Recent Progress in Deep Learning for Natural Language Processing

Zhengdong Lu

DeeplyCurious.AI (深度好奇)

Part-I:
Overview and Background

Overview

Background: *word embedding and composition models*

Progress in terms of tasks:

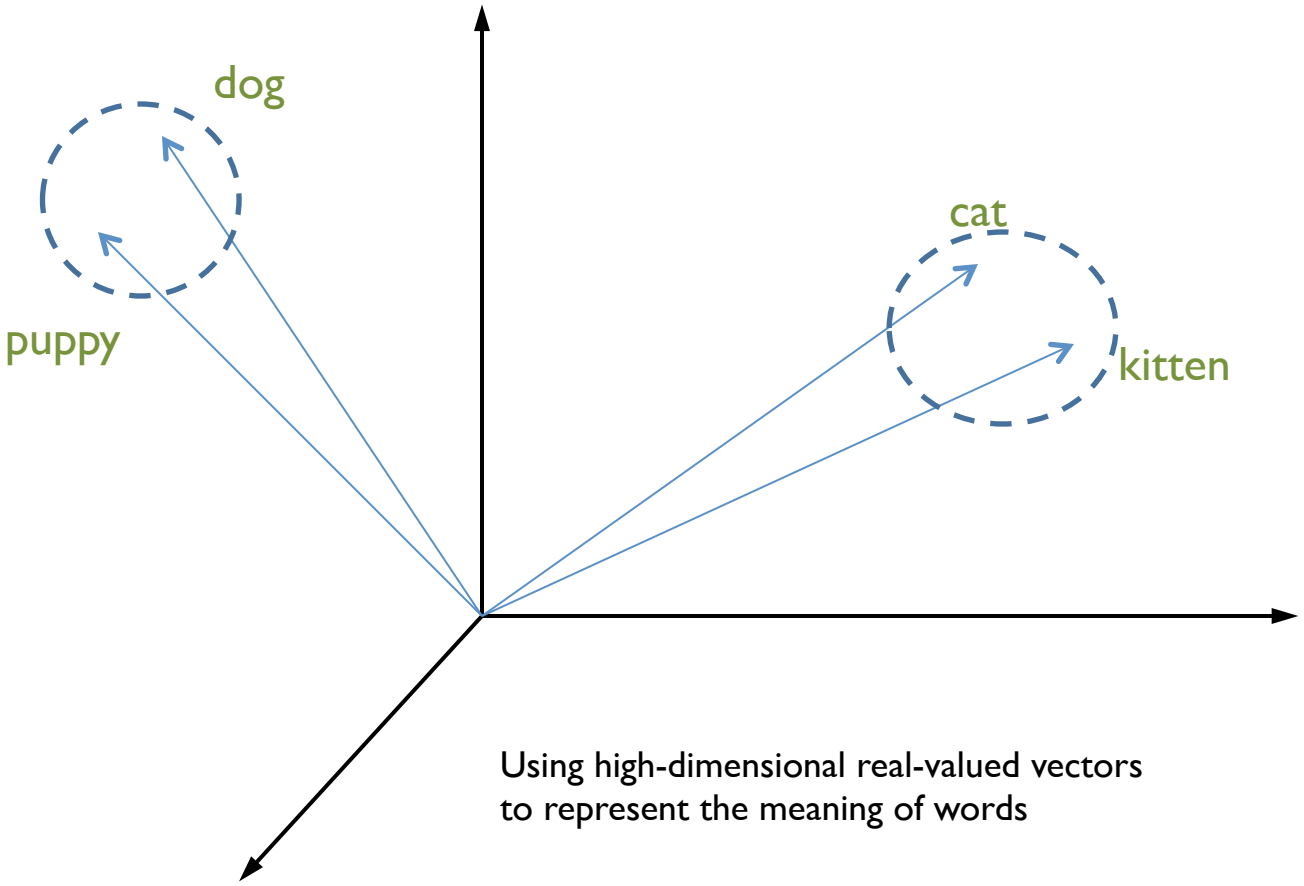
- *Machine translation*
- *Dialogue*
- *Reasoning*
- Image captioning
- Natural language parsing
- … …

Progress in terms of methodology (focus of this tutorial):

- Attention models
- External memories
- Differentiable data structures
- End-to-end learning

Distributed representation of words

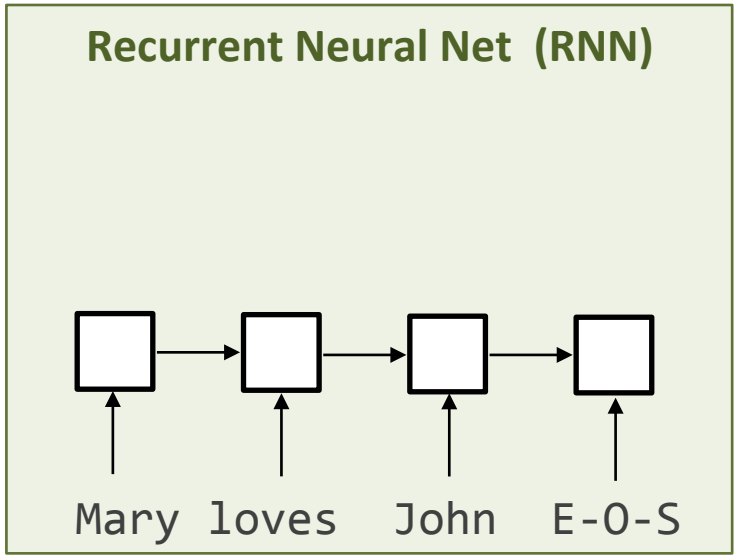
- Distributed representation of words

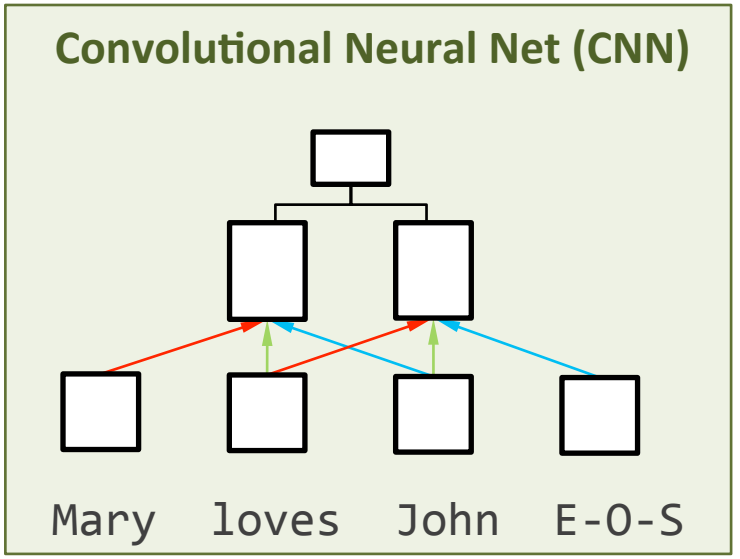


Composition Models

- From words to sentences and beyond
- Architectures that model the syntax and semantics of sentence

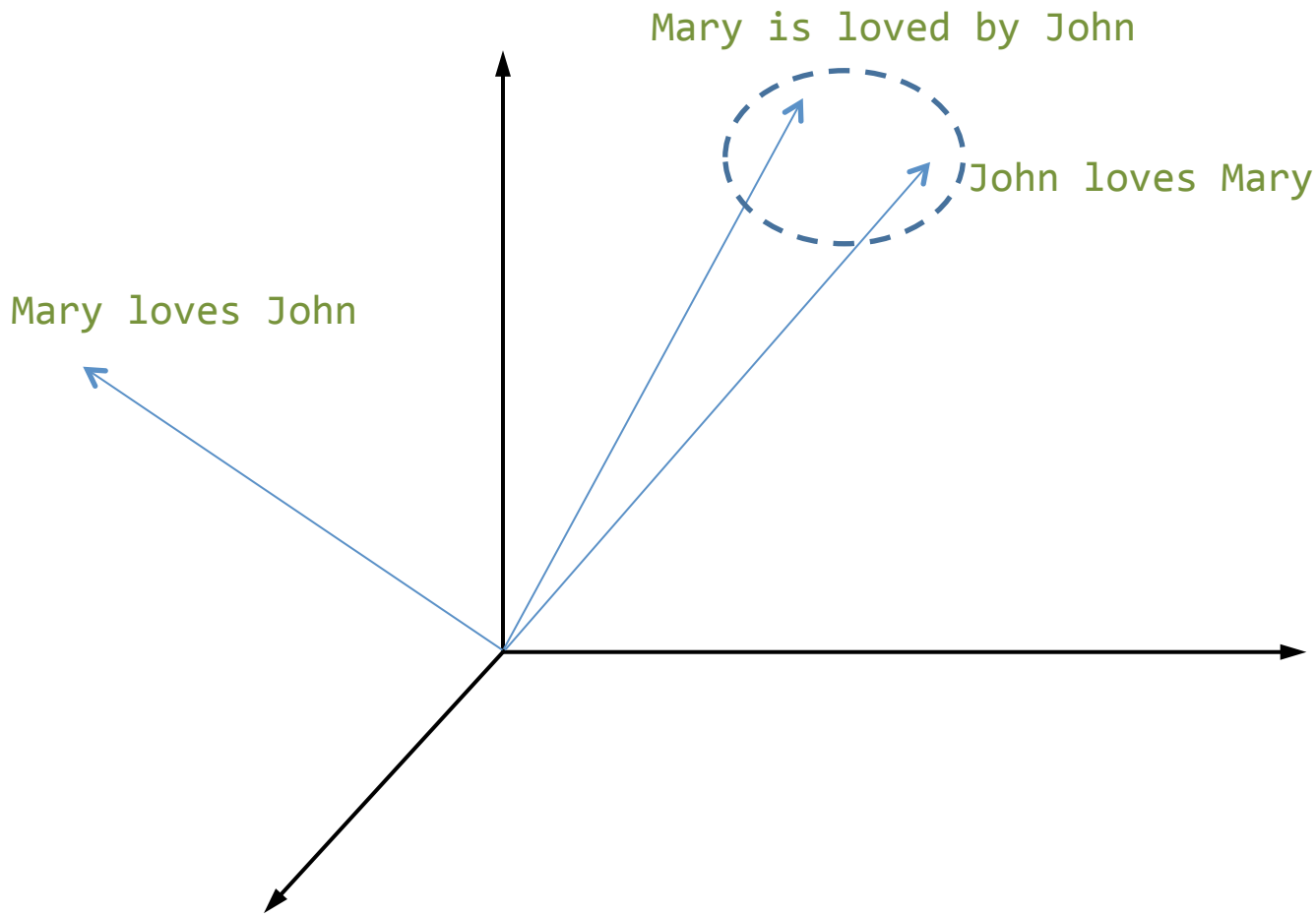
Two basic architectures





Distributed representation of sentences

- Surprisingly, we can use long-enough vectors to represent the meaning of sentences



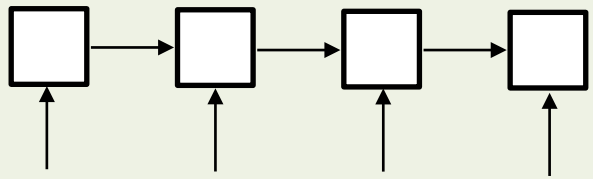
Composition Models (cont'd)

① Sentence as sequence of words, left to right and/or right to left

② Recursively construct the same (one for all) composition, between the history and next word

③ Different gating mechanisms have been proved useful, e.g., LSTM or GRU

Recurrent Neural Net (RNN)

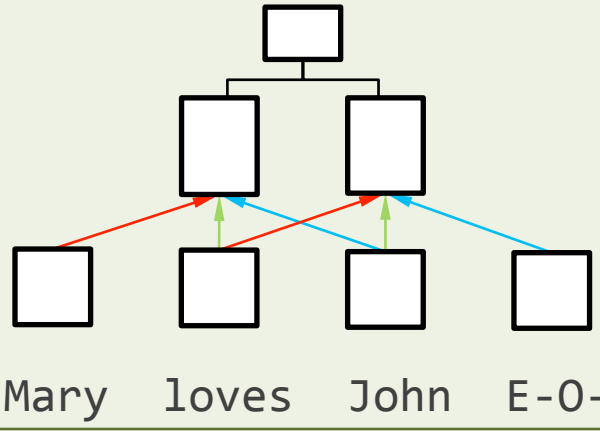


① Bottom up and soft parsing of sentence, create ensemble of parse trees

② Construct all possible compositions, and choose the most suitable ones using pooling (or gating)

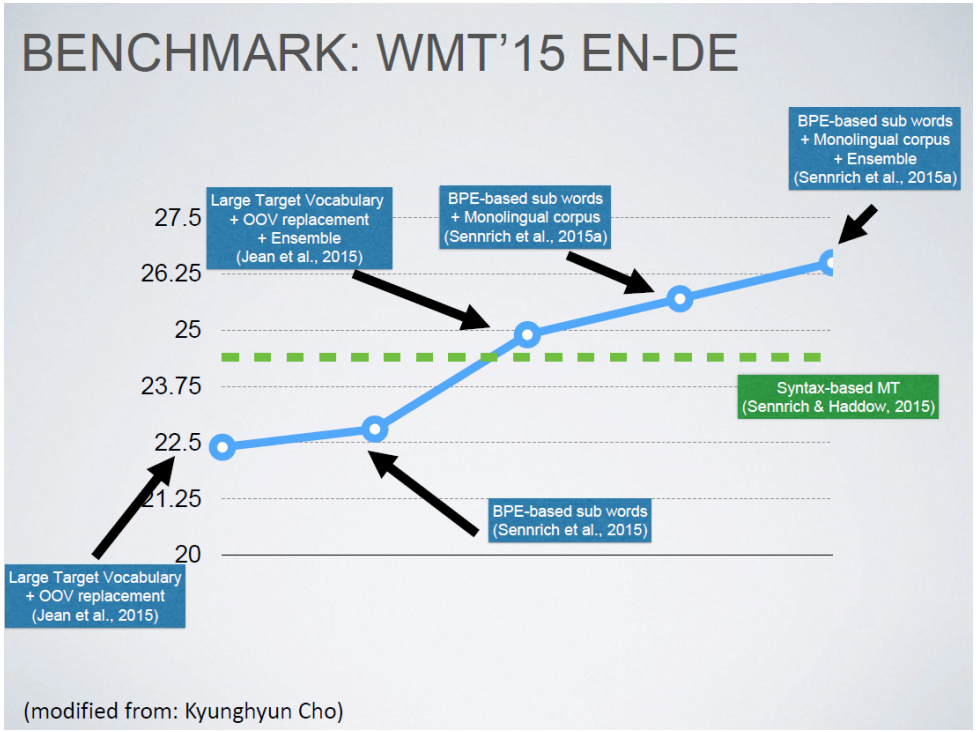
③ Needs some tricks to handle variable lengths of sentences

Convolutional Neural Net (CNN)



Neural Machine Translation (NMT)

- Sequence-to-sequence learning with encoder-decoder framework
- Gated RNN (e.g., LSTM, GRU) as encoder and decoder
- Attention model for automatic alignment
- End-to-end learning
- Other tricks (e.g., large vocabulary etc) for further improvement
- Outperforming STM



Neural Machine Translation (cont'd)

- Comparing with statistical machine translation (SMT)

  SRC: 人类 永久 和平 和 稳定 的 一天 即将 到来 。
  SMT: Mankind permanent peace and stability in the days to come .
  NMT: Peace will soon come in mankind .

