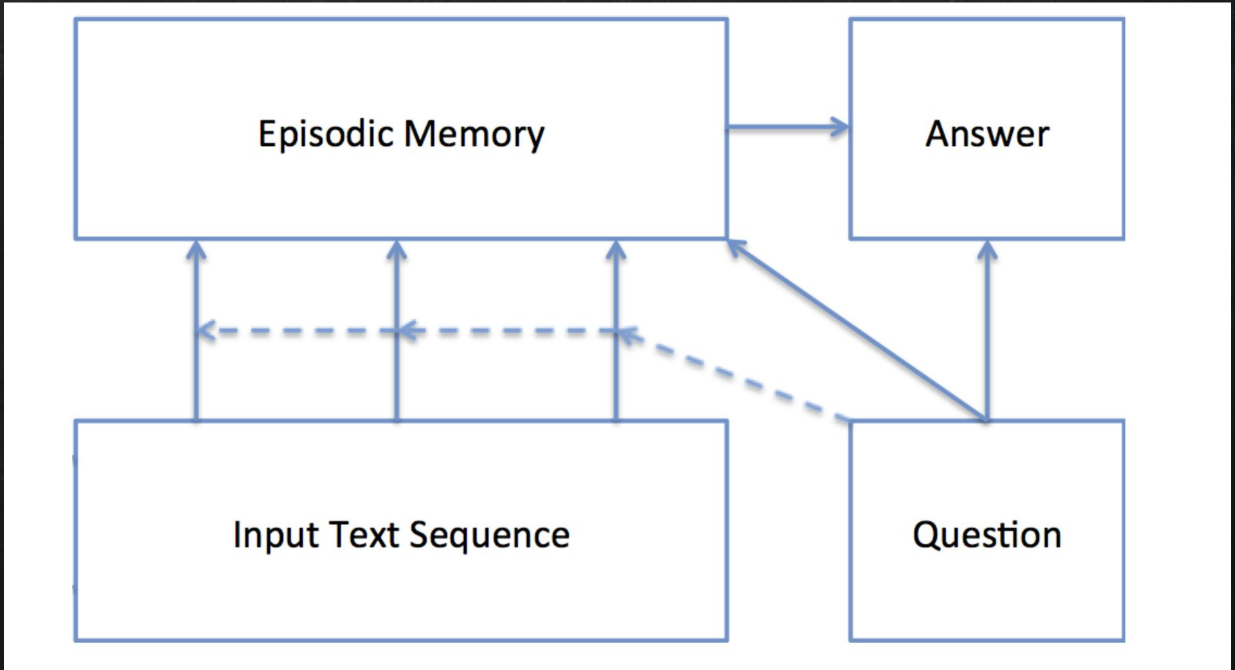
- 사실 대부분의 NLP 문제는 QA 문제와 같다
  - 번역
    - Q: "이 문장을 영어로 번역하면 어떻게 되는가?"
  - Sequence Labeling (POS-tagging, NER, etc.)
    - Q: "이 문장에서 고유명사는 어떤 것들이 있는가?"

I: Jane has a baby in Dresden.
Q: What are the named entities?
A: Jane - person, Dresden - location
I: Jane has a baby in Dresden.
Q: What are the POS tags?
A: NNP VBZ DT NN IN NNP .
I: I think this model is incredible
Q: In French?
A: Je pense que ce modèle est incroyable.

- 그럼 QA 문제만 잘 풀면 되는 것 아닌가?
  - QA 잘 푸는 End-To-End Memory Networks 를 좀 더 발전시켜보자!
    - GRU 3개 + Gating

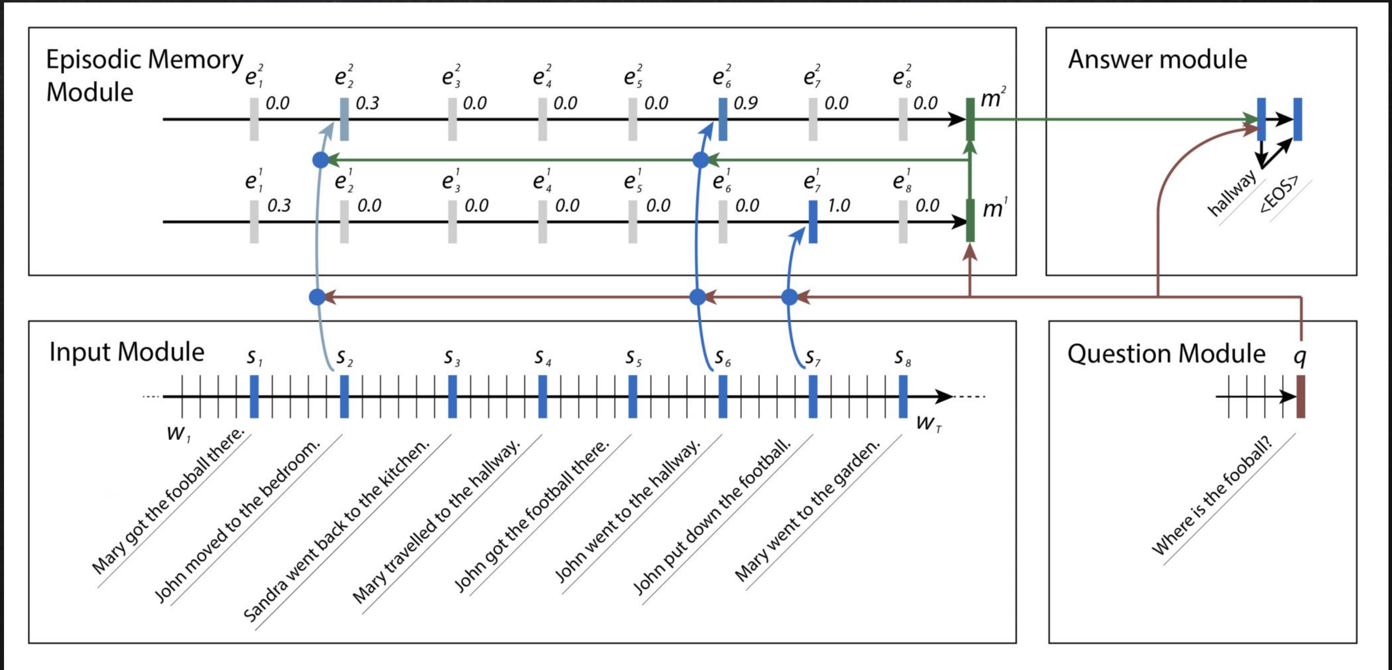
"Ask Me Anything: Dynamic Memory Networks for Natural Language Processing" (2015)

Dynamic Memory Networks



"Ask Me Anything: Dynamic Memory Networks for Natural Language Processing" (2015)

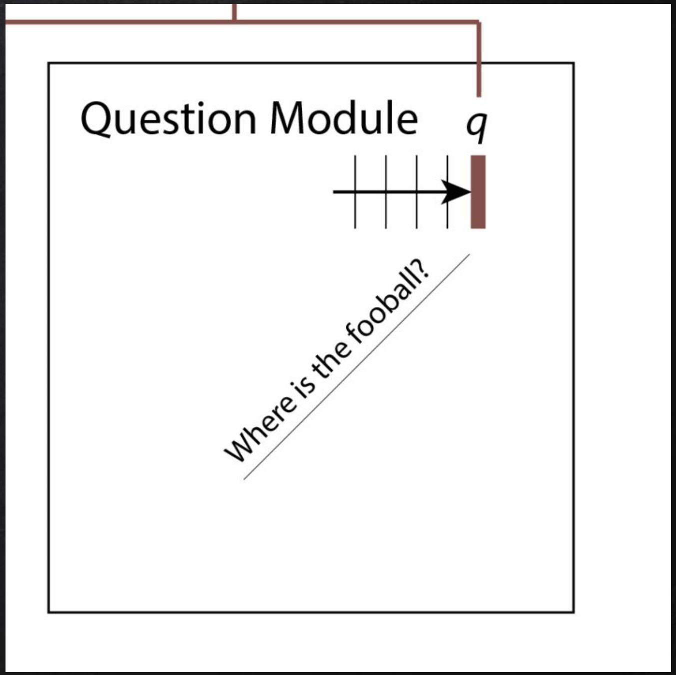
Dynamic Memory Networks



"Ask Me Anything: Dynamic Memory Networks for Natural Language Processing" (2015)

Dynamic Memory Networks

- Question Encoding
  - GRU로 질문의 각 단어를 입력으로 받음
  - 마지막 벡터가 질문의 hidden representation



"Ask Me Anything: Dynamic Memory Networks for Natural Language Processing" (2015)

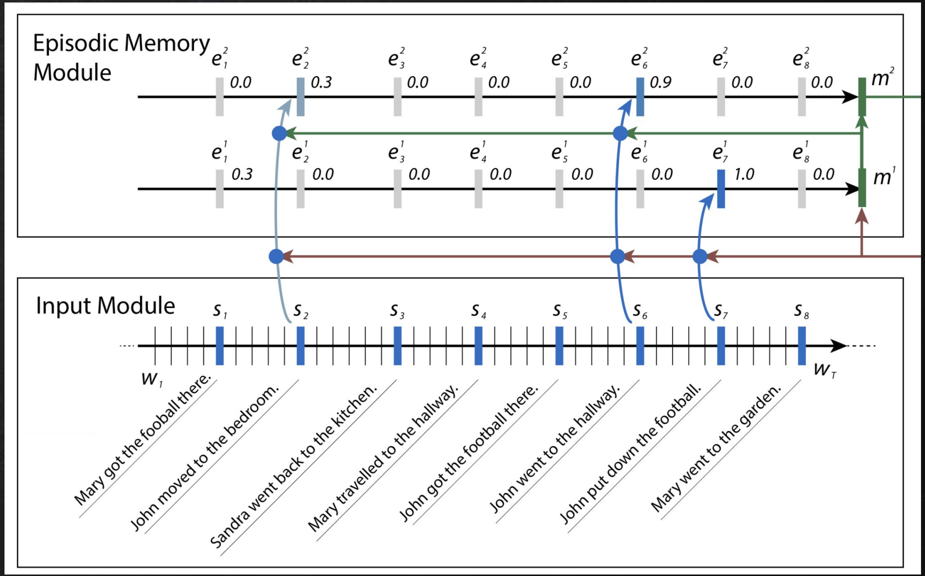
Dynamic Memory Networks

- Episodic Memory Module
  - e: 각 문장 (episode) 의 representation
    - Word-level GRU + Gating

$$
\begin{aligned}
h_t^i &= g_t^i GRU(c_t, h_{t-1}^i) + (1 - g_t^i) h_{t-1}^i \\
e^i &= h_{T_C}^i
\end{aligned}
$$

    - Gating 은 2-layer NN 의 출력
  - m: 지문 전체의 representation
    - GRU

$$
m^i = GRU(e^i, m^{i-1})
$$



"Ask Me Anything: Dynamic Memory Networks for Natural Language Processing" (2015)

Dynamic Memory Networks

- 지문의 문장 (episode) 인코딩 시 Word-level GRU Gating

$$\begin{aligned} h_t^i &= g_t^i GRU(c_t, h_{t-1}^i) + (1 - g_t^i) h_{t-1}^i \\ e^i &= h_{T_C}^i \end{aligned}$$

1) Similarity Score

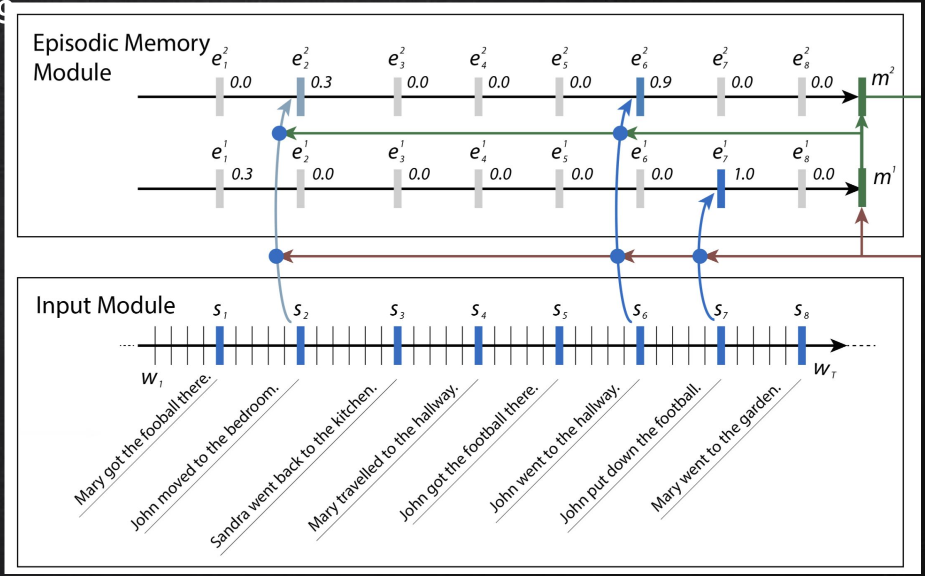
$$z(c, m, q) = \left[c, m, q, c \circ q, c \circ m, |c - q|, |c - m|, c^T W^{(b)} q, c^T W^{(b)} m \right]$$

2) 2-layer NN

$$G(c, m, q) = \sigma \left(W^{(2)} \tanh \left(W^{(1)} z(c, m, q) + b^{(1)} \right) + b^{(2)} \right)$$

3) Gating

$$g_t^i = G(c_t, m^{i-1}, q)$$



- 그런데 지문의 문장을 e로 인코딩할 때 GRU 대신 softmax 를 쓰니까 더 좋았다…

$$e^i = \sum_{t=1}^{T} \text{softmax}(g_t^i) c_t$$

"Ask Me Anything: Dynamic Memory Networks for Natural Language Processing" (2015)

Dynamic Memory Networks

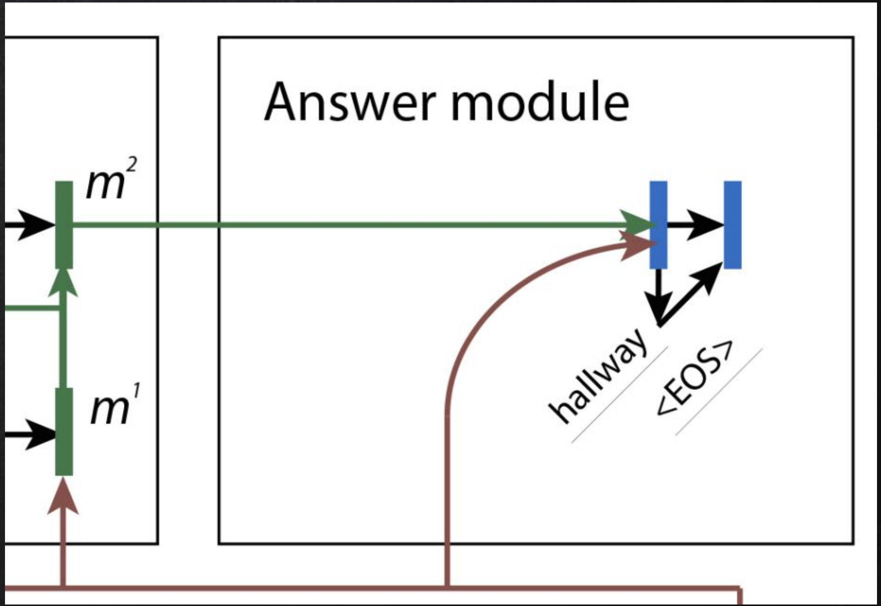
- Answer Module
  - Word-GRU
  - 이전 단어 y_{t-1}, 질문 q, 이전 hidden state

$$a_t = GRU([y_{t-1}, q], a_{t-1})$$

$$y_t = \text{softmax}(W^{(a)} a_t)$$

  - initial hidden state: 마지막 m

$$a_0 = m^{T_M}$$



"Ask Me Anything: Dynamic Memory Networks for Natural Language Processing" (2015)

Dynamic Memory Networks

- Result
  - bAbI (QA)

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

SST (Sentimental Analysis)

Task	Binary	Fine-grained
MV-RNN	82.9	44.4
RNTN	85.4	45.7
DCNN	86.8	48.5
PVec	87.8	48.7
CNN-MC	88.1	47.4
DRNN	86.6	49.8
CT-LSTM	88.0	51.0
DMN	**88.6**	**52.1**

WSJ-PTB (POS-Tagging)

Model	Acc (%)
SVMTool	97.15
Sogaard	97.27
Suzuki et al.	97.40
Spoustova et al.	97.44
SCNN	97.50
DMN	**97.56**

Table 3. Test accuracies on WSJ-PTB

"Ask Me Anything: Dynamic Memory Networks for Natural Language Processing" (2015)

Dynamic Memory Networks

✘ 첫 iteration에서는 best가 처음 높은 attention score을 가졌지만,
두 번째부터는 "is best described" 라는 맥락에서 사용되었다는 것을 파악하고 "lukewarm(미적지근한)"의 score가 높아짐

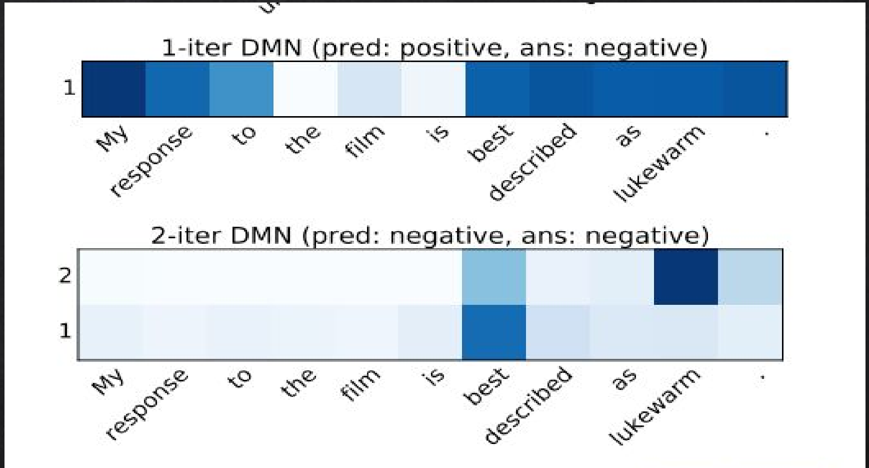


Figure 5. These sentence demonstrate cases where initially positive words lost their importance after the entire sentence context became clear either through a contrastive conjunction ("but") or a modified action "best described."

Neural Turing Machine

- ✘ 앞으로 뉴럴넷한테 보다 어려운 일을 시키려면
  - 모든 걸 다 기억시킬 순 없으니.. 알고리즘 자체를 가르쳐야
- ✘ 제일 간단한 알고리즘들
  - Copy-Paste (복붙) / Sorting (정렬)
- ✘ 기존의 뉴럴넷은 어떻게 Copy를 학습?
  - 가능한 모든 입력을 만들어서 Auto-Encoding
- ✘ 그런데 사실 복사는 레지스터가 있어야 하고...
  - External Memory가 있으면 좋을듯!

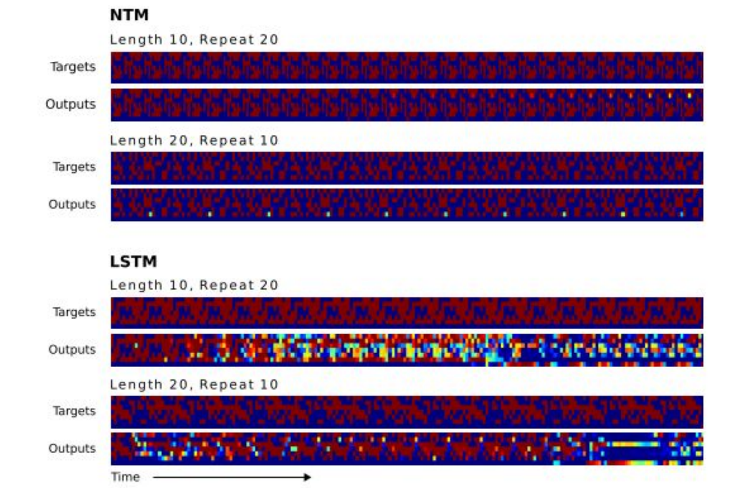


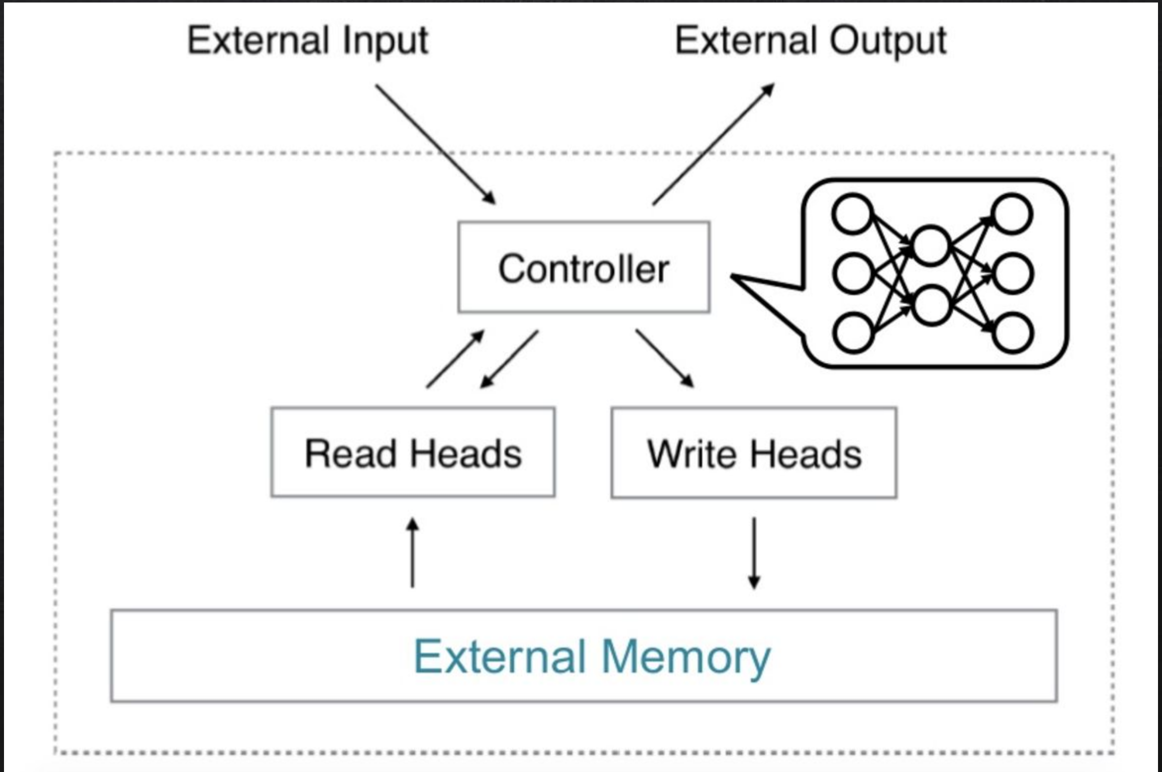
Figure 8: NTM and LSTM Generalisation for the Repeat Copy Task. NTM generalises almost perfectly to longer sequences than seen during training. When the number of repeats is increased it is able to continue duplicating the input sequence fairly accurately; but it is unable to predict when the sequence will end, emitting the end marker after the end of every repetition beyond the eleventh. LSTM struggles with both increased length and number, rapidly diverging from the input sequence in both cases.

"Neural Turing Machines" (2014)

Neural Turing Machine

Check out these awesome visualizations!