  SRC: 欧安 组织 的 成员 包括 北美 、 欧洲 和 亚洲 的 五十五 个 国家 , 明年 将 庆祝 成立 三十 周年 。

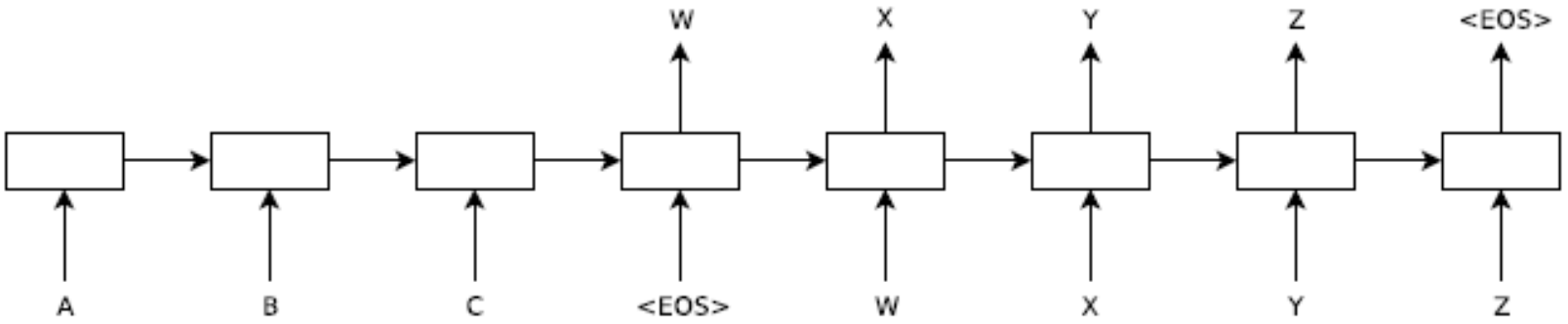
  SMT: the osce members including north america , europe and asia , the 55 countries next year will mark the 30 th anniversary of the establishment .

  NMT: the organization of security includes 55 countries from north america , europe and asia , and next year it will celebrate its 30 th anniversary .

- Better fluency and sentence-level fidelity, catching up on other aspects (e.g., idioms etc)
- Interesting topic: How to combine NMT and SMT

Neural Machine Translation Models

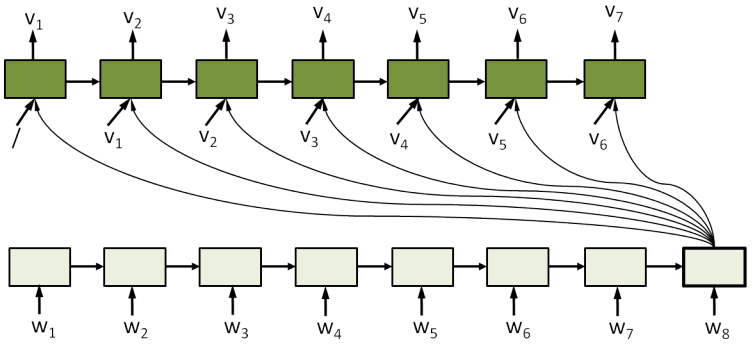
- Machine translation as sequence to sequence learning *(*Sutskever et al., 2014)
- Two LSTMs: one LSTM to encode source sentence, the other LSTM to decode it to target sentence
- Four-layer LSTMs (deep models)
- End-to-end learning



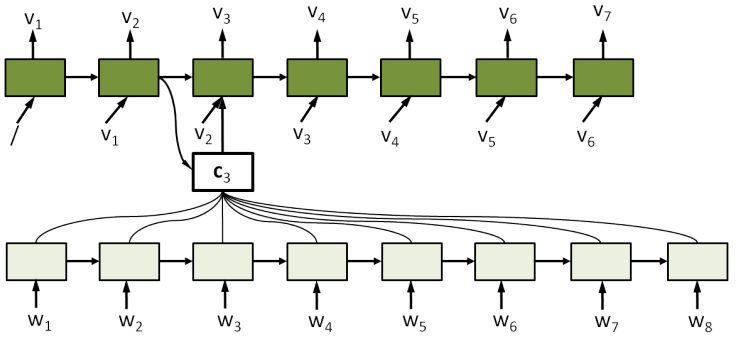
Encoder reads sentence "A B C" in source language and decoder generates sentence "W X Y Z" in target language

Neural Machine Translation Models (cont'd)

- Machine translation using sequence to sequence learning and attention model (Bahdanau et al., 2015)
- RNN encoder decoder framework
- Attention: fixed representation of source sentence → soft and dynamic representation
- End-to-end learning, differentiable data structure



Traditional RNN encoder decoder framework: fixed representation for source sentence



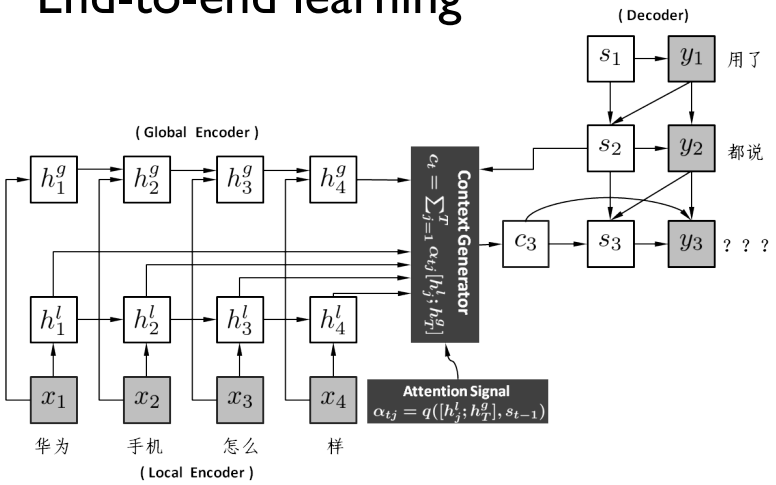
Attention model: mixture of representations of source sentence

Neural Dialogue

- Generation based approach vs retrieval based approach (traditional)
- Single turn dialogue and multi turn dialogue
- Single turn dialogue
  - Sequence-to-sequence learning with encoder-decoder framework
  - 95% of responses are natural sentences, 76% of responses are reasonable replies, trained with social media data (Shang et al. 2015)
- Multi turn dialogue
  - Task specific dialogue
  - Multiple networks, end-to-end learning
  - Success rate is 98% for question answering about restaurants (Wen et al., 2016)
- No human effort, purely data-driven, not possible before

Neural Dialogue Models

- Sequence-to-sequence learning for single-turn dialogue
- Neural Responding Machine (NRM) (Shang et al., 2015)
- Encoder to encode input message, decoder to decode it to output response
- Combination of local context model (attention model) and global context model
- End-to-end learning



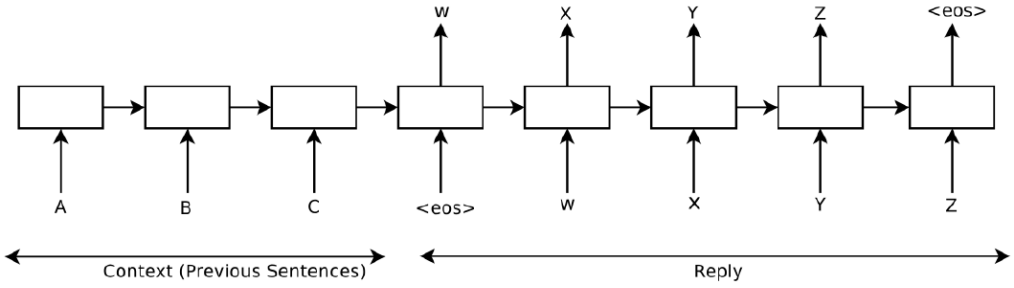
Architecture of NRM, combination of local and global representation

H: 占中终于结束了。	Occupy Central is finally over.
M: 下一个是陆家嘴吧？	Will Lujiazui (finance district in Shanghai) be the next?
H: 我想买三星手机。	I want to buy a Samsung phone
M: 还是支持一下国产的吧。	Why not buy our national brands?

Examples of single turn dialogue between humans and NRM, trained with Weibo data

Neural Dialogue Models (cont'd)

- Sequence-to-sequence learning for multi turn dialogue (Vinyals & Le, 2015)
- Two RNNs: one RNN (encoder) to encode context (previous sentences), the other RNN (decoder) to decode it to response
- End-to-end learning



Sequence to sequence learning: encoding context and generating response

Human: who is skywalker ?
Machine: he is a hero .

Human: what is the color of the sky ?
Machine: blue .

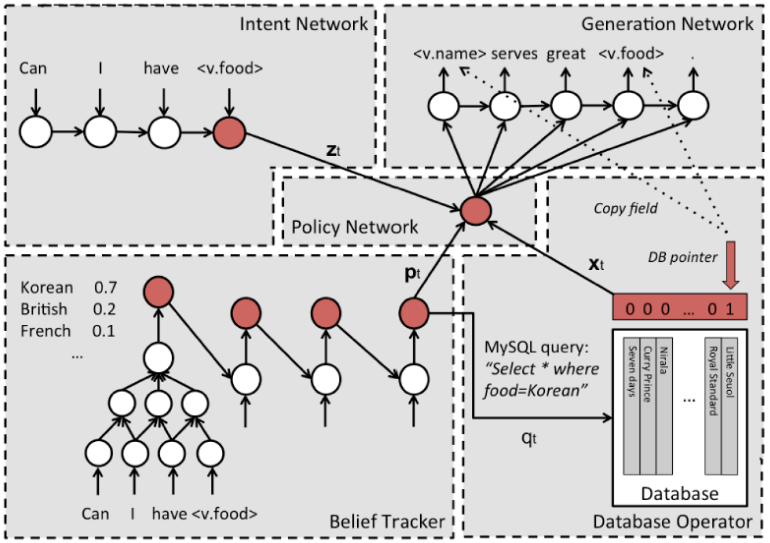
Human: how many legs does a cat have ?
Machine: four , i think .

Human: what is the purpose of living ?
Machine: to live forever

Examples of dialogue between humans and model, trained from movie subtitle data

Neural Dialogue Models (cont'd)

- Task-dependent multi-turn dialogue (Wen et al., 2016), & several other models
- End-to-end learning
- Architecture
  - Intent network, belief state tracker, policy network, generation network
- Complicated task controlled by networks, trained in end-to-end fashion



System architecture

Human	: Hi, I'm hungry for some Indonesian food
Machine	: i am sorry , there are no indonesian restaurants . would you like a different food ?
Human	: Oh no. How about Indian?
Machine	: there are 9 restaurants serving indian types of food . do you have a preference on side of town ?
Human	: East side please
Machine	: sitar tandoori is a indian restaurant in the east side of town . would you like their information ?
Human	: Yes, the address and phone number please
Machine	: sitar tandoori is located at 43 high street cherry hinton cherry hinton and their phone number is 01223 249955 .
Human	: Okay, thank you
Machine	: thank you , goodbye .

Example of multi-turn dialogue between human and system, restaurant domain

Neural Reasoning

- Reasoning over facts
  - Facts and questions are in natural language

bAbi data set, (Weston et al., 2015)

Task 15: Basic Deduction
Sheep are afraid of wolves.
Cats are afraid of dogs.
Mice are afraid of cats.
Gertrude is a sheep.
What is Gertrude afraid of? A:wolves

Task 16: Basic Induction
Lily is a swan.
Lily is white.
Bernhard is green.
Greg is a swan.
What color is Greg? A:white

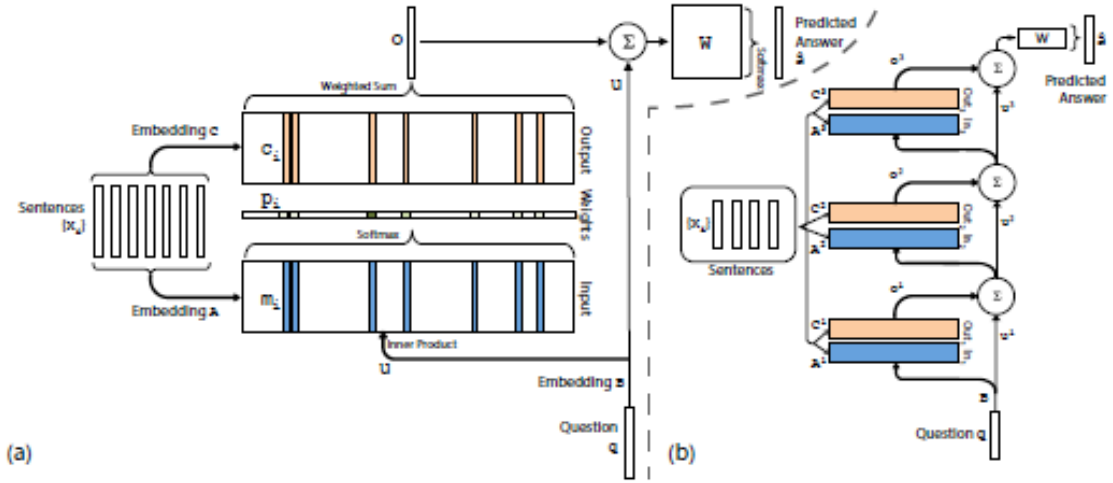
Task 17: Positional Reasoning
The triangle is to the right of the blue square.
The red square is on top of the blue square.
The red sphere is to the right of the blue square.
Is the red sphere to the right of the blue square? A:yes
Is the red square to the left of the triangle? A:yes

Task 18: Size Reasoning
The football fits in the suitcase.
The suitcase fits in the cupboard.
The box is smaller than the football.
Will the box fit in the suitcase? A:yes
Will the cupboard fit in the box? A:no

  - External memory, step-by-step or end-to-end learning
- Natural logic reasoning
  - Learn to conduct natural logic inference
  - Reasoning about semantic relations
  - E.g., from $turtle \prec reptile$, $reptile \prec animal$ to infer $turtle \prec animal$

Neural Reasoning Models

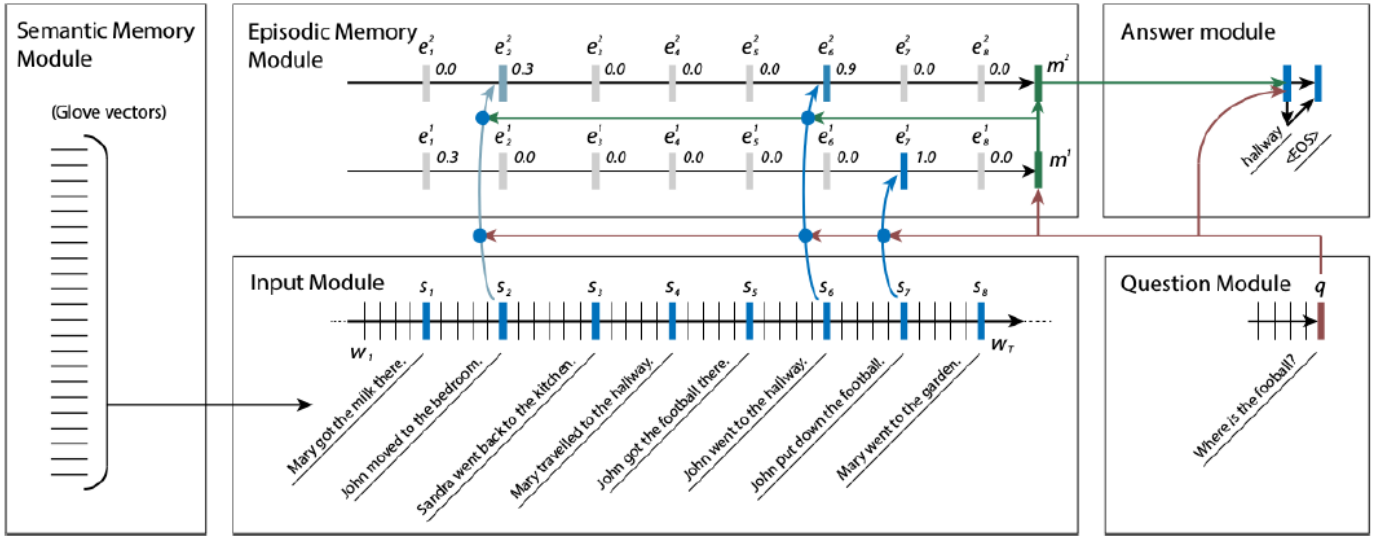
- Memory Networks (Weston et al., 2014, Sukhbaatar et al., 2015)
- Reasoning over facts, language modeling, etc
- Attention model, external memory
- Architecture
  - Multiple layers (steps) of processing
- Write and read intermediate results into external memory, controlled by networks trained in end-to-end fashion



Architecture of Memory Networks, (a) one layer, (b) multiple layers

Neural Reasoning Models (cont'd)