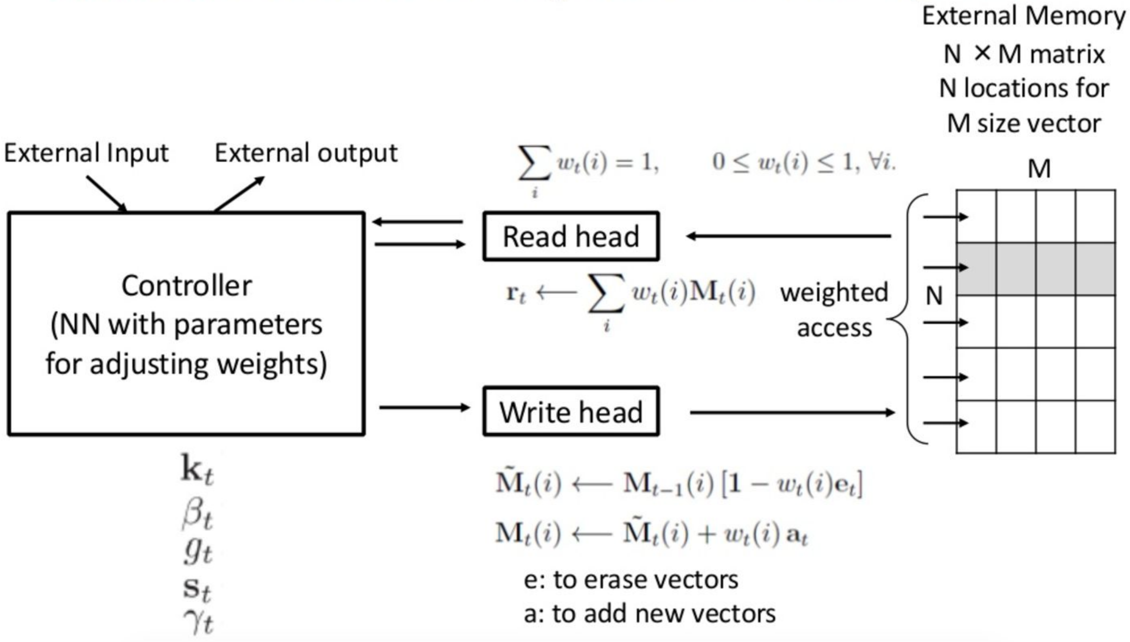
- distill blog
- Mark Chang's blog
- Tristan Deleu's blog
- Kiho Suh's slides <- most pictures from here



"Neural Turing Machines" (2014)

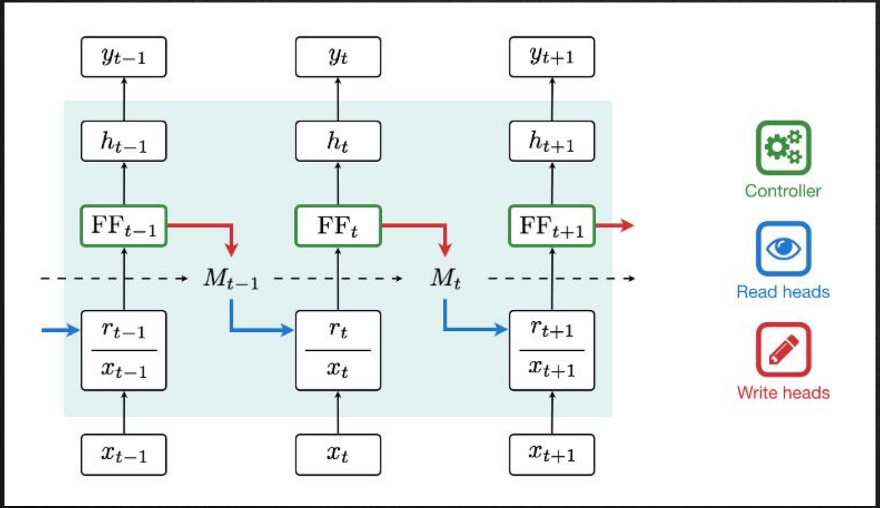
Neural Turing Machine

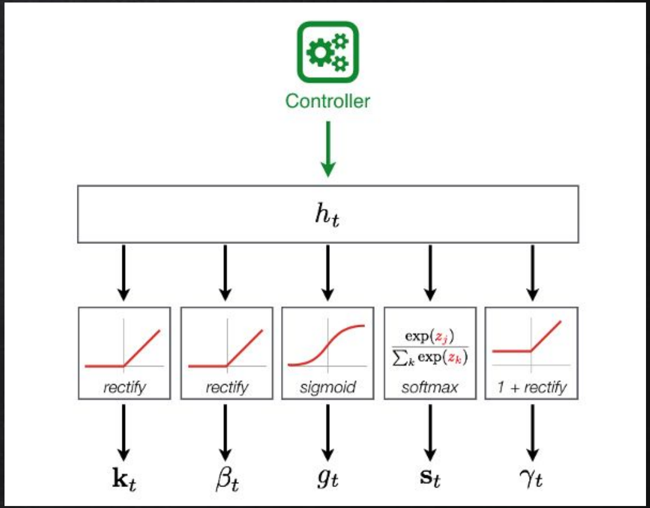
- Read/Write heads use weights to access external memory.
- Weights are determined by the parameters on controller.
- Parameters are learned from large amount of external I/O data.



“Neural Turing Machines” (2014)

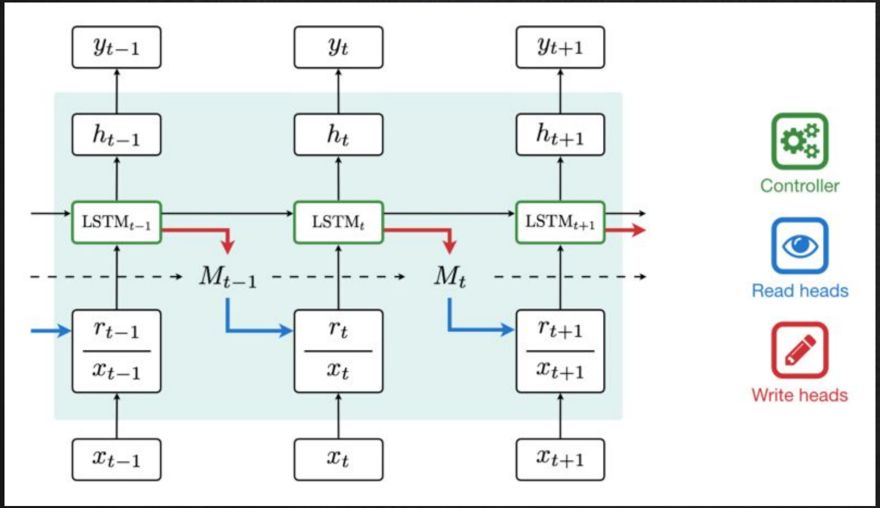
Neural Turing Machine

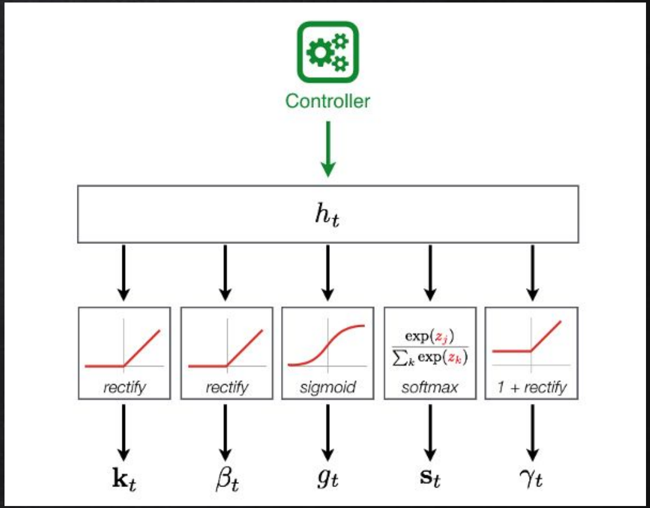




"Neural Turing Machines" (2014)

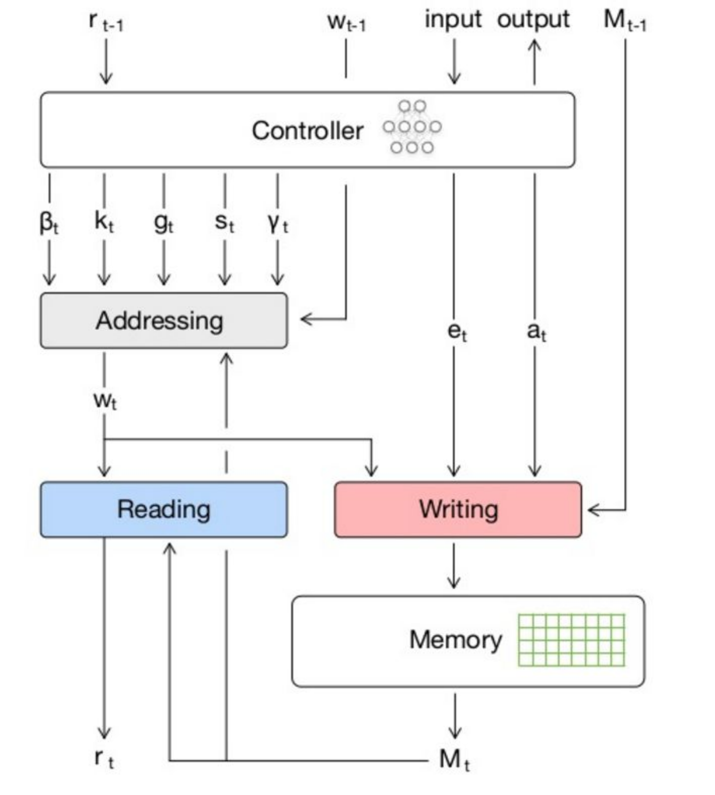
Neural Turing Machine





"Neural Turing Machines" (2014)

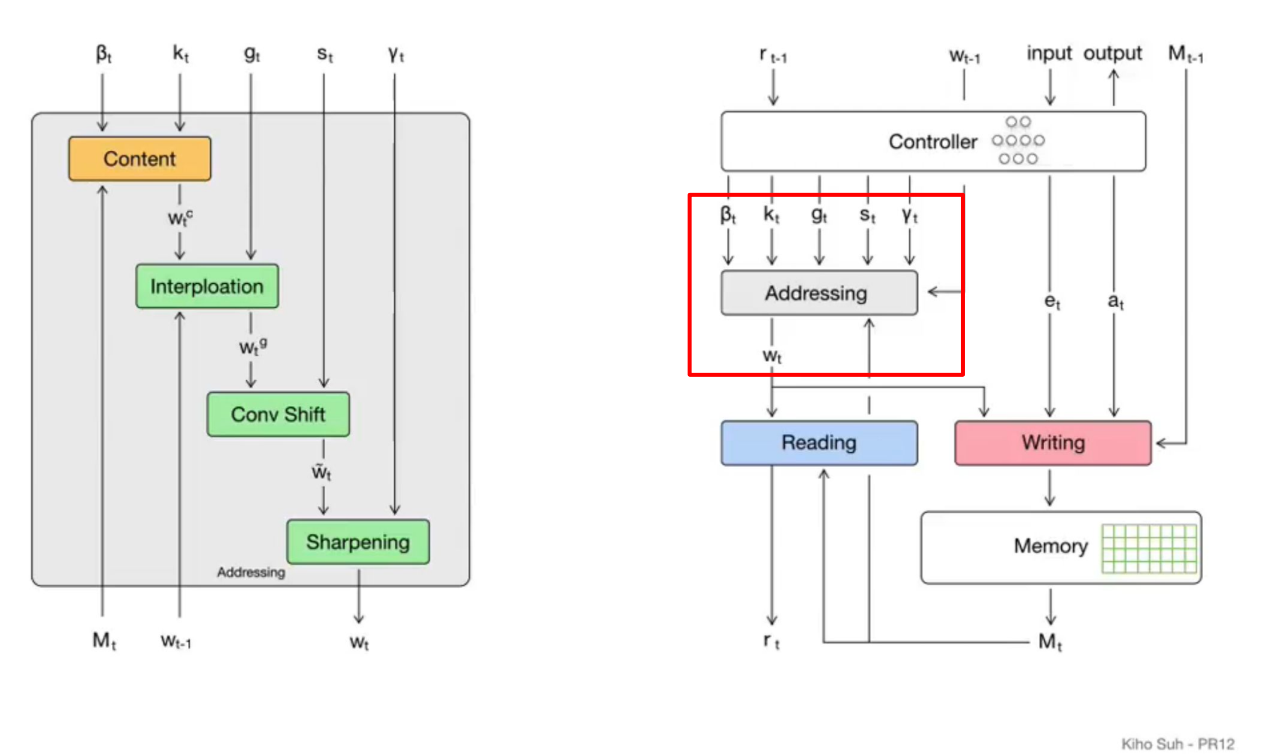
Neural Turing Machine



"Neural Turing Machines" (2014)

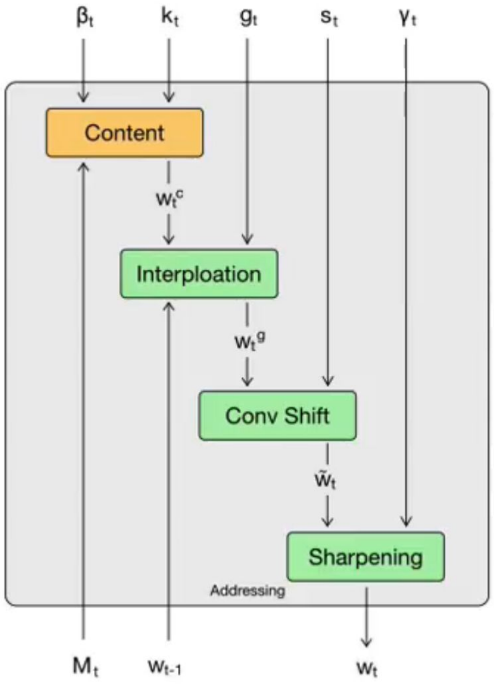
Addressing

- 어떻게 w_t를 만들까?
  - 메모리의 어떤 부분에 집중할까?

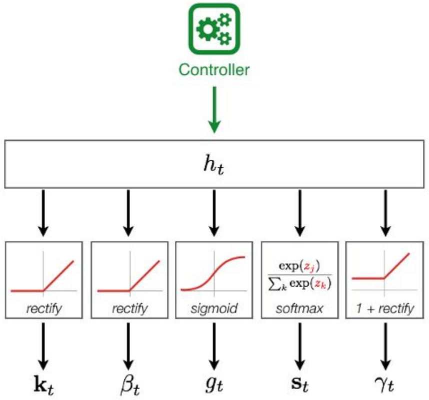


“Neural Turing Machines” (2014)

Selective Memory

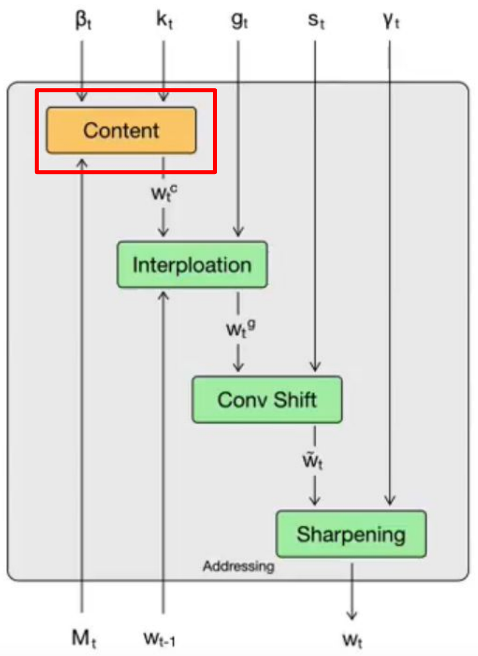


- Key design: how network interacts with memory.
- Making sure not interacting with whole memory at once.
- Do not want to lose nice properties of independence between memory and computation.



“Neural Turing Machines” (2014)

Content Addressing



$$w_t^c(i) \longleftarrow \frac{\exp\Big(\beta_t K\big[\mathbf{k}_t, \mathbf{M}_t(i)\big]\Big)}{\sum_j \exp\Big(\beta_t K\big[\mathbf{k}_t, \mathbf{M}_t(j)\big]\Big)}$$

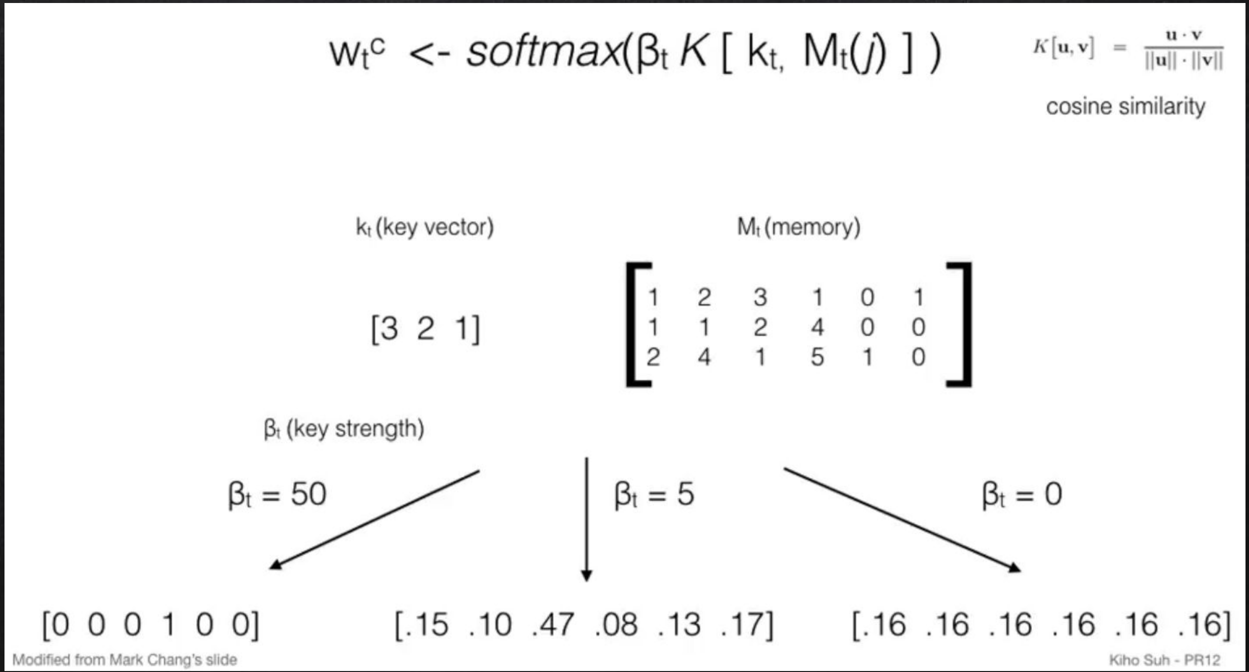
$$\Downarrow$$

w_t^c <- *softmax*(β_t K [k_t, $M_t(j)$])

Complete pattern or specific version of approximate guess

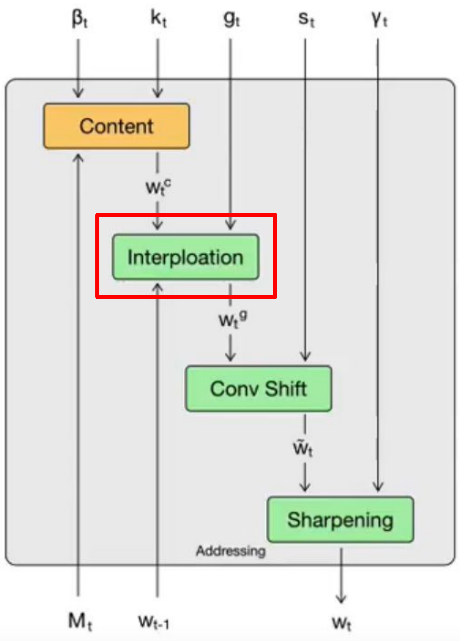
"Neural Turing Machines" (2014)

Content Addressing



"Neural Turing Machines" (2014)

Interpolation (Location Addressing)



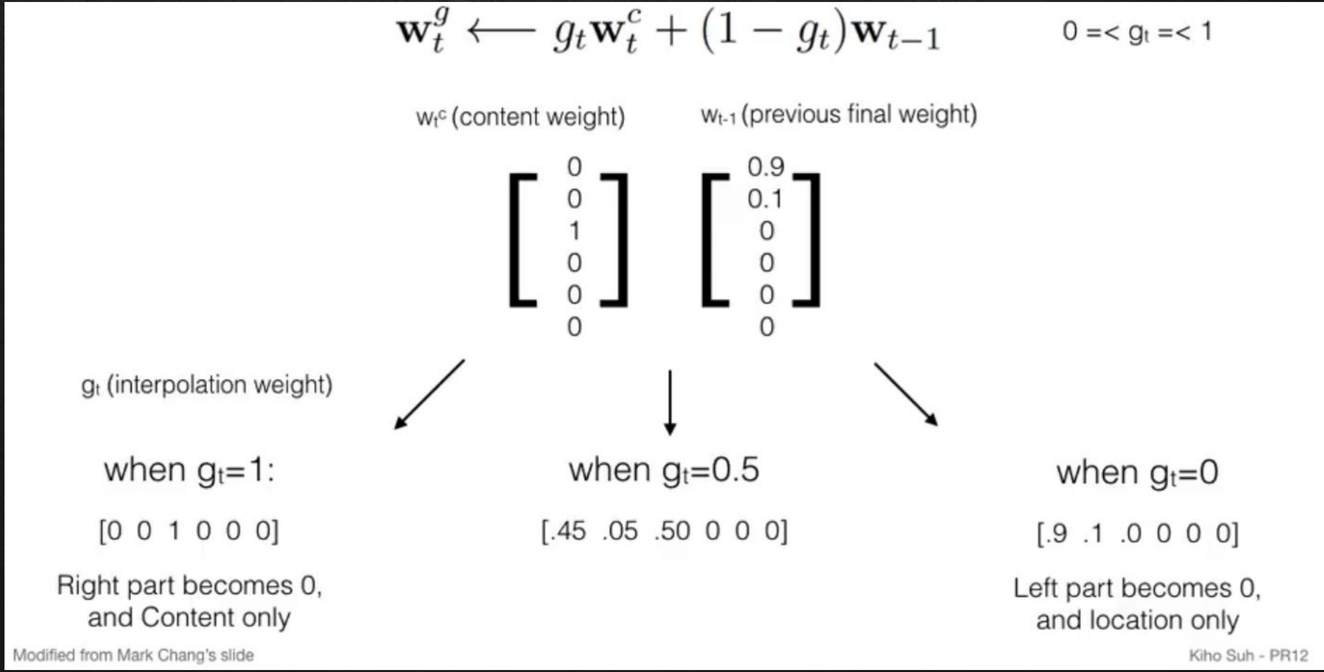
$$\mathbf{w}_t^g \longleftarrow g_t \mathbf{w}_t^c + (1 - g_t)\mathbf{w}_{t-1}$$

0 =< g_t =< 1

Complete pattern or specific version of approximate guess

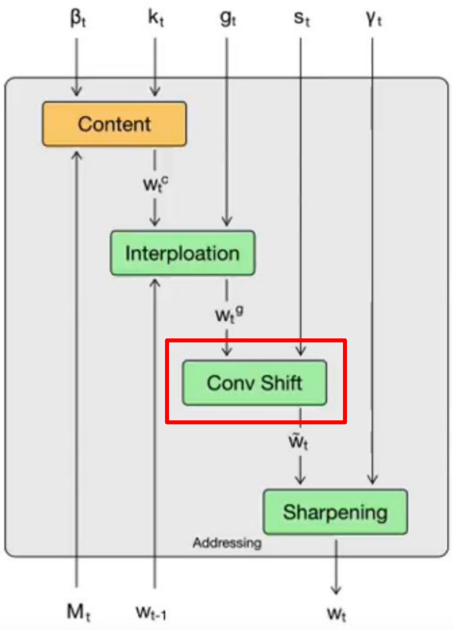
"Neural Turing Machines" (2014)