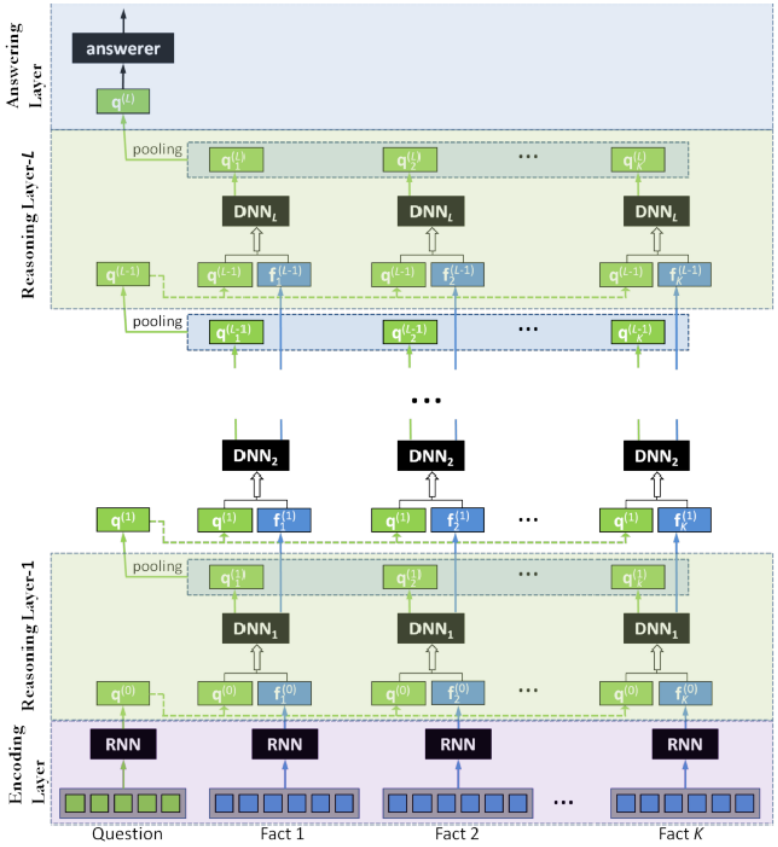
- Dynamic Memory Network: （Kumar et al., 2016）
- External memory, end-to-end learning
- Architecture
  - Input module, question module, answer module, episodic memory module
- Write and read intermediate results into external memory, controlled by networks trained in end-to-end fashion



Architecture of Dynamic Memory Networks

Neural Reasoning Models (cont'd)

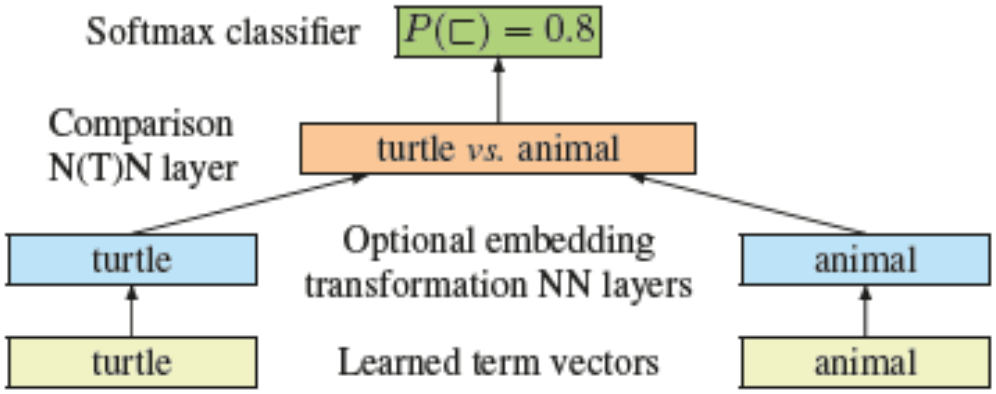
- Reasoning over facts
- Neural Reasoner: (Peng et al. 2015)
- External memory, differentiable data structure, end-to-end learning
- Architecture
  - Encoding layer:
  - Reasoning layers:
  - Answer layer:
- Write and read intermediate results into external memory, controlled by networks trained in end-to-end fashion



Architecture of Neural Reasoner

Neural Reasoning Models (cont'd)

- Model for reasoning using natural logic (Bowman et al., 2015)
- Compare two concepts or terms (distributed representations)
- Model: deep neural network or deep tensor network
- Learn distributed representations of concepts through reasoning with natural logic



Architecture of model

Reference (part I)

- [Graves et al., 2014] Neural Turing Machines. Alex Graves, Greg Wayne, Ivo Danihelka
- [Wen et al., 2016] A Network-based End-to-End Trainable Task-oriented Dialogue System Tsung-HsienWen, Milica Gaši´c, Nikola Mrkši´c, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, David Vandyke, Steve Young
- [Weston et al., 2015] Memory Networks. JasonWeston, Sumit Chopra & Antoine Bordes
- [Shang et al., 2015] Neural responding machine for short-text conversation. Lifeng Shang, Zhengdong Lu, and Hang Li.
- [Vinyals et al., 2014] Sequence to sequence learning with neural networks. Ilya Sutskever, Oriol Vinyals, and Quoc V Le.
- [Vinyals & Le, 2015] A neural conversational model. Oriol Vinyals and Quoc Le.
- [Kumar et al., 2015] Ask Me Anything: Dynamic Memory Networks for Natural Language Processing Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, Richard Socher
- [Sukhbaatar et al., 2015] End-to-end memory networks. S. Sukhbaatar, J. Weston, R. Fergus.
- [Bown et al., 2014] Recursive neural networks can learn logical semantics. S. Bowman, C. Potts, and C. Manning.
- [Weston et al., 2015] Towards ai-complete question answering: A set of prerequisite toy tasks. J. Weston, A. Bordes, S. Chopra, T. Mikolov.
- [Peng et al., 2015] Towards Neural Network-based Reasoning Baolin Peng, Zhengdong Lu, Hang Li, Kam-Fai Wong
- [Ma et al., 2015] Learning to Answer Questions From Image Using Convolutional Neural Network. Lin Ma, Zhengdong Lu, Hang Li
- [Bahdanau et al.2014] Neural machine translation by jointly learning to align and translate. Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio.

Part-II:
Differentiable Data-structures

Differentiable Data-structure: Outline

- What is differentiable data-structure?
- A general formulation
- Addressing strategies
- Memory: types and structures
- Examples
- Concluding remarks

Differentiable Data-structure: Outline

- What is differentiable data-structure?
- A general formulation
- Addressing strategies
- Memory: types and structures
- Examples
- Concluding remarks

What is differentiable data-structure?

- Differentiable data-structure is a memory-like structure which can be controlled by neural network system, with the following properties
  - It can be used to perform rather complicated operations
  - All the operations that can be "tuned", including the read and/or write to the memory, are differentiable

    "so you can just do back-propagations "

- Representative Examples:
  - Neural Turing Machine (a general take)
  - RNNsearch (automatic alignment for M.T.)
  - Memory Network (a different take on memory setting)
  - Neural Random Access Machine (smart design for "pointers")

What is NOT differentiable data-structure?

(Too) many examples

1. Hard attention
   - e.g., hard attention on images (Xu et al. 2015) , for which we have to resort to reinforcement learning or variational methods
2. Varying number of memory cells
   - Typically the number of memory cells are not part of optimization, since they can not be directly optimized via back-propagation
3. Other structural operations
   - e.g,. changing the order of two sub-sequences (Guo 2015), replacing some sub-sequence with others, locating some sub-sequences and store them somewhere etc
4. Other symbolic stuff
   - e.g,. use symbols (discrete classes) in intermediate representations

What is NOT differentiable data-structure?

(Too) many examples

- Hard attention
  - E.g., hard attention on images (Xu et al. 2015) , for which we have to resort to reinforcement learning or variation methods
- Varying number of memory units
  - they

In a sense, the design of differentiable structure is to find a way around, without sacrificing too much efficiency

- Other structural operations
  - E.g,. changing the order of two sub-sequences (Guo 2015), replacing some sub-sequence with others, locating some sub-sequence and store it somewhere etc
- Other symbolic stuff
  - E.g,. use symbols (discrete classes) in intermediate representations

Differentiable Data-structure: Outline

- What is differentiable data-structure?
- A general formulation
- Addressing strategies
- Memory: types and structures
- Examples
- Concluding remarks

Neural Turing Machine (Graves et al., 2014)

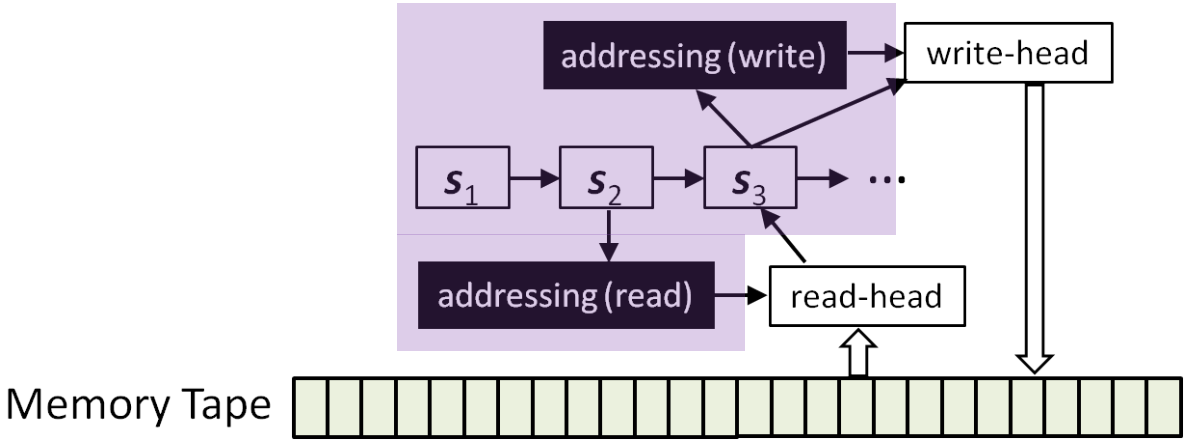
A general formulation:

① **Controller**: typically a gated RNN, controlling the read-write operations

② **Read head**: read the content of the memory at the address given by controller, and return the reading result to the controller

③ **Write head**: write to memory with the content and address both determined by controller

④ **Tape**: the memory, typically a matrix



Neural Turing Machine (Graves et al., 2014)

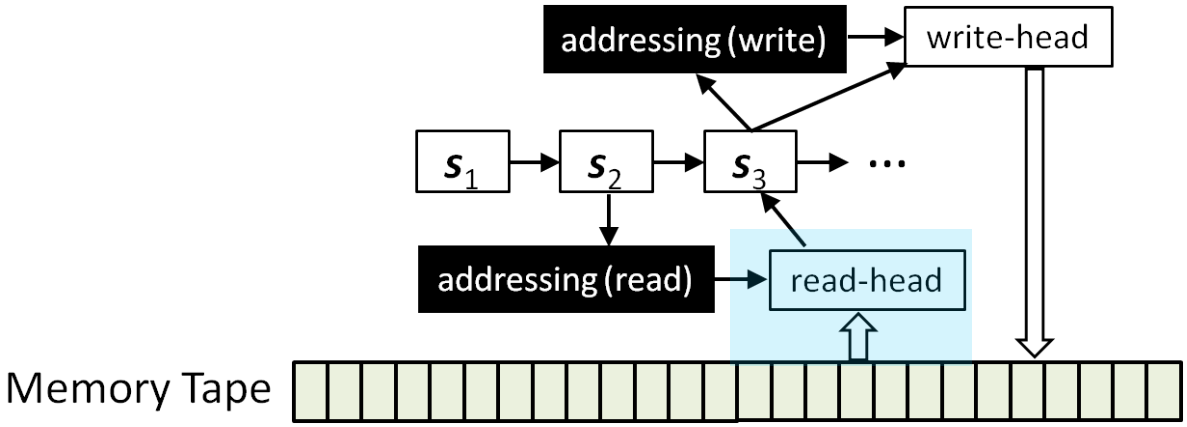
A general formulation:

① **Controller**: typically a gated RNN, controlling the read-write operations

② **Read head**: read the content of the memory at the address given by controller, and return the reading result to the controller

③ **Write head**: write to memory with the content and address both determined by controller

④ **Tape**: the memory, typically a matrix



Neural Turing Machine (Graves et al., 2014)

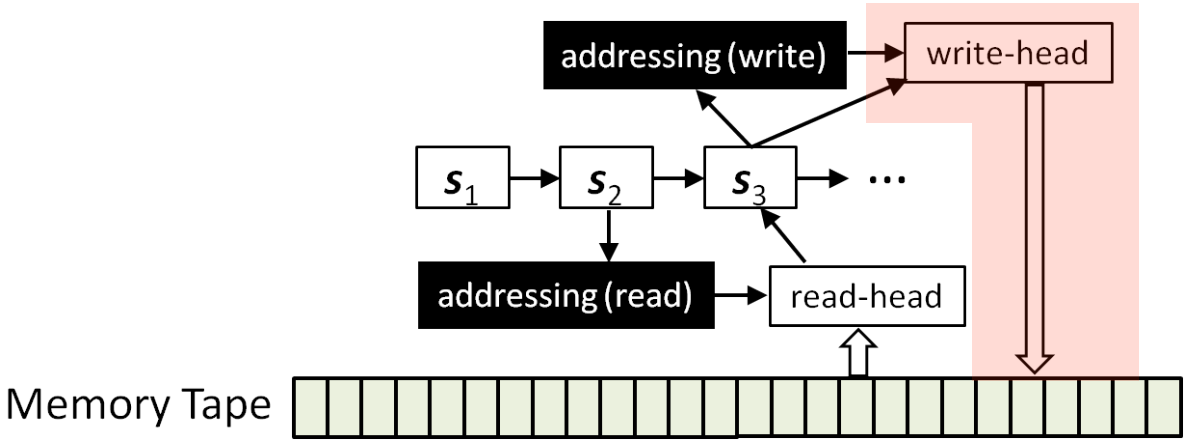
A general formulation:

① **Controller**: typically a gated RNN, controlling the read-write operations

② **Read head**: read the content of the memory at the address given by controller, and return the reading result to the controller

③ **Write head**: write to memory with the content and address both determined by controller

④ **Tape**: the memory, typically a matrix



Neural Turing Machine (Graves et al., 2014)

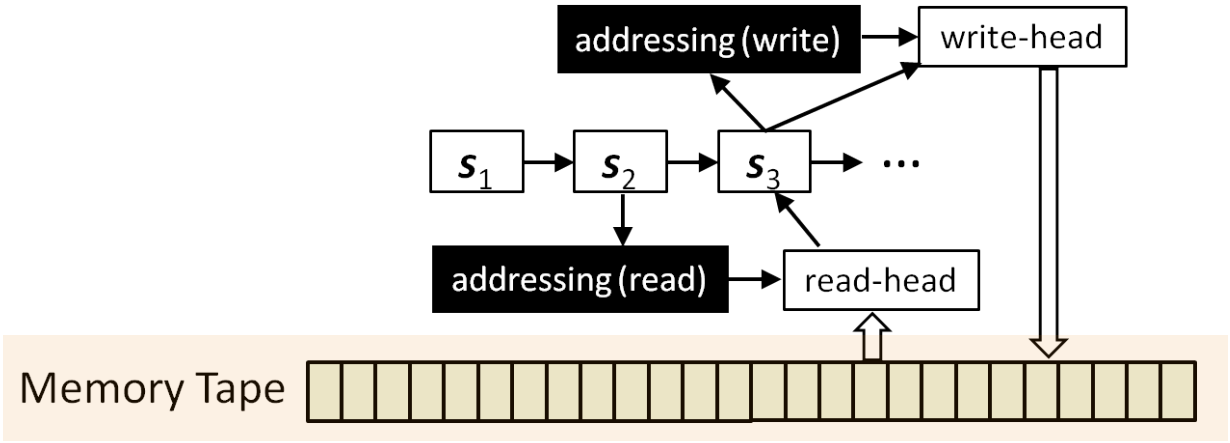
A general formulation:

① **Controller**: typically a gated RNN, controlling the read-write operations

② **Read head**: read the content of the memory at the address given by controller, and return the reading result to the controller