Interpolation (Location Addressing)



"Neural Turing Machines" (2014)

Convolutional Shift (Location Addressing)

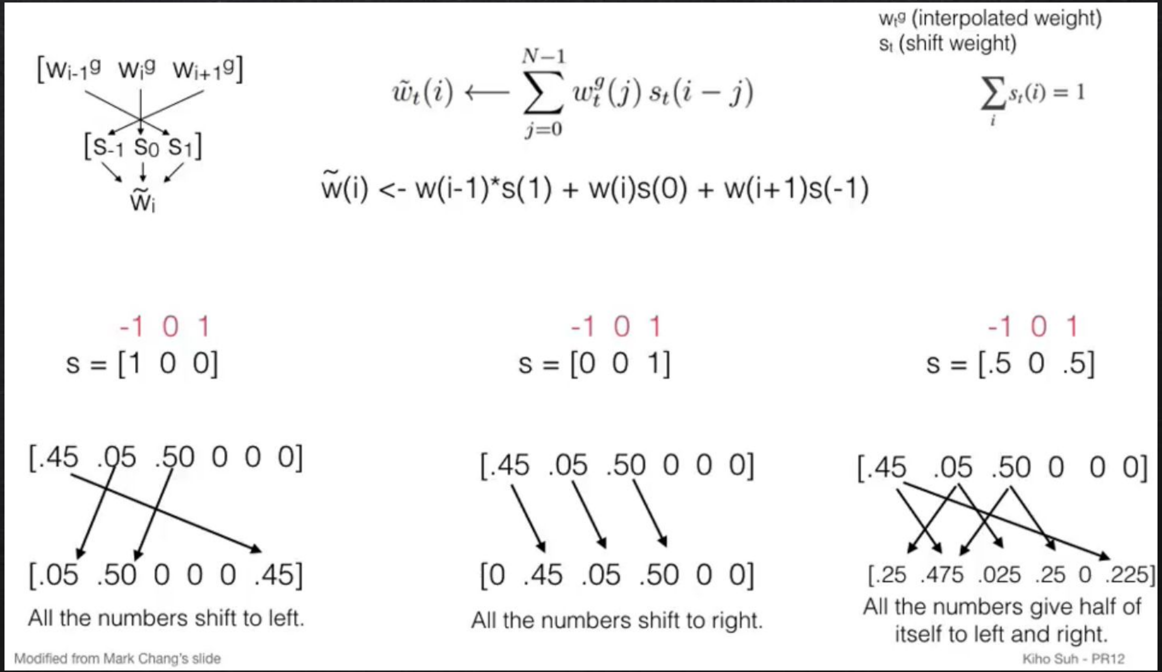


$$\tilde{w}_t(i) \longleftarrow \sum_{j=0}^{N-1} w_t^g(j)\, s_t(i-j)$$

Controller says how wide area should be affected by the action of the head

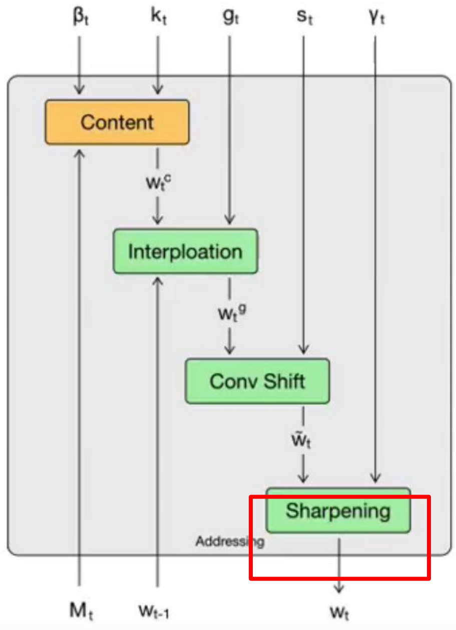
"Neural Turing Machines" (2014)

Convolutional Shift (Location Addressing)



"Neural Turing Machines" (2014)

Sharpening (Location Addressing)



$$w_t(i) \longleftarrow \frac{\tilde{w}_t(i)^{\gamma_t}}{\sum_j \tilde{w}_t(j)^{\gamma_t}}$$

The convolution in the previous step can blur so sharpening. Finally obtain the address (weight value for each memory location) of the memory that Controller thinks we need.

"Neural Turing Machines" (2014)

Sharpening (Location Addressing)

$\tilde{w}_t$ (shifted weight)

γ_t (sharpening)

$$w_t(i) \longleftarrow \frac{\tilde{w}_t(i)^{\gamma_t}}{\sum_j \tilde{w}_t(j)^{\gamma_t}}$$

$\gamma_t >= 1$

[0 .45 .05 .50 0 0]

$\gamma_t = 50$ | $\gamma_t = 5$ | $\gamma_t = 0$

[0 0 0 1 0 0] | [0 .37 0 .62 0 0] | [.16 .16 .16 .16 .16 .16]

because γ is much bigger compared to 5 and 0, the array is sharpened.

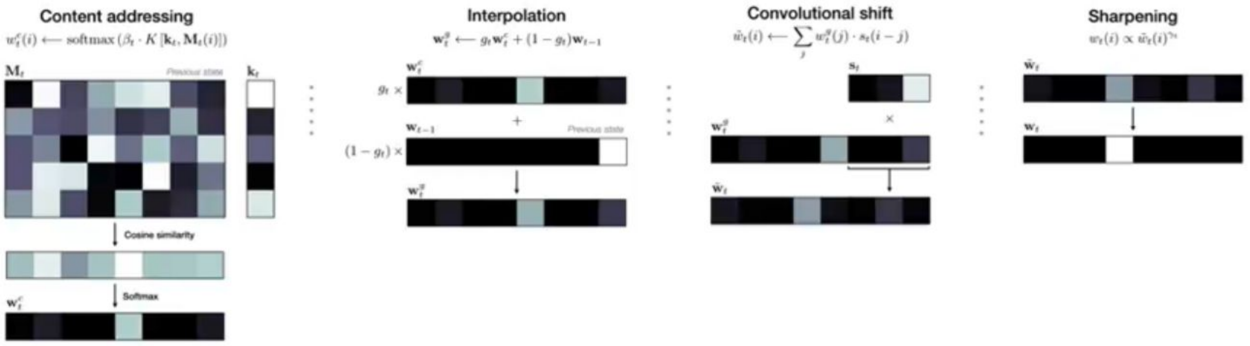
because γ is smaller than 1, here the w_t is even more blurred.

Modified from Mark Chang's slide

Kiho Suh - PR12

"Neural Turing Machines" (2014)

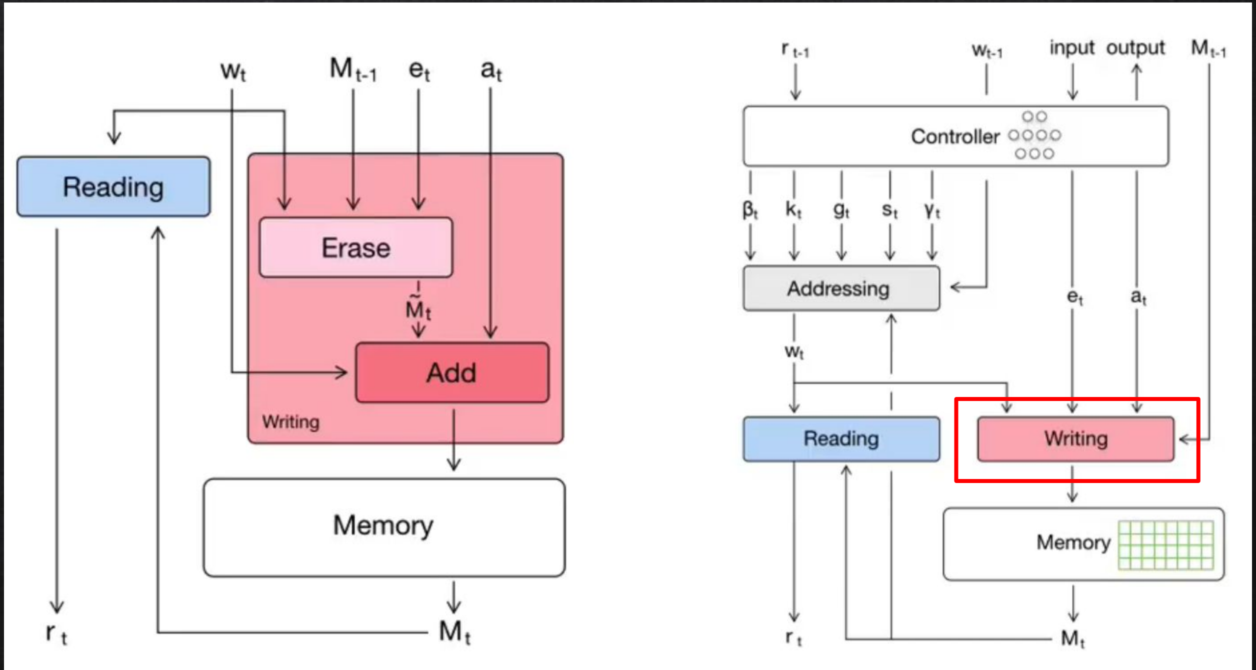
Addressing



Addressing is “soft” and distributed across the entire memory. However, it is focused on very few cells quantitatively.

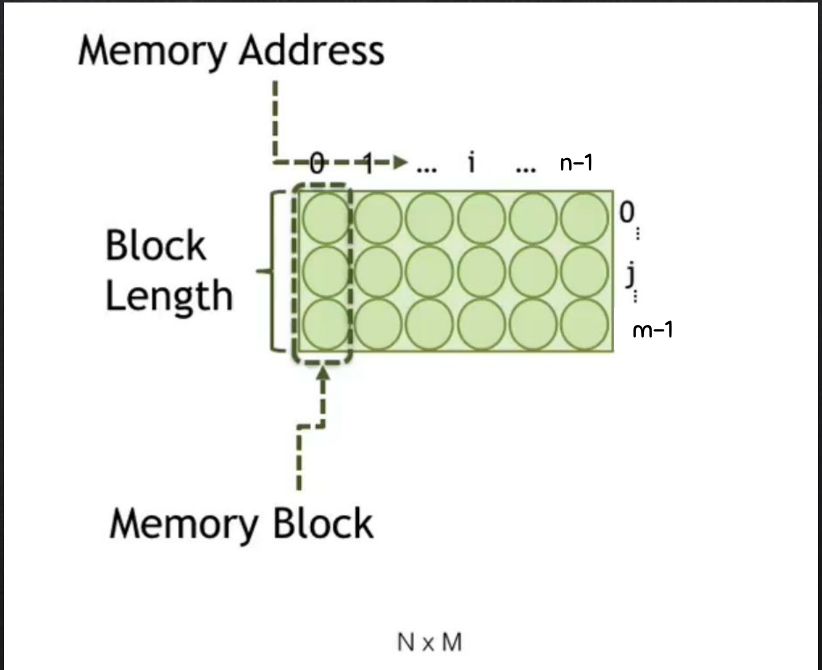
“Neural Turing Machines” (2014)