③ **Write head**: write to memory with the content and address both determined by controller

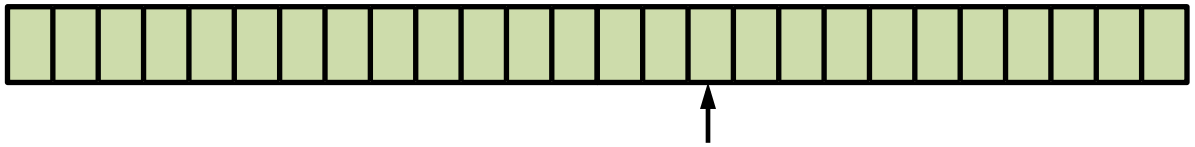
④ **Tape**: the memory, typically a matrix



Hard addressing for reading/writing

Reading

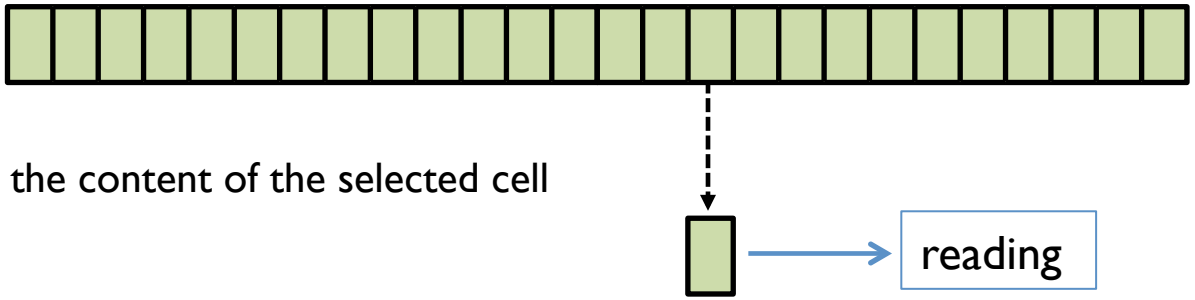
① Determine the cell to read from (addressing)



Hard addressing for reading/writing

Reading

① Determine the cell to read from (addressing)



② Get the content of the selected cell

Hard addressing for reading/writing

Reading

① Determine the cell to read from (addressing)

② Get the content of the selected cell

Writing

Hard addressing for reading/writing

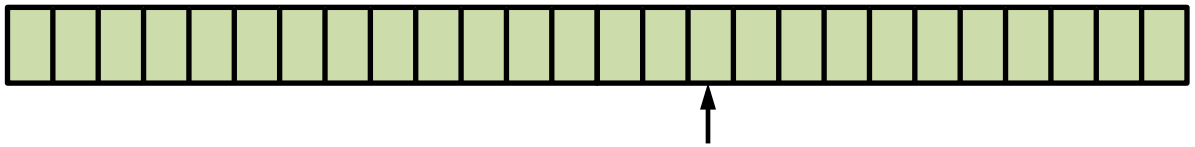
Reading

① Determine the cell to read from (addressing)

② Get the content of the selected cell

Writing

① Determine the cell to write to (addressing)



Hard addressing for reading/writing

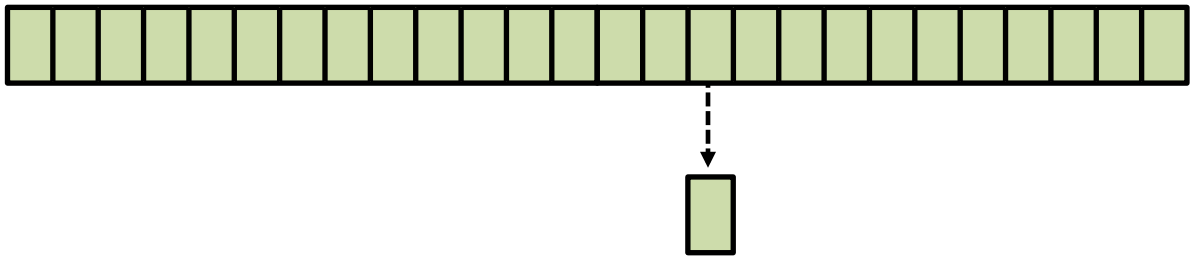
Reading

① Determine the cell to read from (addressing)

② Get the content of the selected cell

Writing

① Determine the cell to write to (addressing)



Hard addressing for reading/writing

Reading

① Determine the cell to read from (addressing)

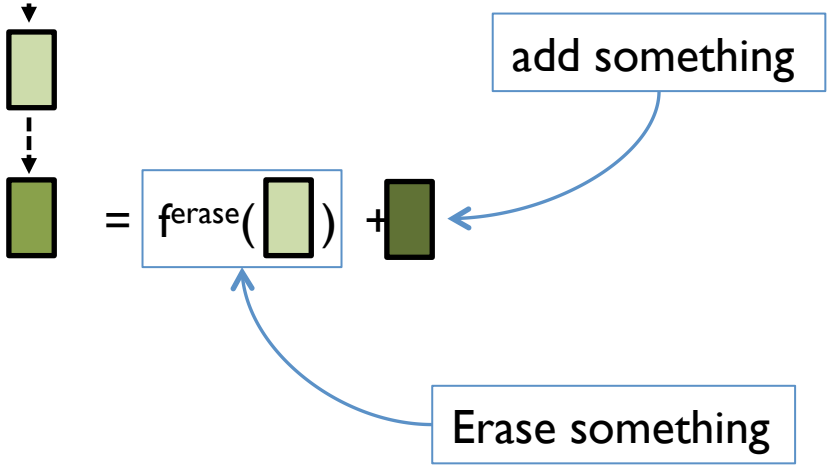
② Get the content of the selected cell

Writing

① Determine the cell to write to (addressing)

② Modify the content of it

- Typically with some forgetting factor



Hard addressing for reading/writing

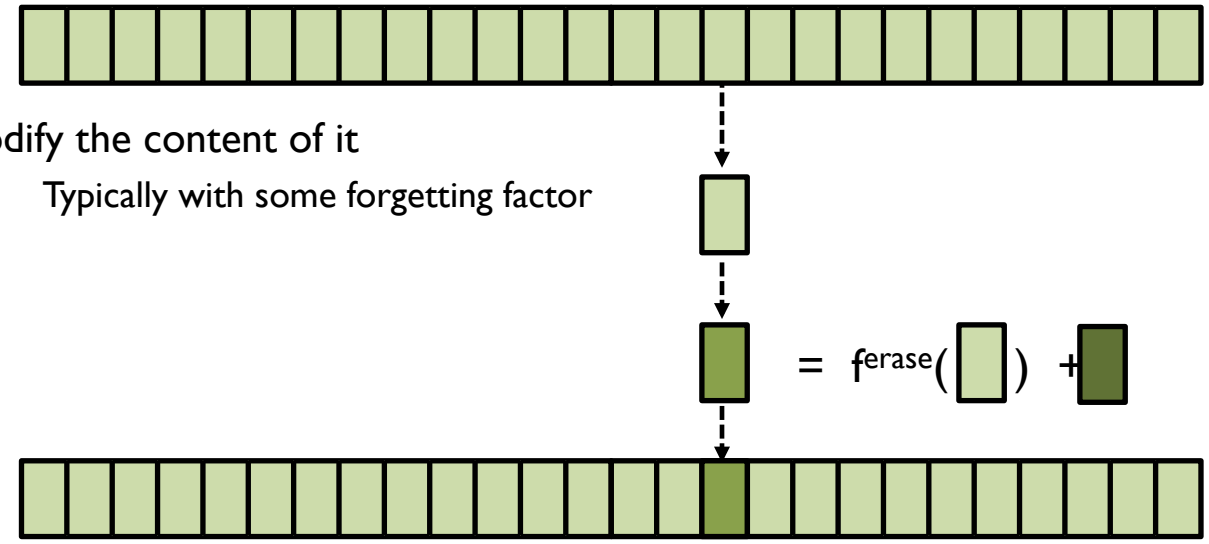
Reading

① Determine the cell to read from (addressing)

② Get the content of the selected cell

Writing

① Determine the cell to write to (addressing)



② Modify the content of it

- Typically with some forgetting factor

Soft addressing for reading/writing

- With soft addressing, the system doesn't pinpoint a memory cell, instead, it gives a distribution over cells it is going to read from or write to.

- So the reading/writing occurs on essentially all memory cells, with different weights (given in the distribution)

- It can be viewed as an expectation (weighted sum) of all the possible read/write actions

The essence of differentiable data-structures

- The essence of differentiable data-structure is to represent the distribution properly, including
  - ❶ every tunable component in the system
  - ❷ joint distribution when multiple components meet

  so the supervision can go through the entire system to
  - ❶ increase the probabilities of the promising candidates,
  - ❷ decrease the probabilities of the poor candidates,

  without killing anyone

- When we have to explicitly model many discrete operations, we often need to generate the **representation** of all of them and do a weighted average

You have to customize your own model

- The generic NTM won't work for most real-world tasks

  "Try a machine translation task, and then you will see"

- You have to design your own model (in a sense, probably a special case of Neural Turing Machine) to put in your domain knowledge
  1. For better inductive bias (so it won't need too many samples to learn)
  2. To better decompose the complicated operations (so each sub-operation can be easily "represented" and "learned")
  3. For better efficiency (so the entire operations take less time)

Differentiable Data-structure: Outline

- What is differentiable data-structure?
- A general formulation
- Memory: types and structures
- Addressing strategies
- Examples
- Concluding remarks

Addressing (for both read and write)

Roughly, three types

- Location-based addressing
  - The controller determines which cells to read/write purely based on the location information
- Content-based addressing
  - The controller determines which cells to read/write based on the content of the cells
- Hybrid addressing
  - The addresses are determined based on both location and content
  - Many different ways to combine the two strategies

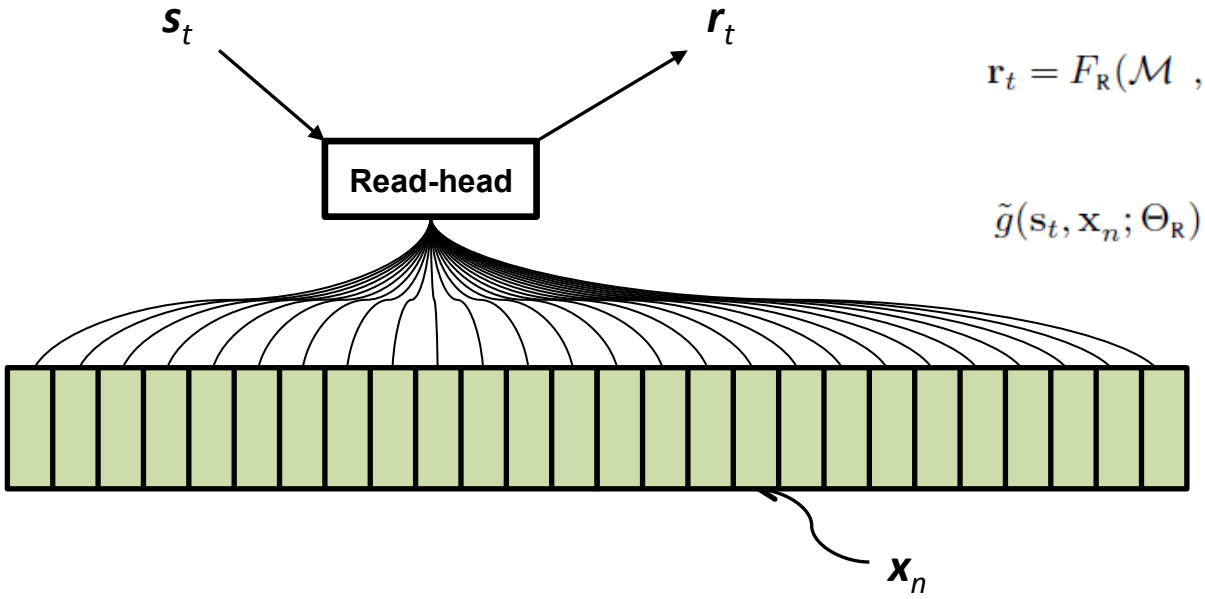
Content-based Addressing

- Content-based addressing determines which cells to read/write based on the state of the controller
- To make it differentiable, the reading is a weighed average of the content of all the cells

s_t : The querying signal at time t

r_t : The reading at time t

x_n : The content of the n^{th} memory cell

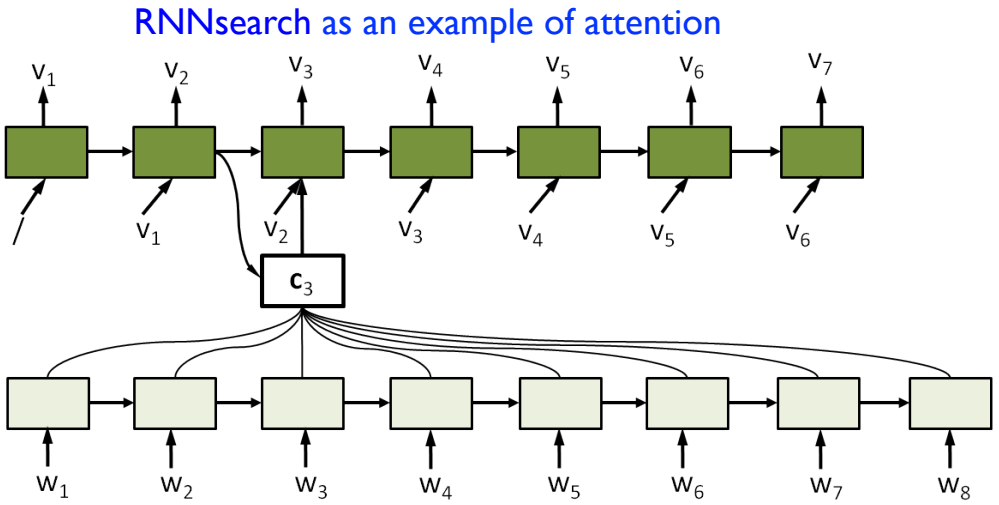


$$\mathbf{r}_t = F_{\mathrm{R}}(\mathcal{M}\ , \mathbf{s}_t; \Theta_{\mathrm{R}}) = \sum_{n=1}^{N_{\mathrm{R}}} \tilde{g}(\mathbf{s}_t, \mathbf{x}_n; \Theta_{\mathrm{R}})\mathbf{x}_n,$$

$$\tilde{g}(\mathbf{s}_t, \mathbf{x}_n; \Theta_{\mathrm{R}}) = \frac{g(\mathbf{s}_t, \mathbf{x}_n; \Theta_{\mathrm{R}})}{\sum_{n'=1}^{N_{\mathrm{R}}} g(\mathbf{s}_t, \mathbf{x}_{n'}; \Theta_{\mathrm{R}})},$$

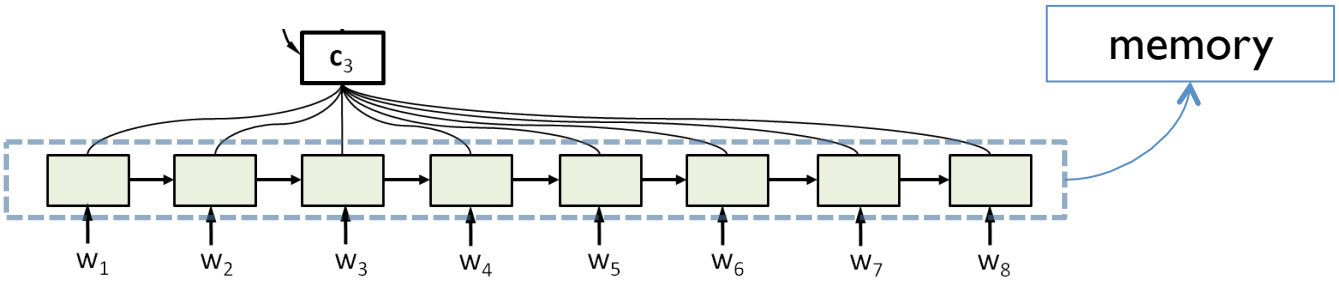
Attention is also addressing

- Roughly speaking, attention is just a way to dynamically determine which memory cells to read



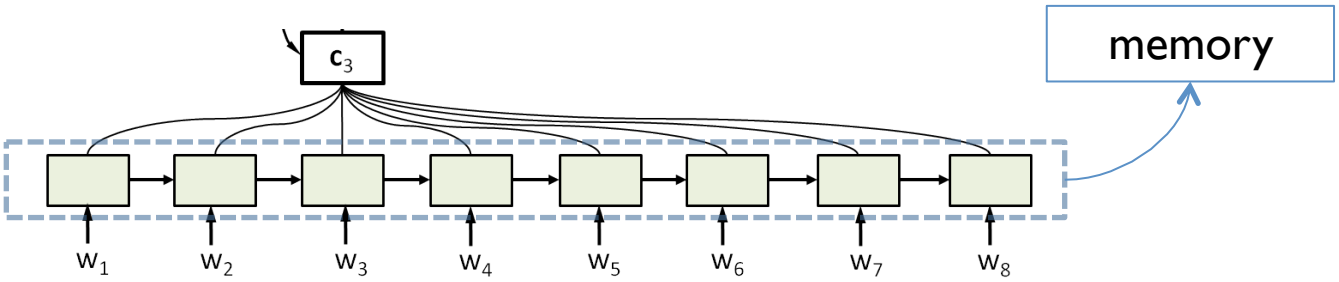
Attention is also addressing

- Roughly speaking, attention is just a way to dynamically determine which memory cells to read



Attention is also addressing

- Roughly speaking, attention is just a way to dynamically determine which memory cells to read



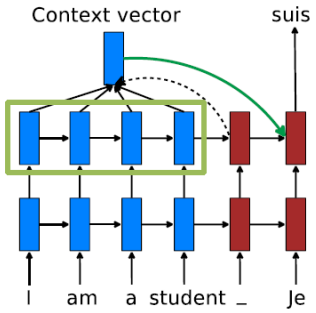
- Attentive read could be based on content, location (Graves, 2012), or both

More complicated attention mechanism

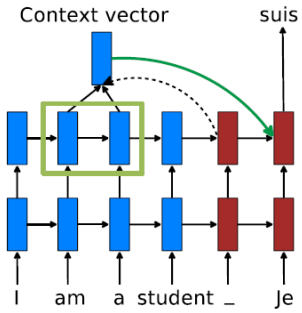
Local+global attention

① Multiple attention strategy to find the "context" vector with different resolutions

② Often outperforms just the global one



Global: ***all*** source states.



Local: ***subset*** of source states.

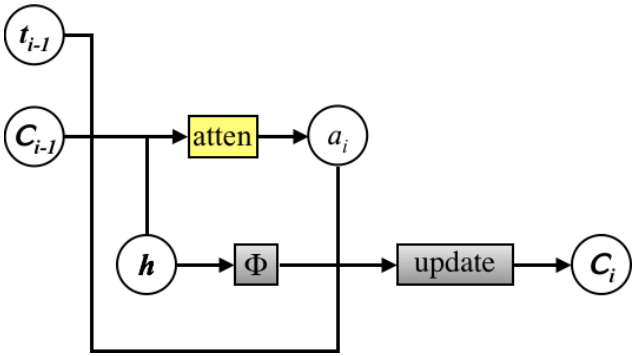
Luong et al. (2015)

Modeling "coverage"

① Maintain an extra "coverage" vector on memory cells to record the history of "attentive read"

② The model will learn to read and update the coverage vector to coordinate the attentive reading

③ Help to handle under-translation and over-translation in NMT



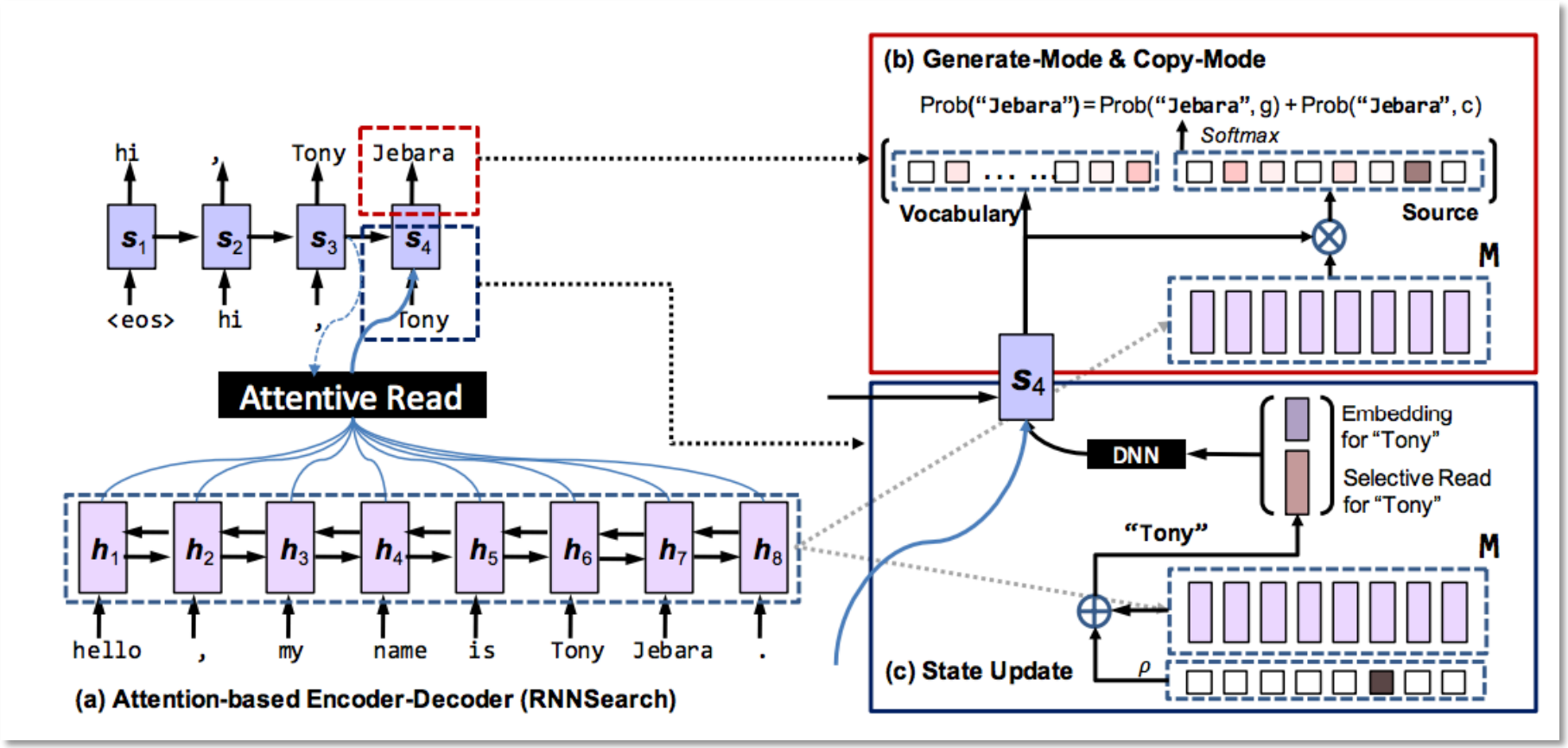
Tu et al. (2016)

Location-based Addressing

- If location-based addressing is part of the optimization, it is likely to be non-differentiable. There are three strategies to get around
  - **Strategy-I**: the location-based part is hard-wired into the hybrid addressing, e.g. in the generic NTM (Graves et al., 2014)
  - **Strategy-II**: location−based addressing is part of hard−wired operations which are controlled by the neural network, e.g, Neural Random Access Machines (Kurach et al., 2015)
  - **Strategy-III**: both the architecture and the learning setting are designed to encourage location-based addressing, e.g., CopyNet (Gu et al., 2016)

Location-based Addressing: CopyNet (Gu et al., 2016)

- Sequence-to-sequence model with two attention mechanisms
- **The encoder** is encouraged to put location information in the content of memory, while t**he decoder** is encouraged to uses this location information for "copying" segments of source



Differentiable Data-structure: Outline

- What is differentiable data-structure?
- A general formulation
- Memory: types and structures
- Addressing strategies
- Examples
- Concluding remarks

In terms of "term"

A very sloppy categorization

- Short-term memory

  Short-term memory (STM) stores the representation of current input, e.g., the representation of source sentence in neural machine translation

- Intermediate memory

  Intermediate memory lies between short-term and long-term memory, it stores the facts and contextual information related to the current problem.

- Long-term memory

  Long-term memory (LTM) stores the instance-independent knowledge, for example the factoid knowledge in Q.A., or the rule memory for translation in M.T.

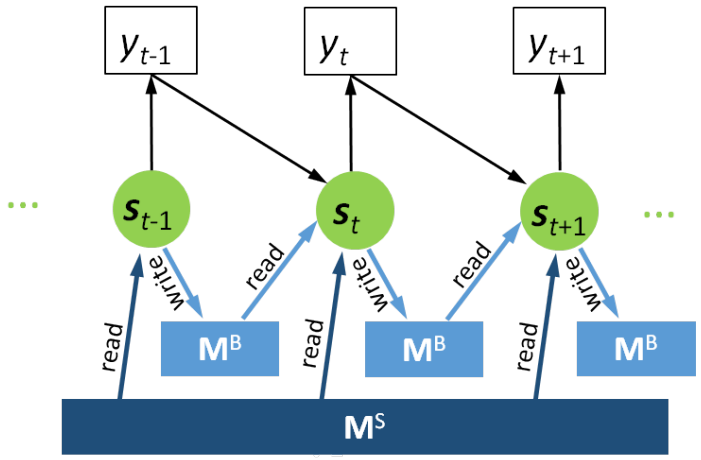
Examples about terms of memories (1)

"Short Short-term Memory" in NMT

① Memory as a direct extension to the state of decoder RNN, big improvement of performance

② This memory is read from and written to by the decoder RNN at each time step

③ It stores part of source and target sentences relevant for a particular time in decoding

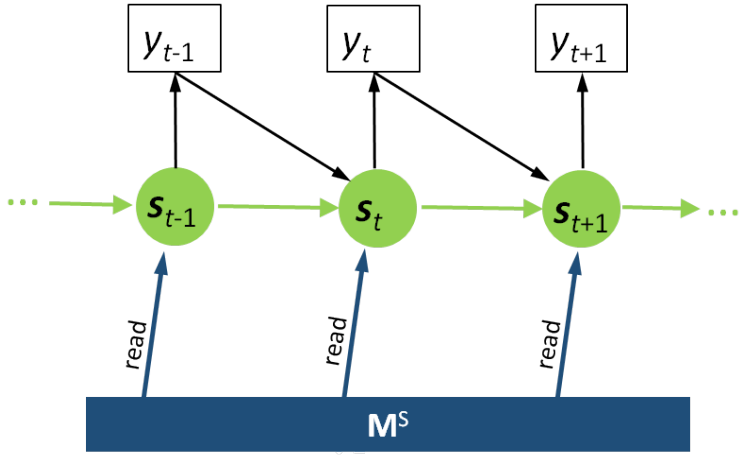


Wang et al. (2016)

Short-term Memory in NMT

① The model will learn to form the source memory in the encoding phase, to read from it ("attentive read") during decoding, and sometimes write a little bit too

② Now a common practice in NMT



Bahdanau et al. (2015)
and many many more

Examples about terms of memories (2)

Intermediate Memory for Representing A Dialog Session, A Paragraph …

① E.g, save the dialog history in a memory-net and attend to the relevant part when generating a response

bot: which price range are you looking for?

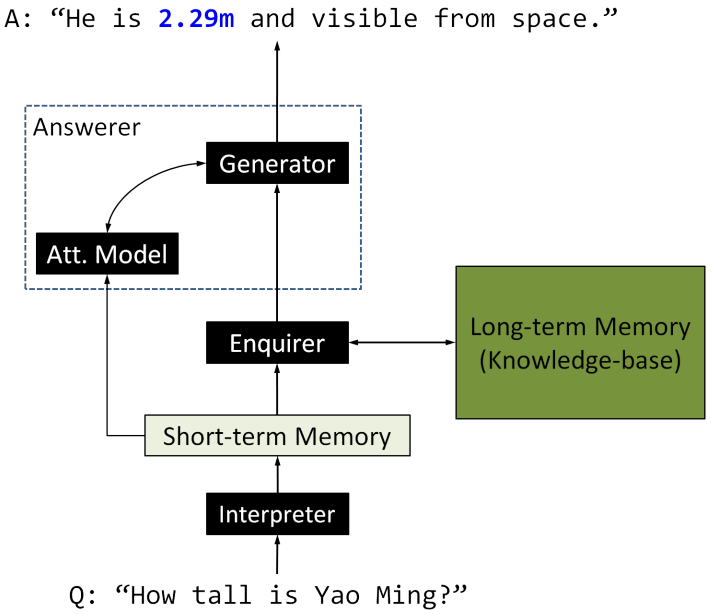
user: hi
bot: hello what can I help you with today
user: may I have a table in Paris
bot: I am no it
user: <silence>
bot: any preference on a type of cuisine
user: I love Indian food
bot: how many people would be in your part?
user: we will be six

Bordes & Weston et al. (2016)

Long-term Memory in Neural QA

① Have a memory to save the factoid knowledge (tables, triples)

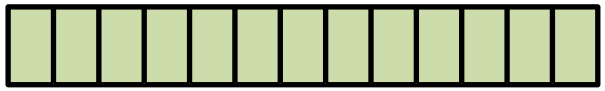
② An generative model will "fetch" the knowledge from the LTM as needed



Yin et al. (2016a)

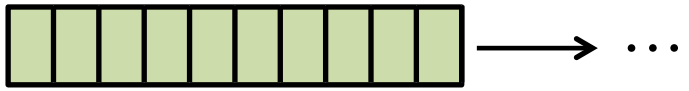
In term of "structure"

- Pre-determined Size (generic NTM)



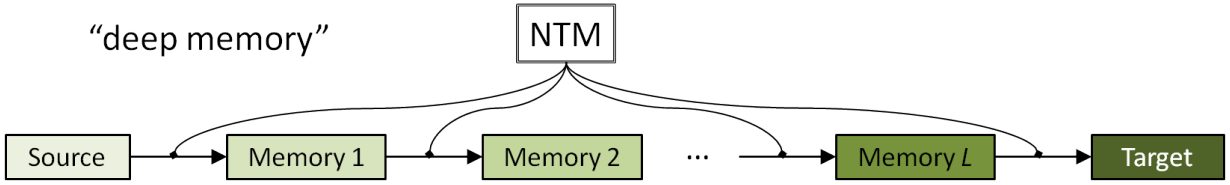
  - Memory of fixed size is claimed (independent of instances) and the read/write operation is on the entire memory

- Linear (Neural Stack, Neural Queue)



  - The number of memory "cells" is linear to the length of sequence

- Stacked-Memory (DeepMemory)