Writing



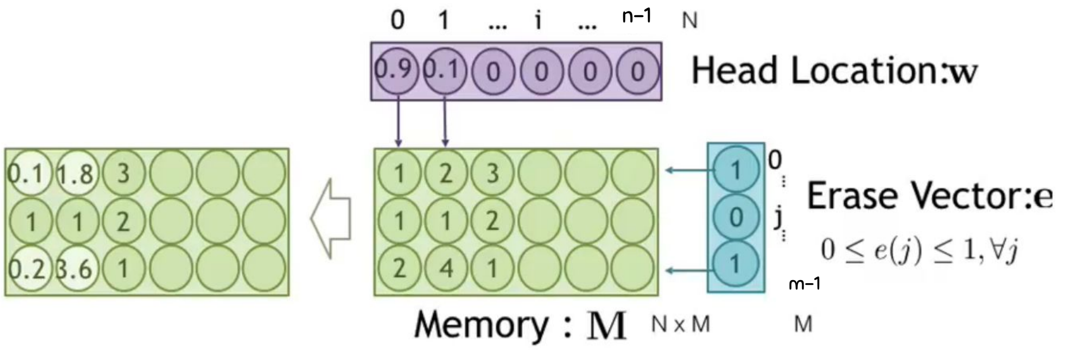
"Neural Turing Machines" (2014)

Memory



“Neural Turing Machines” (2014)

Erase



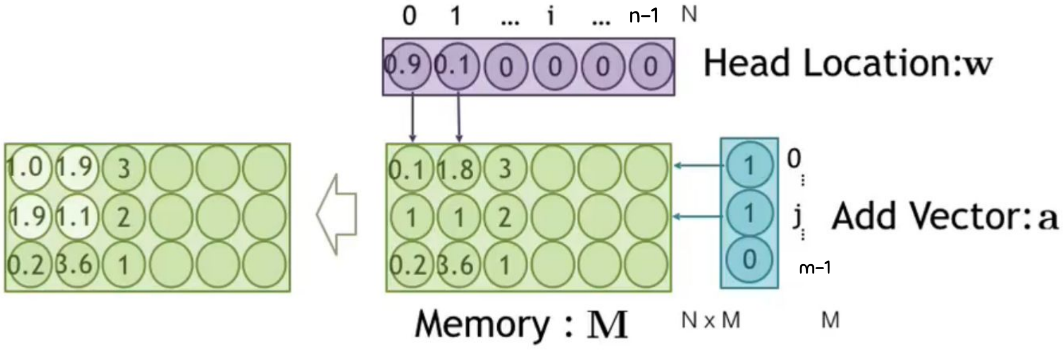
Erase Operation: $\mathbf{M}(i) \leftarrow (1 - w(i)\mathbf{e})\mathbf{M}(i)$

$$M = \begin{bmatrix} 1(1-0.9) & 2(1-0.1) & 3 & \dots \\ 1 & 1 & 2 & \dots \\ 2(1-0.9) & 4(1-0.1) & 1 & \dots \end{bmatrix} = \begin{bmatrix} 0.1 & 1.8 & 3 & \dots \\ 1 & 1 & 2 & \dots \\ 0.2 & 3.6 & 1 & \dots \end{bmatrix}$$

From Mark Chang's slide

“Neural Turing Machines” (2014)

Add



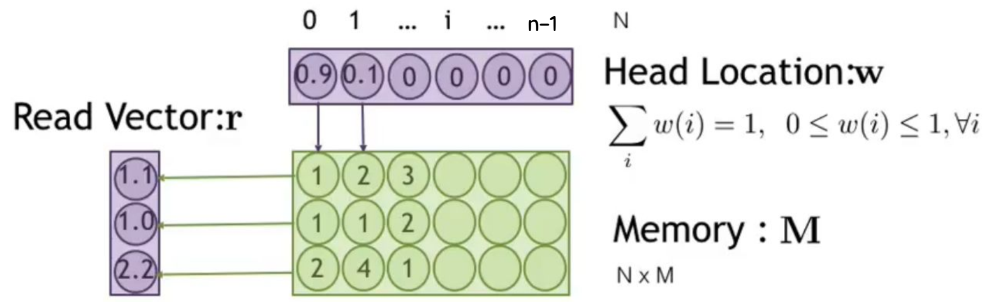
Add Operation: $\mathbf{M}(i) \leftarrow \mathbf{M}(i) + w(i)\mathbf{a}$

$$M = \begin{bmatrix} 0.1 + 0.9 & 1.8 + 0.1 & 3 & ... \\ 1.0 + 0.9 & 1.0 + 0.1 & 2 & ... \\ 0.2 & 3.6 & 1 & ... \end{bmatrix} = \begin{bmatrix} 1.0 & 1.9 & 3 & ... \\ 1.9 & 1.1 & 2 & ... \\ 0.2 & 3.6 & 1 & ... \end{bmatrix}$$

From Mark Chang's slide

"Neural Turing Machines" (2014)

Read



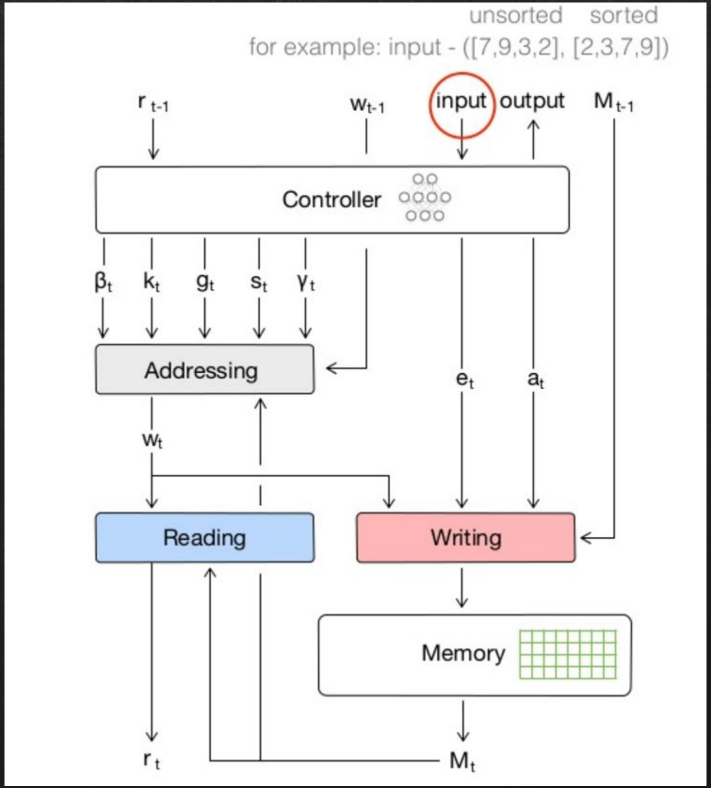
Read Operation: $\mathbf{r} \leftarrow \sum_i w(i)\mathbf{M}(i)$

$$\begin{bmatrix} r_0 \\ r_1 \\ r_2 \end{bmatrix} = \begin{bmatrix} 1 * 0.9 + 2 * 0.1 \\ 1 * 0.9 + 1 * 0.1 \\ 2 * 0.9 + 4 * 0.1 \end{bmatrix} = \begin{bmatrix} 1.1 \\ 1.0 \\ 2.2 \end{bmatrix}$$

From Mark Chang's slide

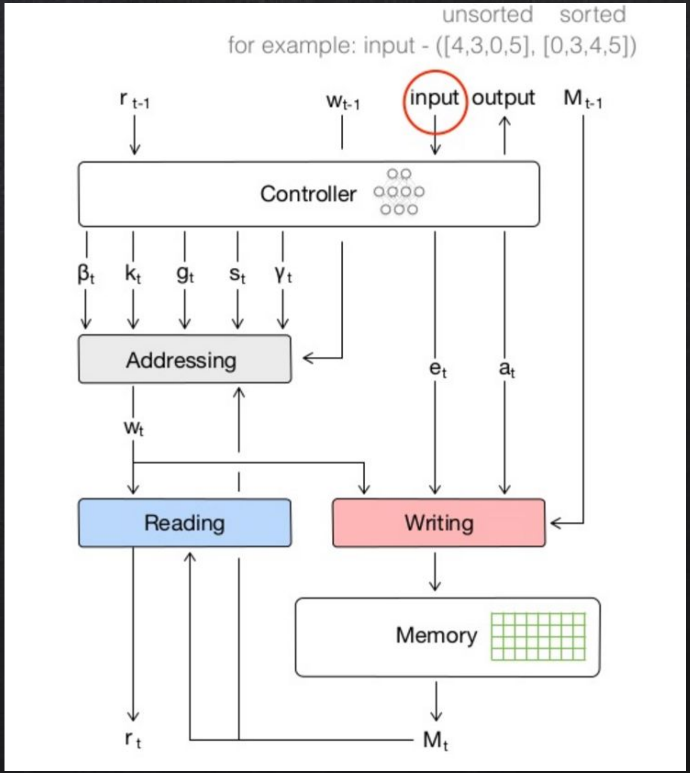
"Neural Turing Machines" (2014)

Neural Turing Machine



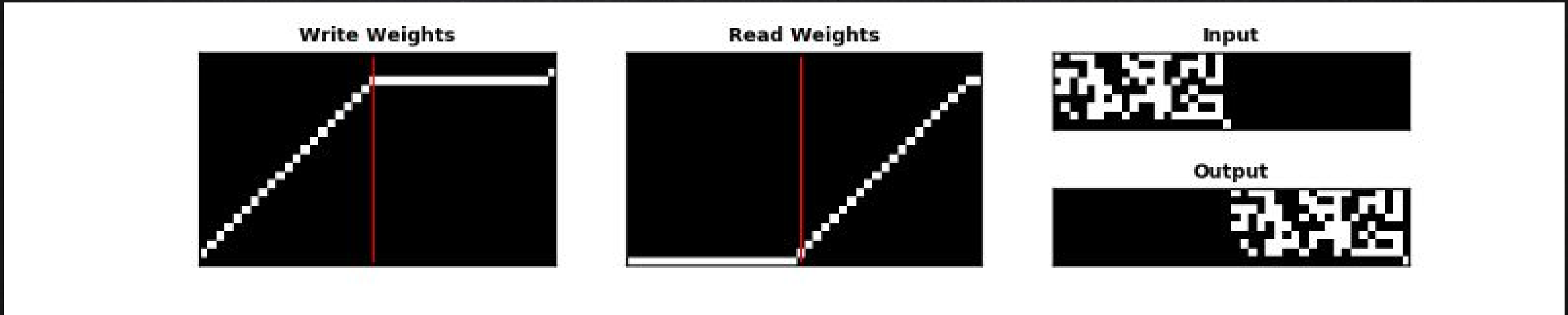
"Neural Turing Machines" (2014)

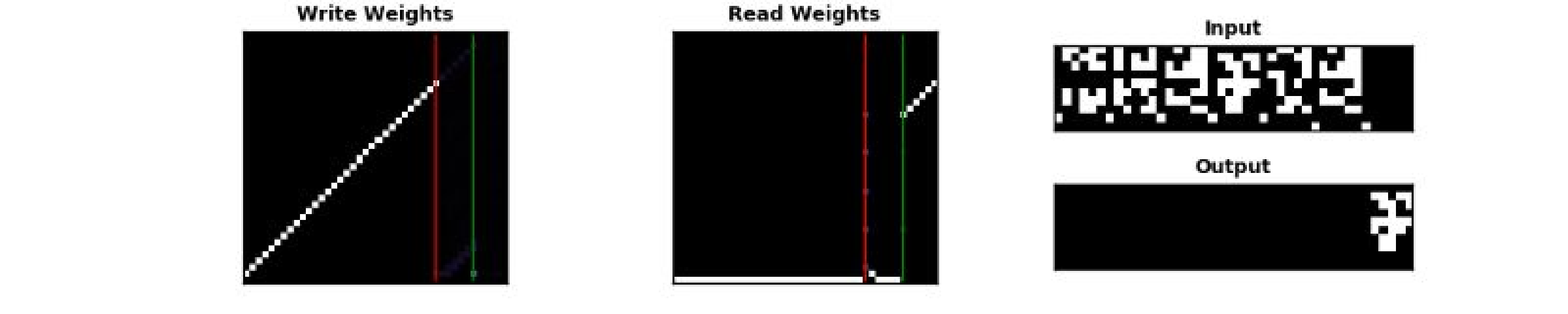
Neural Turing Machine



"Neural Turing Machines" (2014)

Copy / Associative Recall





"Neural Turing Machines" (2014)

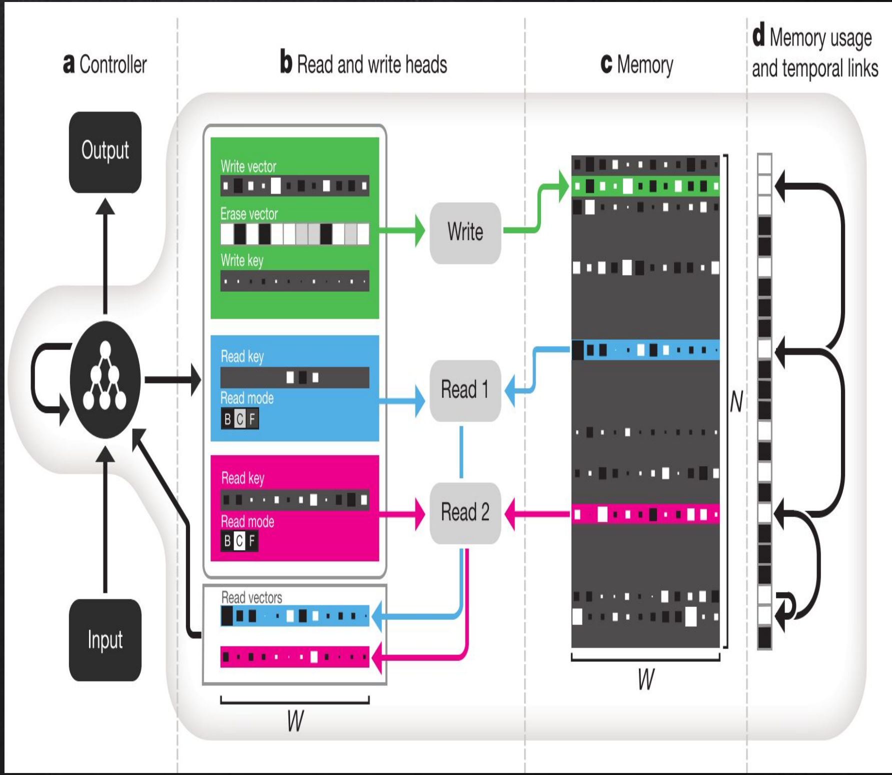
2.

Advanced External Memory Architectures

Differentiable Neural Computer (DNC)
Life-long Memory Module
Context-Sequence Memory Networks

Differentiable Neural Computer

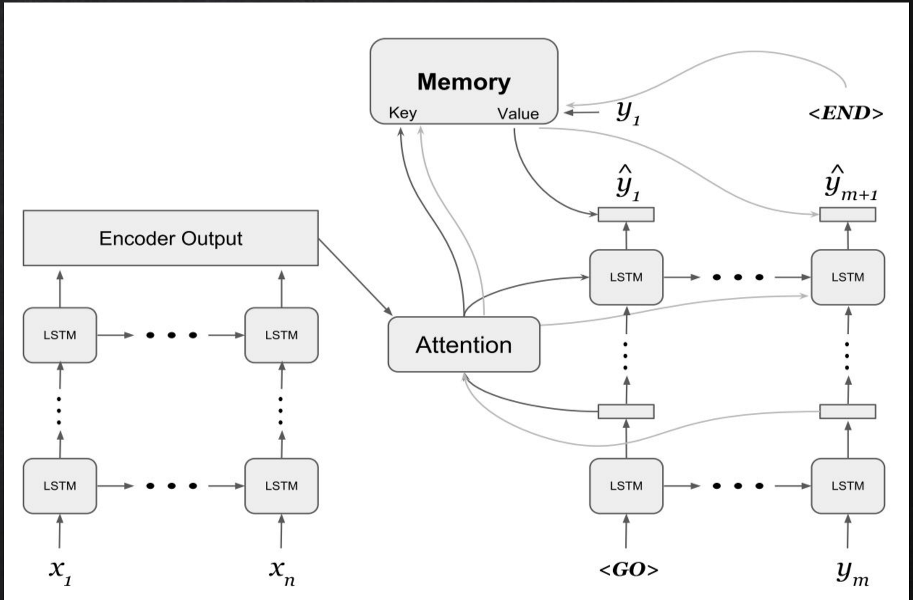
- Advanced addressing mechanisms
  - Content Based Addressing
  - Temporal Addressing
    - Maintains notion of sequence in addressing
    - Temporal Link Matrix L (NxN)
      - L[i,j]
        - degree to location i was written to after location j.
  - Usage Based Addressing



"Hybrid computing using a neural network with dynamic external memory" (2016)

Life-long Memory Module

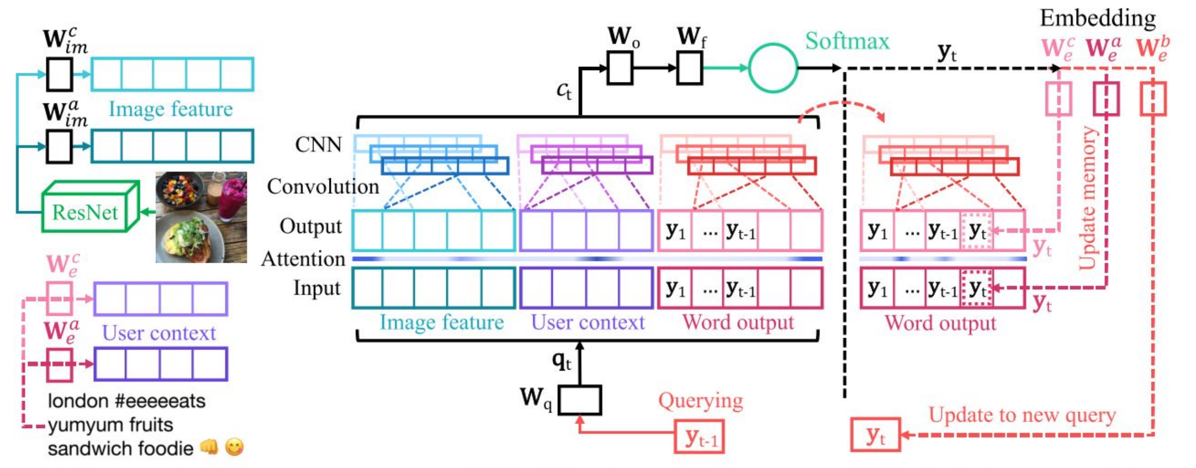
- Inspired by Matching Network for One-shot Learning
- Memorize every given sentences in memory
  - Locality Sensitive Hash (LSH)
  - Life-long learning
- Improve Attention Mechanism
  - Not only attend on source words
  - But also attend of memory



"Learning To Remember Rare Events" (2017)

Context Sequence Memory Networks

- ✘ Image Captioning
  - ○ Hashtag Prediction
- ✘ No RNN
  - ○ Sequentially store all of previous generated words into memory
- ✘ User context memory
  - ○ TF-IDF
  - ○ Top N words for given user



(a) Memory setup (b) Prediction step (c) Word output memory update

"Attend to You: Personalized Image Captioning with Context Sequence Memory Networks" (2017)

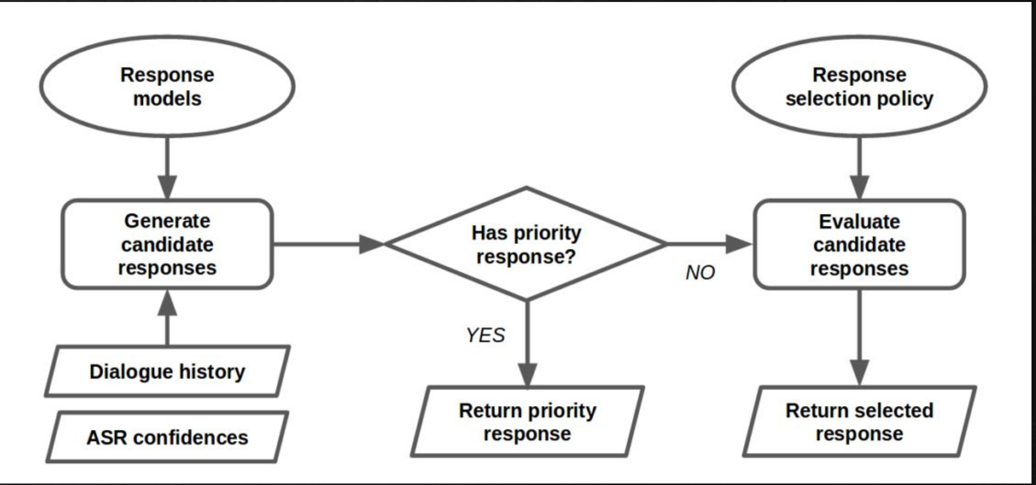
3.

Advanced Dialogue Architecture

MILABOT
Dialogue Based Language Learning
Learning End-to-End Goal-Oriented Dialog
Deep RL for Dialogue Generation
Advanced Learning for Neural Dialogue Generation

MILABot

- ✘ MILA's Amazone Alexa Prize 2017
  - Ensemble of 22 response systems
  - Model Selection policy => Reinforcement Learning
- ✘ Template
  - Initiator, Story
  - Knowledge based
- ✘ Retrieval
  - VHRED, Skip-Thought, Dual-Encoder
  - Bag-of-Words, TF-IDF
- ✘ Search Engine
  - MS Macro
- ✘ Generation
  - Question Generator



"A Deep Reinforcement Learning Chatbot" (2017)

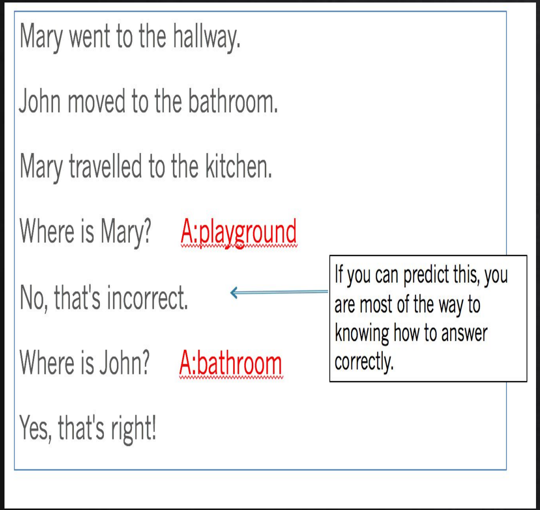
MILABot

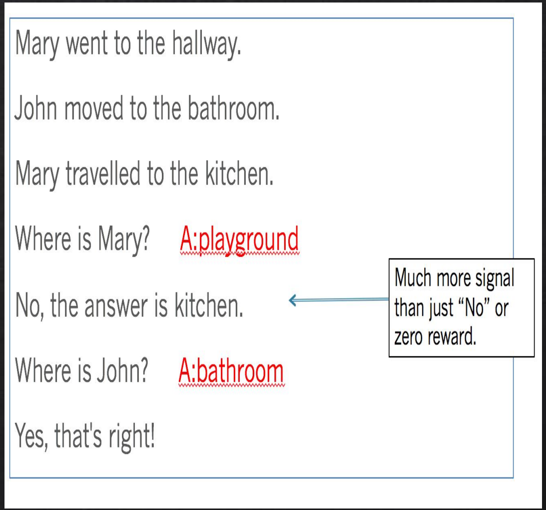
Dialogue	Dialogue
SYSTEM: Hi. This is an Alexa Prize socialbot. USER: Hi SYSTEM: Do you have pets? USER: Yes, I have two rabbits.	... USER: Let's talk about movies. SYSTEM: What's your favorite movie? USER: star wars movie
Candidate Responses	**Candidate Responses**
BoWEscapePlan: I don't have an answer for this. BoWFactGenerator: Did you know that male rabbits are called bucks, females are does? GRUQuestionGenerator: How about the American family of it? Initiatorbot: Do you believe in love at first sight? LSTMClassifierMSMarco: Mar 23, 2017 Two Parts: Learning What to Feed Your Rabbit Learning How to ... DualEncoderRedditNews: bye doggie DualEncoderRedditPolitics: Who's edger u shboys? **Alicebot: Hurrah! Two is a good number of rabbits.** Elizabot: OK, but can you elaborate a bit? VHREDRedditMovies: Hmm. Nope, never heard of it. VHREDSubtitles: Would you like to see them? HelpResponse: I would really like to talk about news, politics or movies.	BoWEscapePlan: I don't have an answer for this. BoWFactGenerator: Here's an interesting fact. The international telephone dialing code for Antarctica is 672. GRUQuestionGenerator: how about your New York City? Initiatorbot: How was your day? LSTMClassifierMSMarco: A third Anthology film will be released in 2020. DualEncoderRedditNews: Now whisper it to me , one more time bby. DualEncoderRedditPolitics: hahaha on mobile this ain't even close. I wish I could show you but this looks awful I'm sorry man. Alicebot: What did you like about the robots in that movie? Elizabot: How does that make you feel? **Evi: Star Wars movie a movie in the Star Wars series.** VHREDRedditMovies: Oh please. Please. Pleeeease. Let this happen. VHREDSubtitles: What? HelpResponse: I would really like to talk about news, politics or movies.