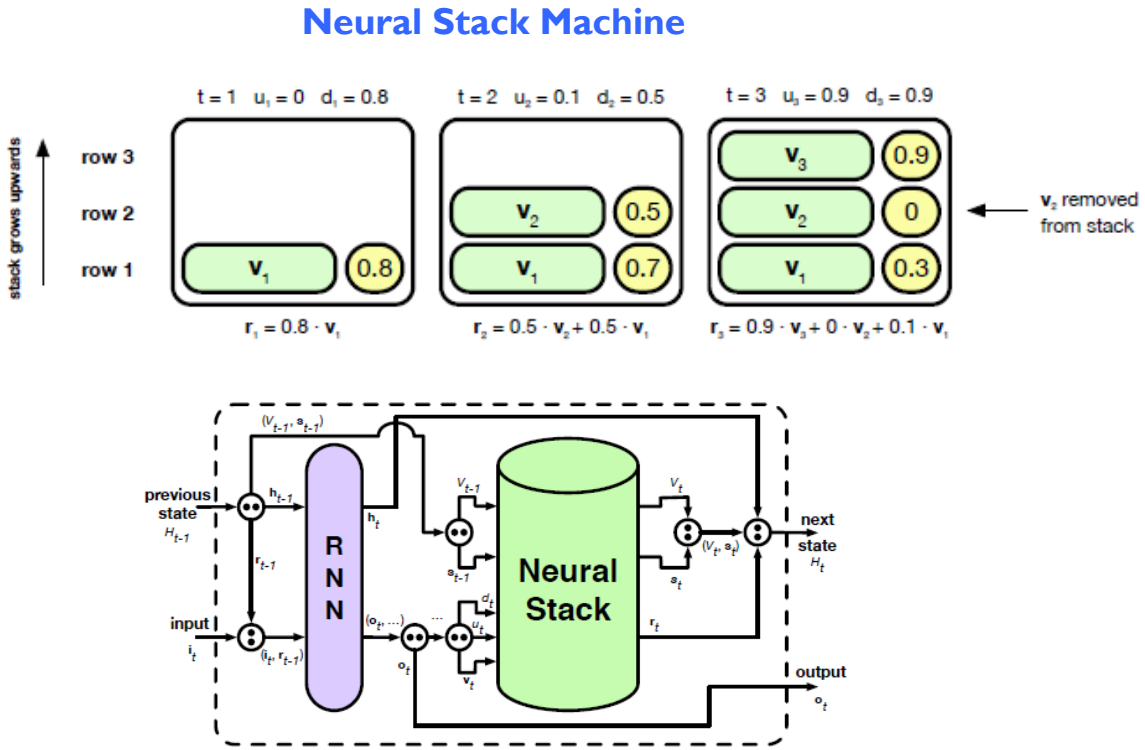
  - It could be based on memory with fixed or linear size

Linear Memory

The number of memory cells is linear to the length of sequence

❶ It is **unbounded,** but the number of cells can often be pre-determined (e.g., in translation, after you see the entire source sentence)

❷ Can take the form queue, stack…, for different modeling tasks

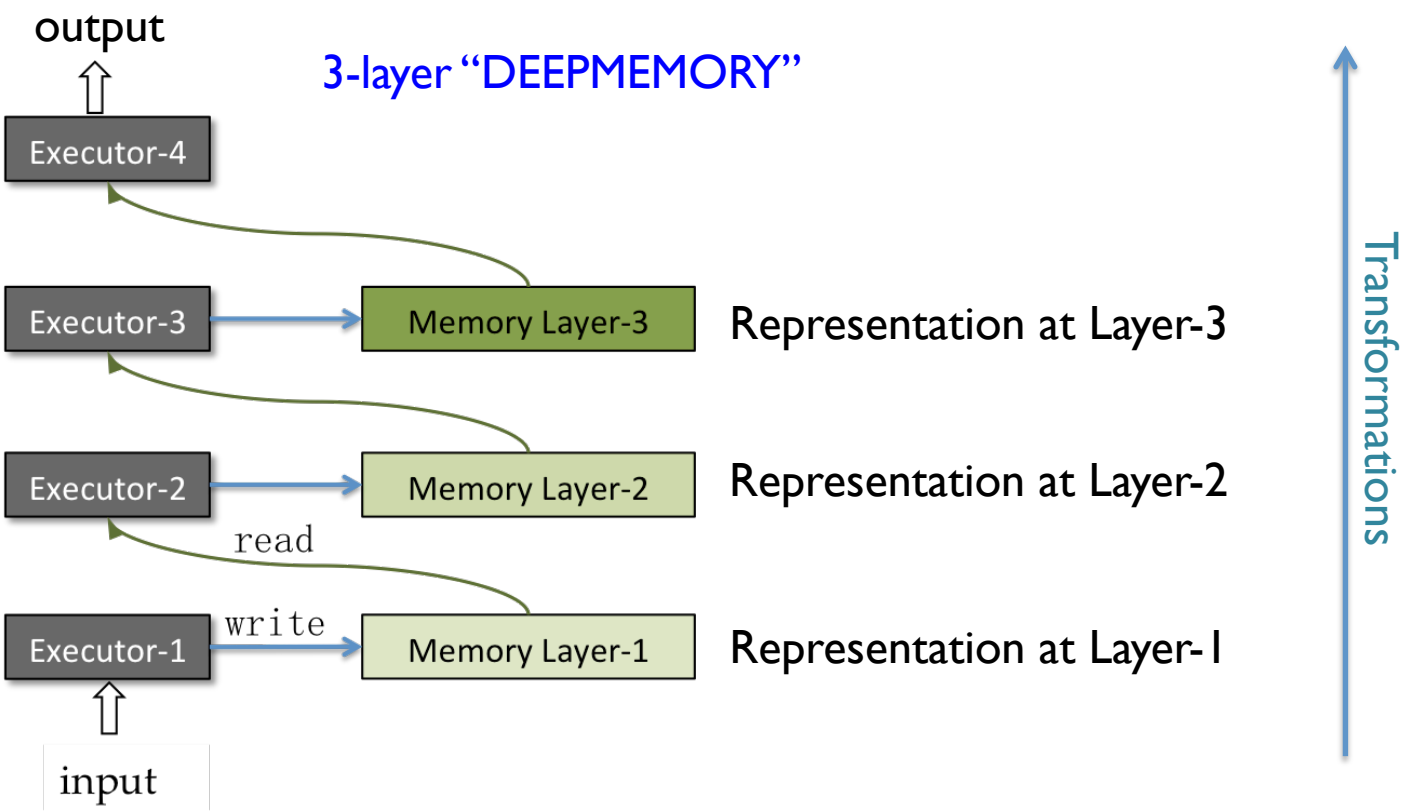


Grefenstette et al. (2015)

"Deep Memory"

Deep memory-based architecture for NLP

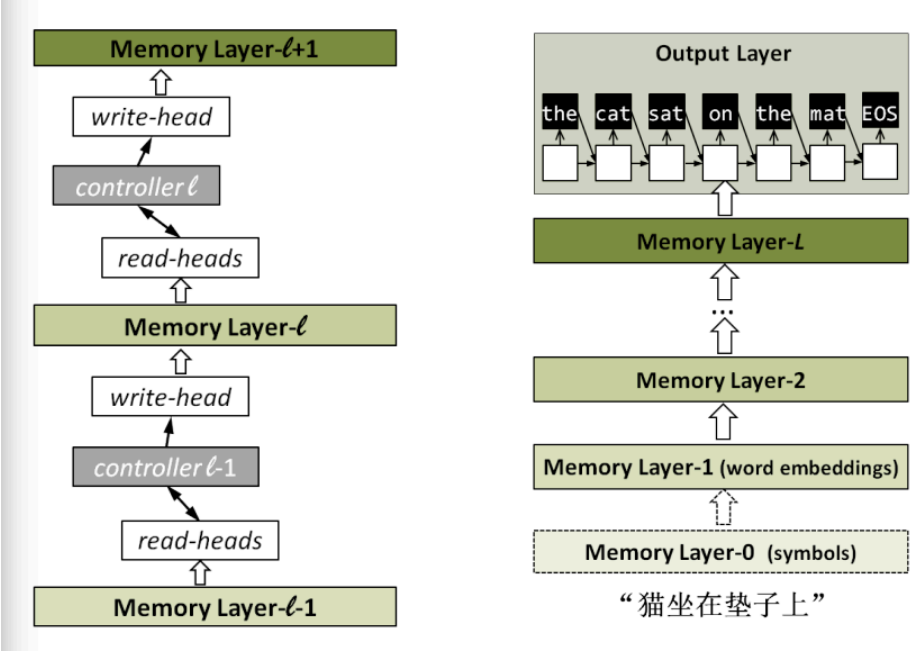
1. Different layers of memory, equipped with suitable read/write strategies to encourage layer by layer transformation of input (e.g, a sentence)
2. A generalization of the deep architectures in DNN to richer form of representations to handle more complicated linguistic objects



"Deep Memory" (cont'd)

- Powerful architecture for NLP tasks that require complicated transformations from input to output
- Has been applied to many tasks with success
  - Machine translation (Meng et al., 2015)
  - Querying knowledge-bases (Yin et al., 2016a)
  - Reasoning (Peng et al., 2015)

"DEEPMEMORY" for Machine Translation



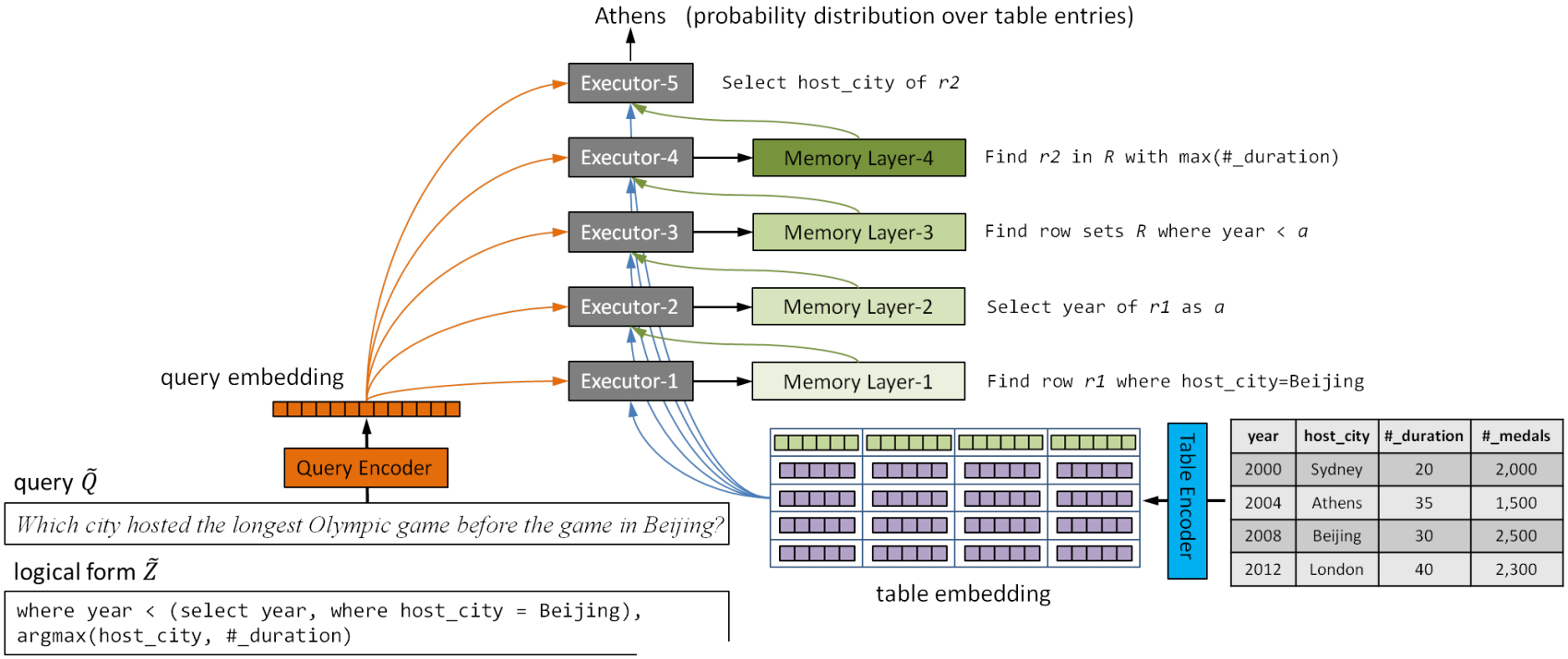
Meng et al. (2015)

Differentiable Data-structure: Outline

- What is differentiable data-structure?
- A general formulation
- Addressing strategies
- Memory: types and structures
- Examples
- Concluding Remarks

Neural Enquirer: both STM and LTM

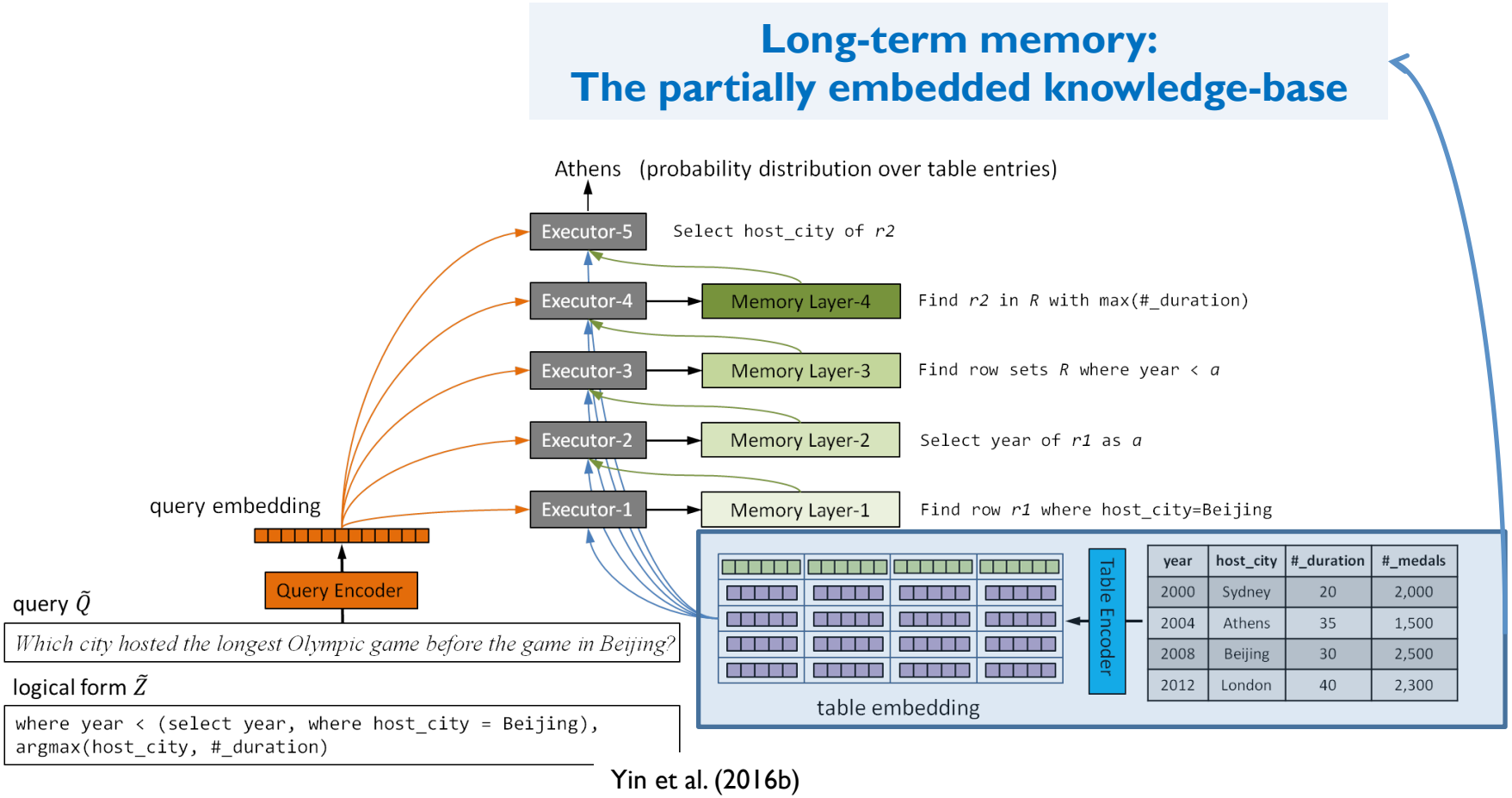
- A neural network system for querying KB tables
- Neural Enquirer has both short-term and long-term memory



Yin et al. (2016b)

Neural Enquirer: both STM and LTM

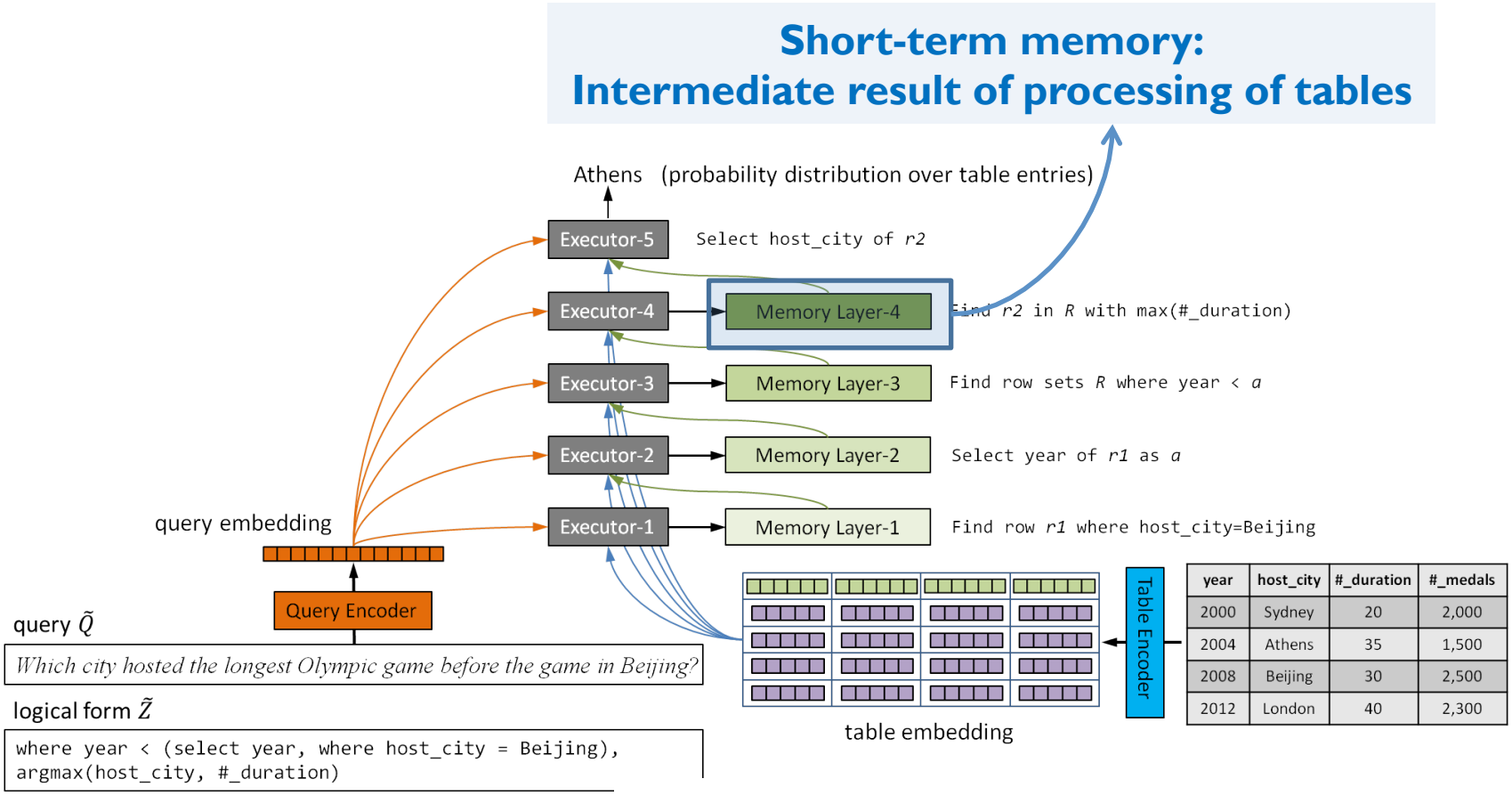
- A neural network system for querying KB tables
- Neural Enquirer has both short-term and long-term memory



Yin et al. (2016b)

Neural Enquirer: both STM and LTM

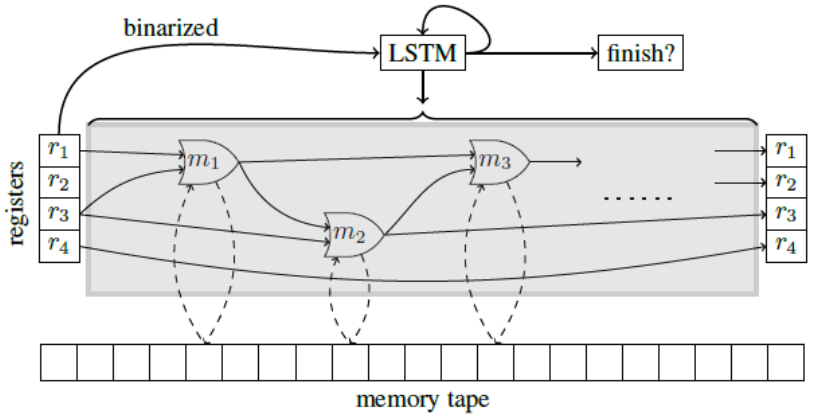
- A neural network system for querying KB tables
- Neural Enquirer has both short-term and long-term memory



Yin et al. (2016b)

Neural Random Access Machine

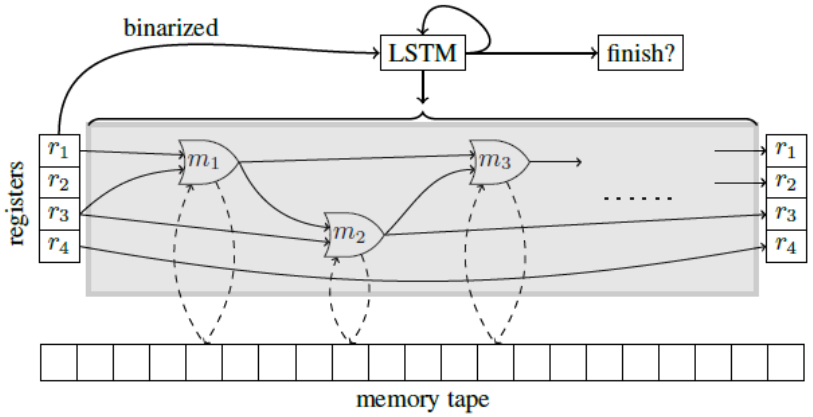
- Neural random access machine is special case of NTM that supports pointer in a differentiable data-structure
- An intriguing example to use differentiable data-structure for seemingly non-differentiable operations
  - Many hard modules to access the memory and soft mechanism (a special kind of NN controller) to call them



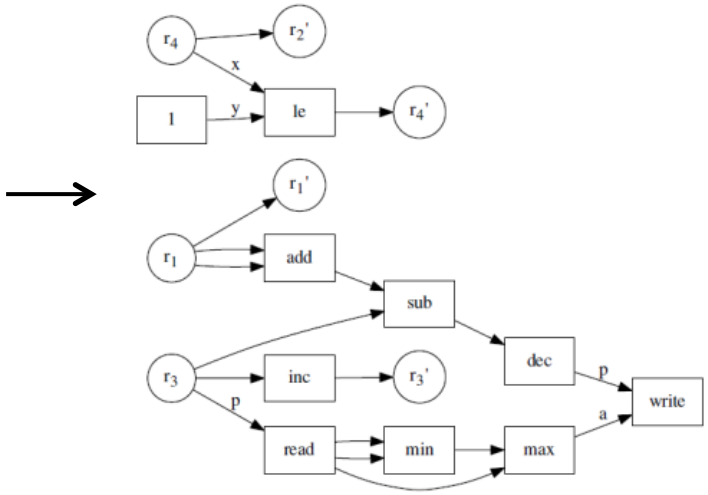
Kurach et al. (2016)

Neural Random Access Machine

- Neural random access machine is special case of NTM that supports pointer in a differentiable data-structure
- An intriguing example to use differentiable data-structure for seemingly non-differentiable operations
  - Many hard modules to access the memory and soft mechanism (a special kind of NN controller) to call them



Kurach et al. (2016)



The circuit generated at every time step >2 for the task **Reverse**

Differentiable Data-structure: Outline

- What is differentiable data-structure?
- A general formulation
- Addressing strategies
- Memory: types and structures
- Examples
- Concluding Remarks

Pros and Cons

Differentiability requires maintaining the entire distribution, which has its advantages and disadvantages

Pros:

1. it makes the optimization straightforward and efficient, since every member gets a non-zero share of the mass (vs. non-differentiable cases)
2. Memory and all that give great space for architectural and mechanism design

Cons:

1. Maintaining this distribution and properly representing it is not always easy
2. Dropping the differentiability requirement often makes the design (for example the pointer) much easier

Reference (part II)

- [Xu et al.,2015] Show, Attend and Tell: Neural Image Caption Generation with Visual Attention
  Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, Yoshua Bengio
- [Graves et al., 2014] Neural Turing Machines. Alex Graves, Greg Wayne, Ivo Danihelka
- [Grefenstette et al., 2015] Learning to Transduce with Unbounded Memory. Edward Grefenstette, Karl Moritz Hermann, Mustafa Suleyman, Phil Blunsom
- [Gu, 2016] Incorporating Copying Mechanism in Sequence-to-Sequence Learning. Jiatao Gu, Zhengdong Lu, Hang Li, Victor O.K. Li
- [Kurachet al., 2015] Neural Random-Access Machines Karol Kurach, Marcin Andrychowicz, Ilya Sutskever
- [Wang et al., 2016] Memory-enhanced Decoder for Neural Machine Translation Mingxuan Wang, Zhengdong Lu, Hang Li, Qun Liu
- [Yin et al., 2016a] Neural Generative Question Answering Jun Yin, Xin Jiang, Zhengdong Lu, Lifeng Shang, Hang Li, Xiaoming Li
- [Yin et al., 2016b] Neural Enquirer: Learning to Query Tables with Natural Language. Pengcheng Yin, Zhengdong Lu, Hang Li, Ben Kao
- [Weston et al., 2015] Memory Networks. Jason Weston, Sumit Chopra & Antoine Bordes [Hochreiter & Schmidhuber, 1997] Long short-term memory. Sepp Hochreiter and Jurgen Schmidhuber.
- [Shang et al., 2015] Neural responding machine for short-text conversation. Lifeng Shang, Zhengdong Lu, and Hang Li.
- [Vinyals et al., 2014] Sequence to sequence learning with neural networks. Ilya Sutskever, Oriol Vinyals, and Quoc V Le.
- [Vinyals et al., 2015] Pointer networks. Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly.
- [Tu et al., 2016] Modeling Coverage for Neural Machine Translation Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, Hang Li
- [Sukhbaatar et al., 2015] End-to-end memory networks. S. Sukhbaatar, J. Weston, R. Fergus.
- [Peng et al., 2015] Towards Neural Network-based Reasoning Baolin Peng, Zhengdong Lu, Hang Li, Kam-Fai Wong
- [Meng et al., 2015] A Deep Memory-based Architecture for Sequence-to-Sequence Learning Fandong Meng, Zhengdong Lu, Zhaopeng Tu, Hang Li, Qun Liu
- [Luong et al. 2015] Effective approaches to attention-based neural machine translation. Minh-Thang Luong, Hieu Pham, and Christopher D Manning.
- [Bahdanau et al.2014] Neural machine translation by jointly learning to align and translate. Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio.

Part-III:
Learning Paradigms

Learning: Outline

- Overview
- End-to-end learning (or not?)
- Dealing with non-differentiability
- Grounding-based learning

Human language learning

It is a complex (and powerful) mixture of

- Supervised learning:
  - when we are taught words and grammar
  - when we got corrected in making a sentence
- Unsupervised learning:
  - when we learn French by reading a French novel
  - when we figure out the meaning of words by seeing how they are used
- Reinforcement learning:
  - when we are learning through "trial and error"
- Explanation-based learning:
  - when we are building a theory based on our domain knowledge to make sense of a new observation

...

Several dimensions of learning paradigm

- End-to-end vs. "step-by-step"
- Gradient descent vs. Non-differentiable objectives
- Supervision from "grounding" or human labeling
- Supervised learning vs. Reinforcement learning

Several dimensions of learning paradigm

- End-to-end vs. "step-by-step"

  End-to-end learning tunes the parameters of the entire model based on the correctional signal from the output. No supervision added to the intermediate layers

  In **Step-by-step** learning, we have **specifically designed** supervision on the intermediate representations

- Gradient-based vs. Non-differentiable objectives
- Supervision from "grounding" or human labeling
- Supervised learning vs. Reinforcement learning

Several dimensions of learning paradigm

- End-to-end vs. "step-by-step"
- Gradient-based vs. Non-differentiable objectives

  Gradient-based learning tunes the parameters by minimizing the differentiable cost and back-propagating the gradient to the other part of the network

  If the objective is **non-differentiable**, we have to resort to some smart sampling methods (e.g., RL) or some smart approximation methods (e.g., MRT)

- Supervision from "grounding" or human labeling
- Supervised learning vs. Reinforcement learning

Several dimensions of learning paradigm

- End-to-end vs. "step-by-step"
- Gradient-based vs. Non-differentiable objectives
- Supervision from "grounding" or human labeling

Grounding-based learning seeks supervision from the interaction between the model and the environment in real-world

Human labeling gives the "truth" or human expectation on the aspect of the linguistic objects that is of interest

- Supervised learning vs. Reinforcement learning

Several dimensions of learning paradigm

- End-to-end vs. "step-by-step"
- Gradient-based vs. Non-differentiable objectives
- Supervision from "grounding" or human labeling
- Supervised learning vs. Reinforcement learning

Supervised learning relies on the label information and propagates the correction signal to the neural net to tune the parameters

Reinforcement learning gets potentially delayed rewards for the decision of the model and learn through maximizing the expected rewards

Several dimensions of learning paradigm

- End-to-end vs. "step-by-step"
- Gradient-based vs. Non-differentiable objectives
- Supervision from "grounding" or human labeling
- Supervised learning vs. Reinforcement learning

They are orthogonal to each other, e.g, a NMT algorithm could be

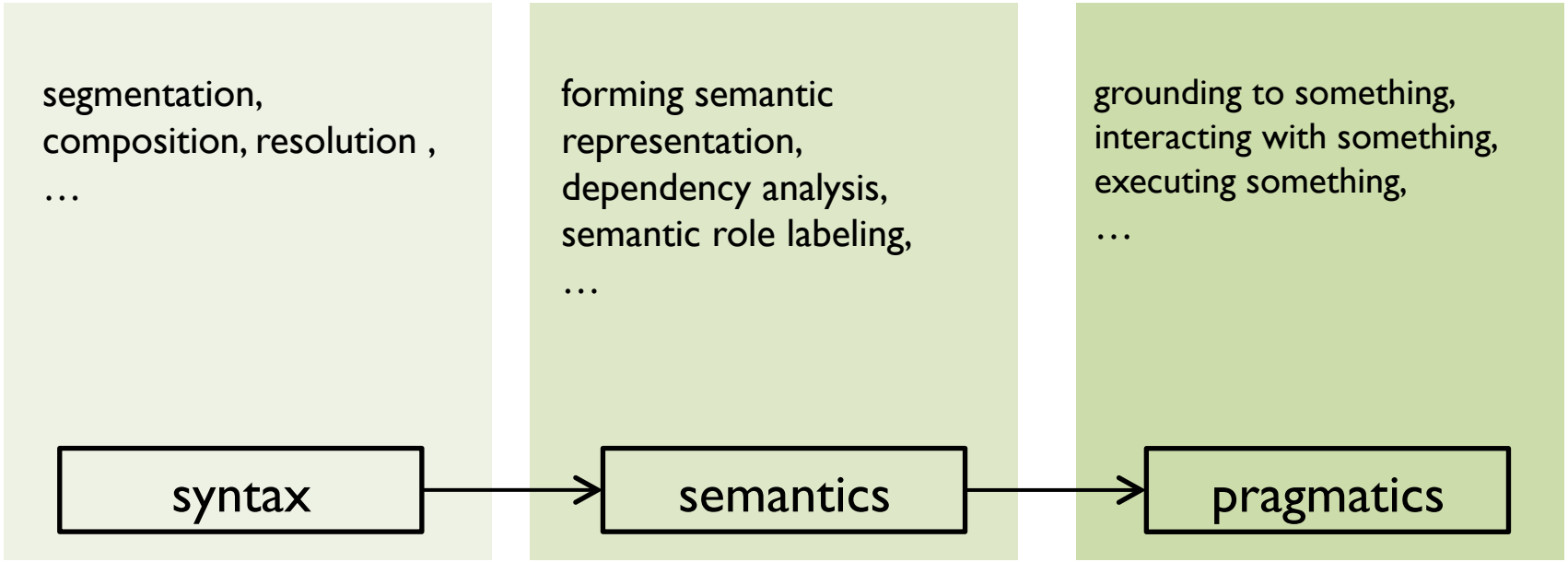
- End-to-end learning: sequence-to-sequence learning
- Non-differentiable objective: BLEU
- Human labeling: human given reference
- Reinforcement learning: decoding as sequential decisions