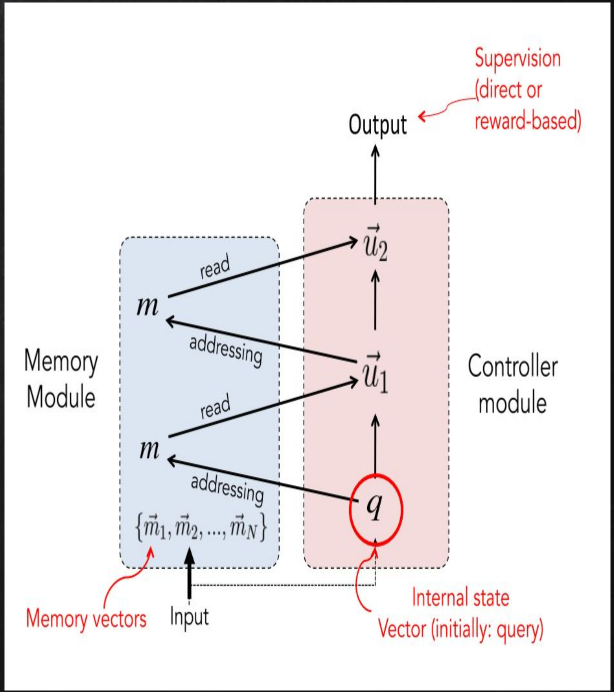
"A Deep Reinforcement Learning Chatbot" (2017)

Dialog-based Language Learning

- 아기처럼 대화를 통해 언어를 배우자
  - 처음에는 Yes/No가 무슨 뜻인지도 모르지만, 계속 대답을 해주자
  - 최종 answer를 평가함으로써 각 단어의 의미/문맥 유추
- (Reward-based) Imitation Learning, Forward Prediction



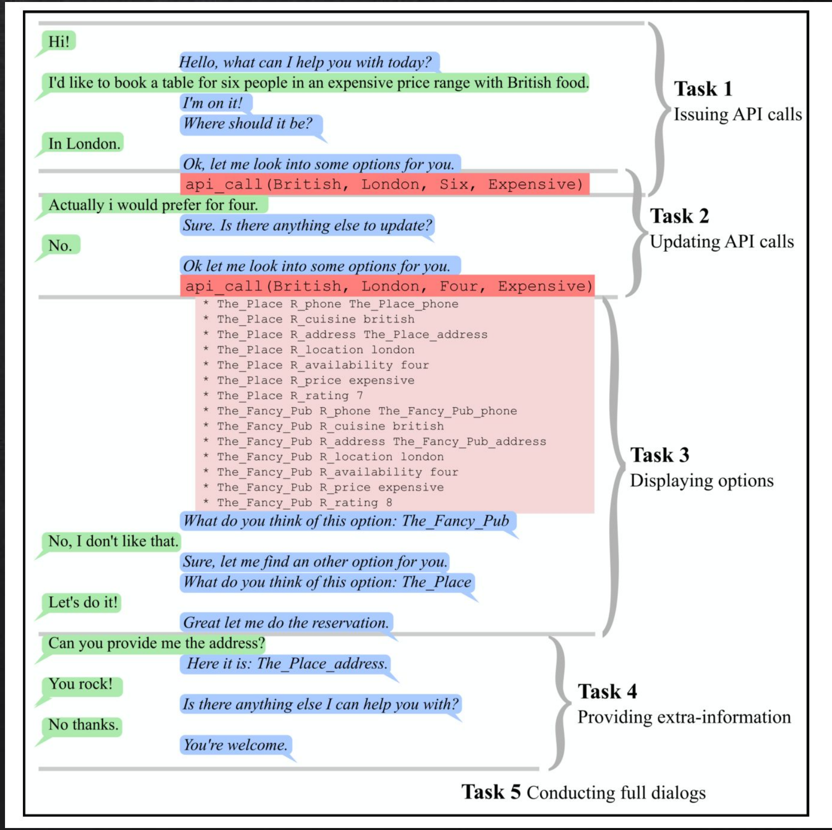




"Dialog-based Language Learning" (2016)

End-to-End Goal-Oriented Dialog

- ✘ 챗봇을 MLE 기반으로 훈련시키면
  - 현재 문장에서 나올 수 있는 가장 그럴듯한 대답 생성
  - ‘의도 파악/과제 수행’ 보다는 정확한 문법 (Language Modeling) 구사에 초점이 맞춰짐
- ✘ 챗봇에게 말할 수 있는 권한 외에도 API 사용권한을 부여
  - 식당 예약 API
  - ‘시간, 장소, 인원’ 등의 정보 필요
    - 정보를 채우기 위해 ‘고객에게 질문하기’
- ✘ Model: End-to-End Memory Networks
  - Multi-hop Attention 으로 ‘세 번 생각한 후', 현재 상황에서 가장 적합한 API 실행



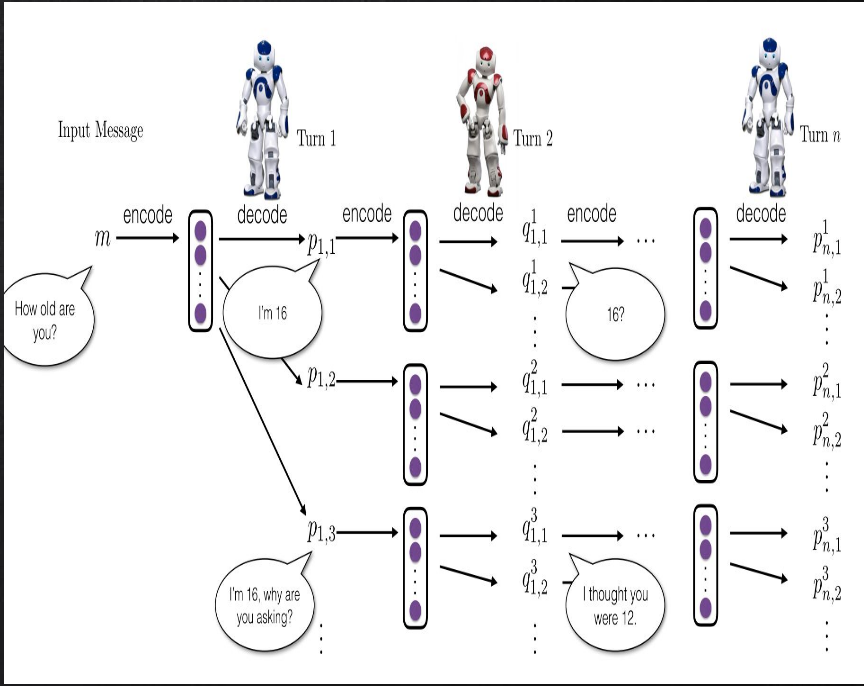
“Learning End-to-End Goal-Oriented Dialog” (2016)

Deep RL for Dialogue Generation

- 사람의 대화의 목적 ≠ log-likelihood 최대화
- '대화 전략'을 학습하자!
  - 대화를 잘 하면 reward를 주자!
- 어떤 대화가 좋은 대화?
  - Reward

$$r(a, [p_i, q_i]) = \lambda_1 r_1 + \lambda_2 r_2 + \lambda_3 r_3$$

  - Ease of answering
  - Information Flow
  - Semantic Coherence



"Deep Reinforcement Learning for Dialogue Generation" (2016)

Adversarial Learning for Neural Dialogue Generation

- ✘ Adversarial REINFORCE
- ✘ Reward for Every Generation Step
- ✘ 사람인지 시험하는 Discriminator

  vs

  사람을 따라하는 대화하는 Generator
- ✘ 대화를 잘 해서 튜링테스트를 통과하자!

For number of training iterations **do**
. **For** i=1,D-steps **do**
. Sample (X,Y) from real data
. Sample $\hat{Y} \sim G(\cdot|X)$
. Update D using (X, Y) as positive examples and $(X, \hat{Y})$ as negative examples.
. **End**
.
. **For** i=1,G-steps **do**
. Sample (X,Y) from real data
. Sample $\hat{Y} \sim G(\cdot|X)$
. Compute Reward r for $(X, \hat{Y})$ using D.
. Update G on $(X, \hat{Y})$ using reward r
. Teacher-Forcing: Update G on (X, Y)
. **End**
End

Input	tell me ... how long have you had this falling sickness ?
Vanilla-MLE	i 'm not a doctor .
Vanilla-Sample	well everything you did was totally untrue .
REINFORCE	i don 't know how long it 's been .
REGS Monte Carlo	A few months, I guess .
Input	so i had the doctors test sammy 's response to conditioning .
Vanilla-MLE	sammy wrote the test sammy wrote the test .
Vanilla-Sample	objects pick objects up objects objects objects objects objects
REINFORCE	i 'm not sure that 's relevant .
REGS Monte Carlo	so he took the pills .
Input	what are you going to do ?
Vanilla-MLE	i 'm going to the movies .
Vanilla-Sample	get him outta here first !
REINFORCE	i 'm going to get you .
REGS Monte Carlo	i 'm going to get you a new car .
Input	they fear your power your intellect .
Vanilla-MLE	you 're the only one who knows what 's going on .
Vanilla-Sample	when they are conquered and you surrender they will control all of us .
REINFORCE	i 'm afraid i 'm not ready yet .
REGS Monte Carlo	i 'm not afraid of your power .

"Adversarial Learning for Neural Dialogue Generation" (2017)

4.

Wrap up!

Dataset / Tokenization / Vectorization
Classification / Sequence Generation
Attention / External Memory
Advanced Deep NLP models

Review

- ✘ Dataset
  - English: SQUAD / bAbI / MS MARCO / Ubuntu / Cornell / xxQA
  - Korean: Sejong / Wiki / Namu / Naver movie sentiment
- ✘ Tokenization
  - Whitespace
  - Regular expression
  - POS-tagger
  - Noun / Verb only
- ✘ Vectorization
  - N-gram
  - TF-IDF
  - CBOW/Skip-gram
  - Word2Vec / Glove
  - Character embedding
  - Byte-pair encoding
  - Positional Encoding

Review

- ✘ Residual Connection
- ✘ Weight Initialization
- ✘ Normalization
  - Batch / Layer / Weight
- ✘ Classification
  - Naive Bayes / Logistic Regression / Random Forest / SVM
  - CNN / RNN (Many-to-one)
- ✘ Ensemble
  - StackNet
- ✘ Sequence Generation
  - RNN Encoder-RNN Decoder
  - CNN Encoder-RNN Decoder
  - CNN Encoder-Decoder (ConvS2S)
  - Self Attention (TransFormer)

Review

- Attention
  - Luong / Bahdanau
  - Global / Local
  - Scoring method
  - Pointer (sentinel)
  - Bidirectional
  - Multi-hop
  - Transformer (Attention-is-all-you-need)
- External Memory
- Advanced Deep QA
  - Goal-oriented (RL)
  - Persona-based
  - Hierarchical Attention
  - Adversarial
  - Generative

THANKS!

Any questions?