Learning: Outline

- Overview
- End-to-end learning (or not?)
- Dealing with non-differentiability
- Grounding-based learning

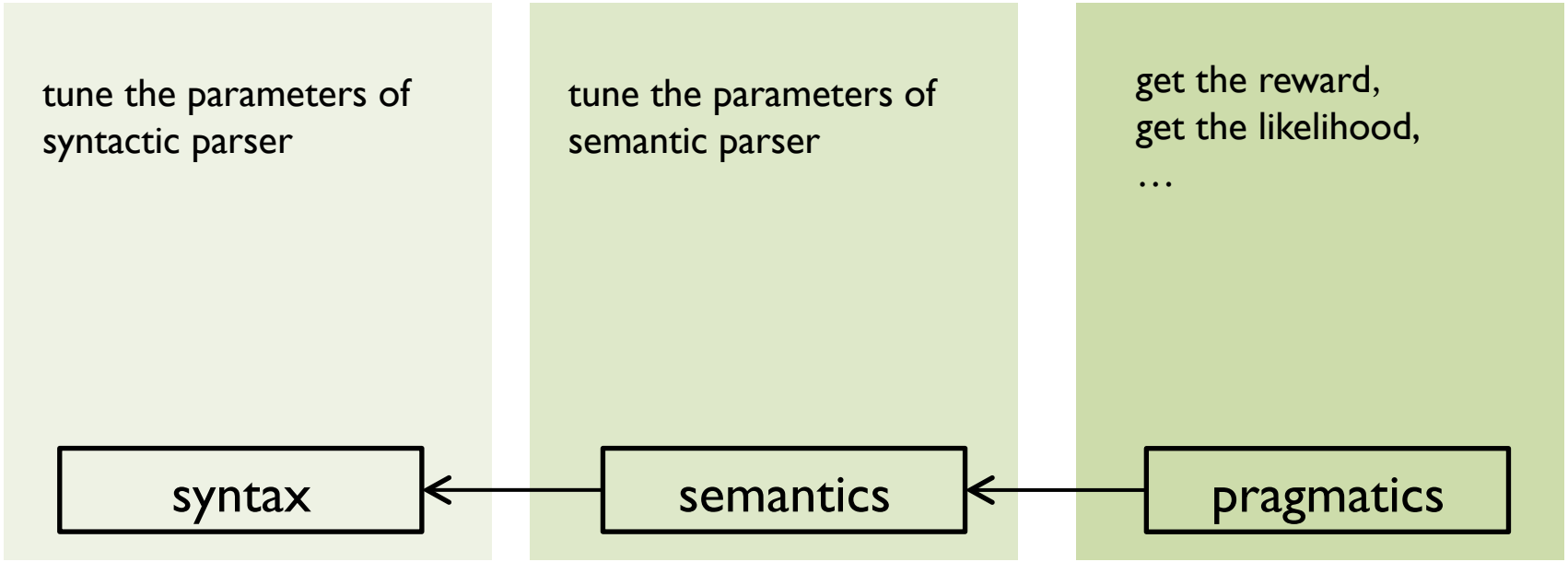
End-to-end: in action

- Deep NLP model in action



End-to-end: in learning

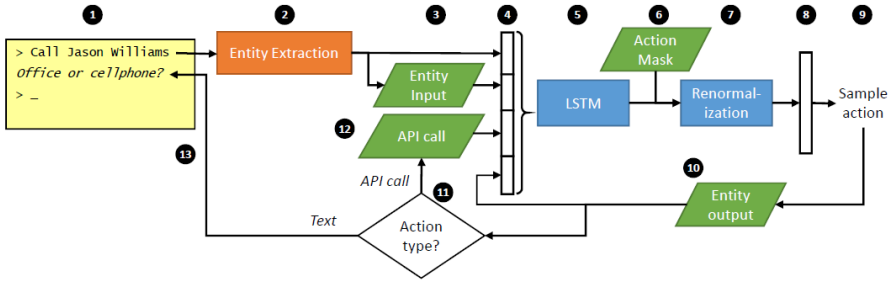
- Deep NLP model in learning



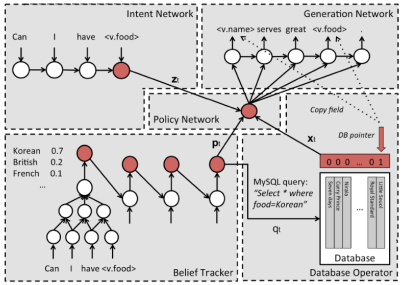
Success stories

- Most of the sequence-to-sequence models (e.g., for MT, or single-turn dialog) are based on generalized encoder-decoder framework, trained in end-to-end fashion

- System for multi-turn dialogue with specifically designed inner structure (e.g., state with designed semantic structure and policy-net)



Williams & Zweig (2016)

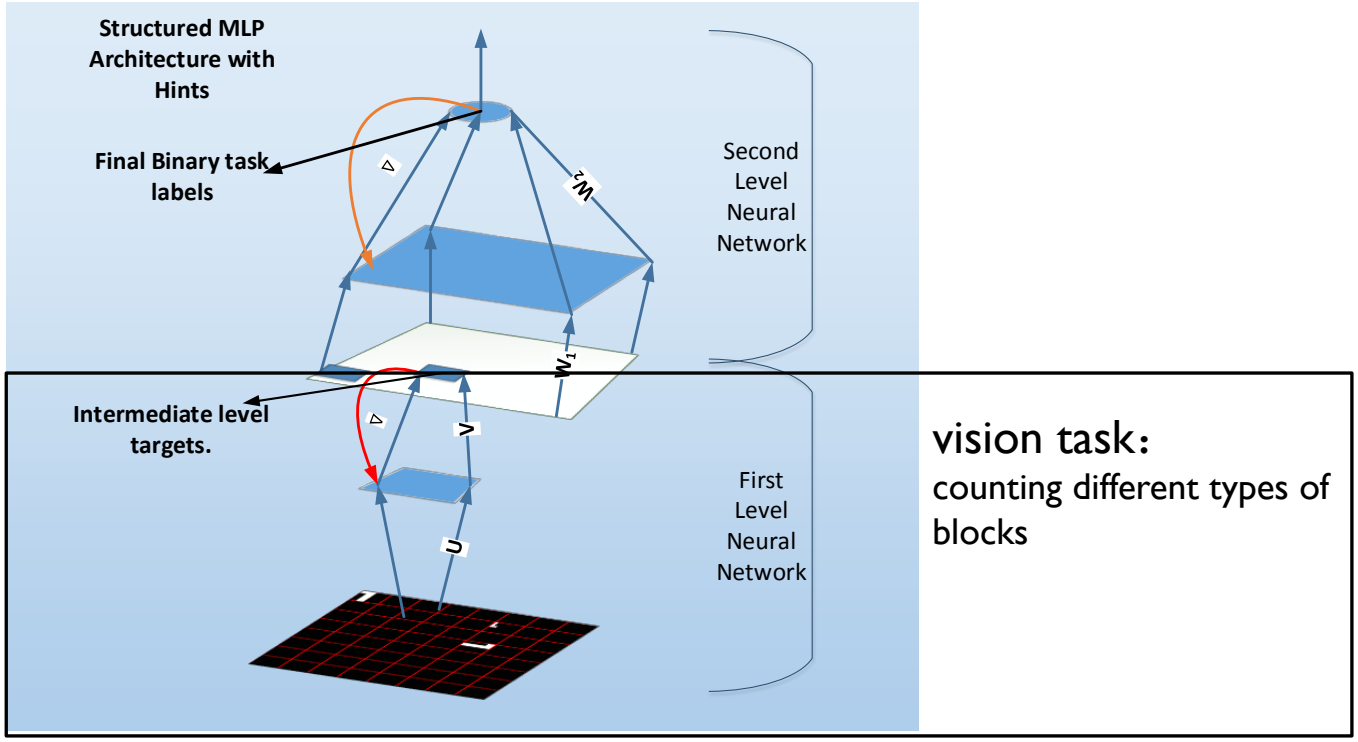


Wen et al. (2016)

- More complicated end-to-end model can be built which get the supervision from its interaction with the world

Failure story
{End-to-end learning could be very hard}

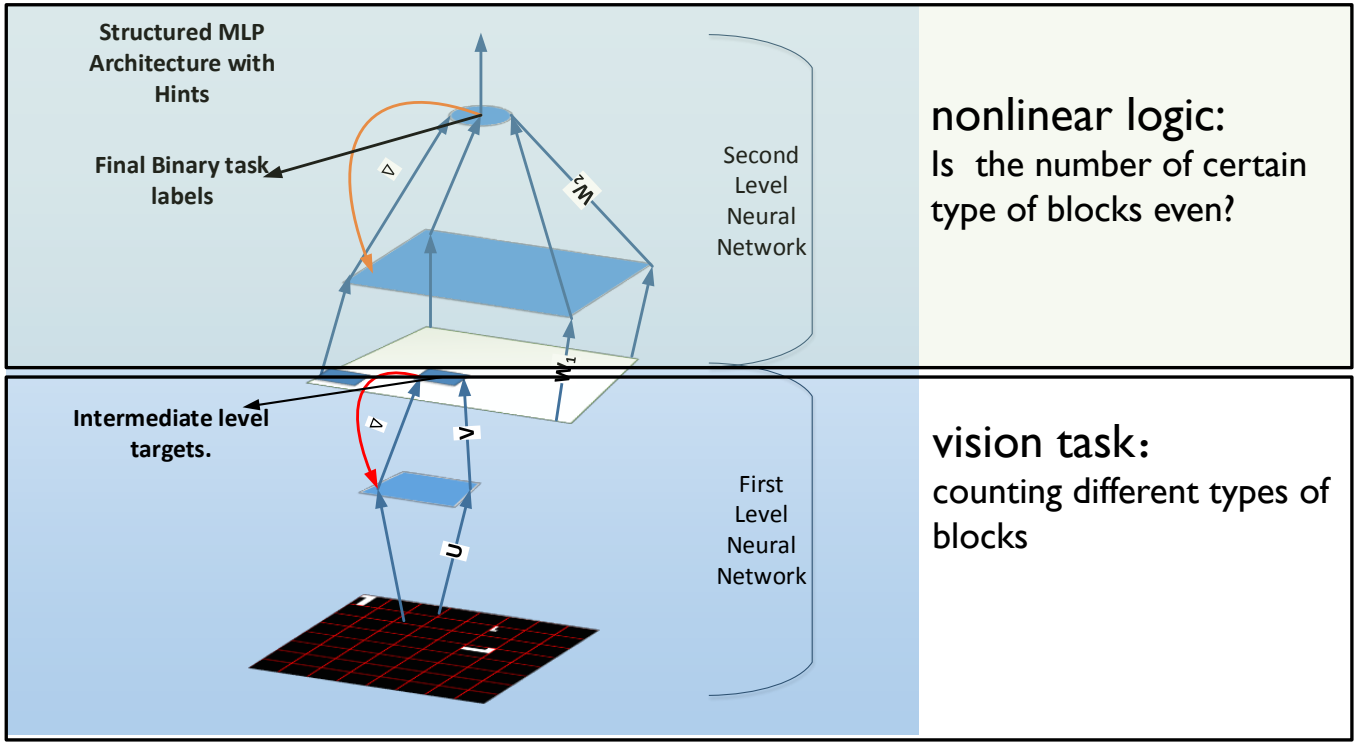
An example from computer vision, with weird problem structure



Gülçehre and Bengio (2013)

Failure story
{End-to-end learning could be very hard}

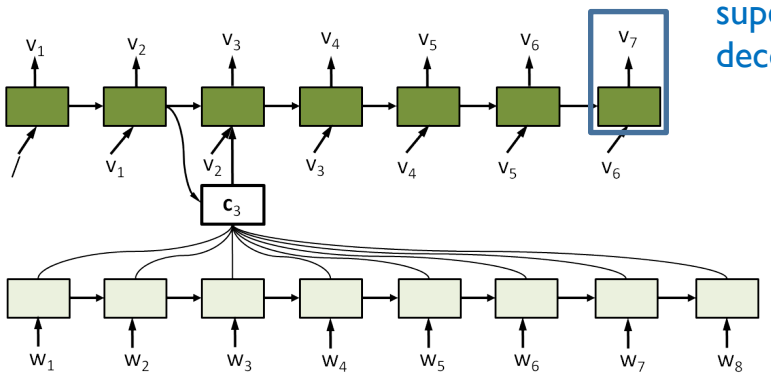
An example of weird problem structure



Gülçehre and Bengio (2013)

Why: The credit assignment difficulty

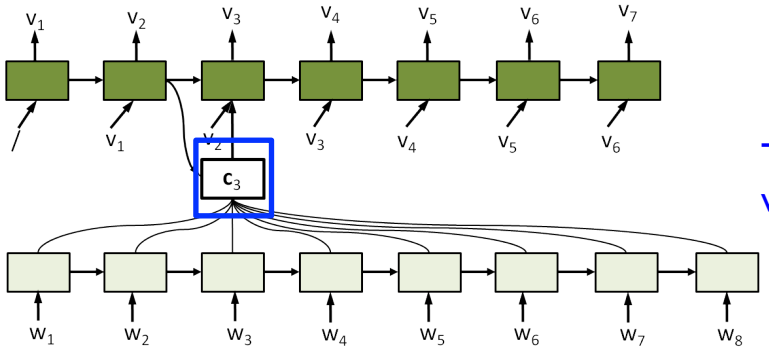
Most serious end-to-end neural systems are rather deep



supervision signal received at decoding time 7

Why: The credit assignment difficulty

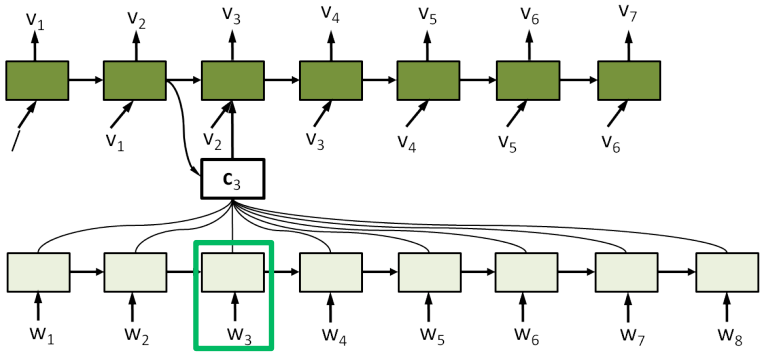
Most serious end-to-end neural systems are rather deep



The shortest path for it to go to source words is through attention（if any）

Why: The credit assignment difficulty

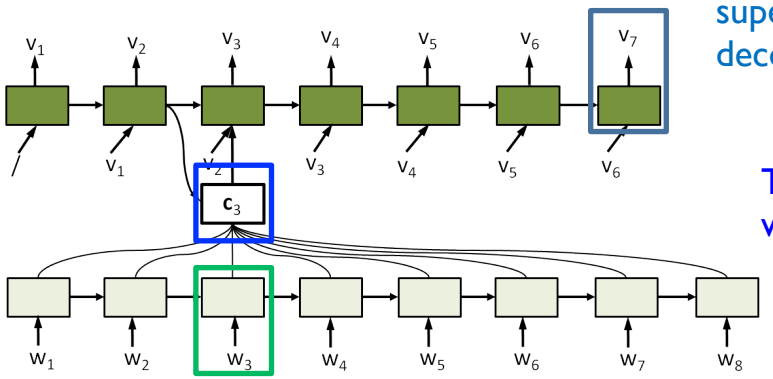
Most serious end-to-end neural systems are rather deep



Still , it needs to go to many layers of transformation
to modify the relevant parameters

Why: The credit assignment difficulty

Most serious end-to-end neural systems are rather deep



supervision signal received at decoding time 7

The shortest path for it to go to source words is through attention（if any）

Still , it needs to go to many layers of transformation to modify the relevant parameters

- The difficulty in credit assignment often renders the training hard

Stronger supervision helps

- Instead of injecting supervision only at the end, we can have supervision for intermediate steps/representations/layers

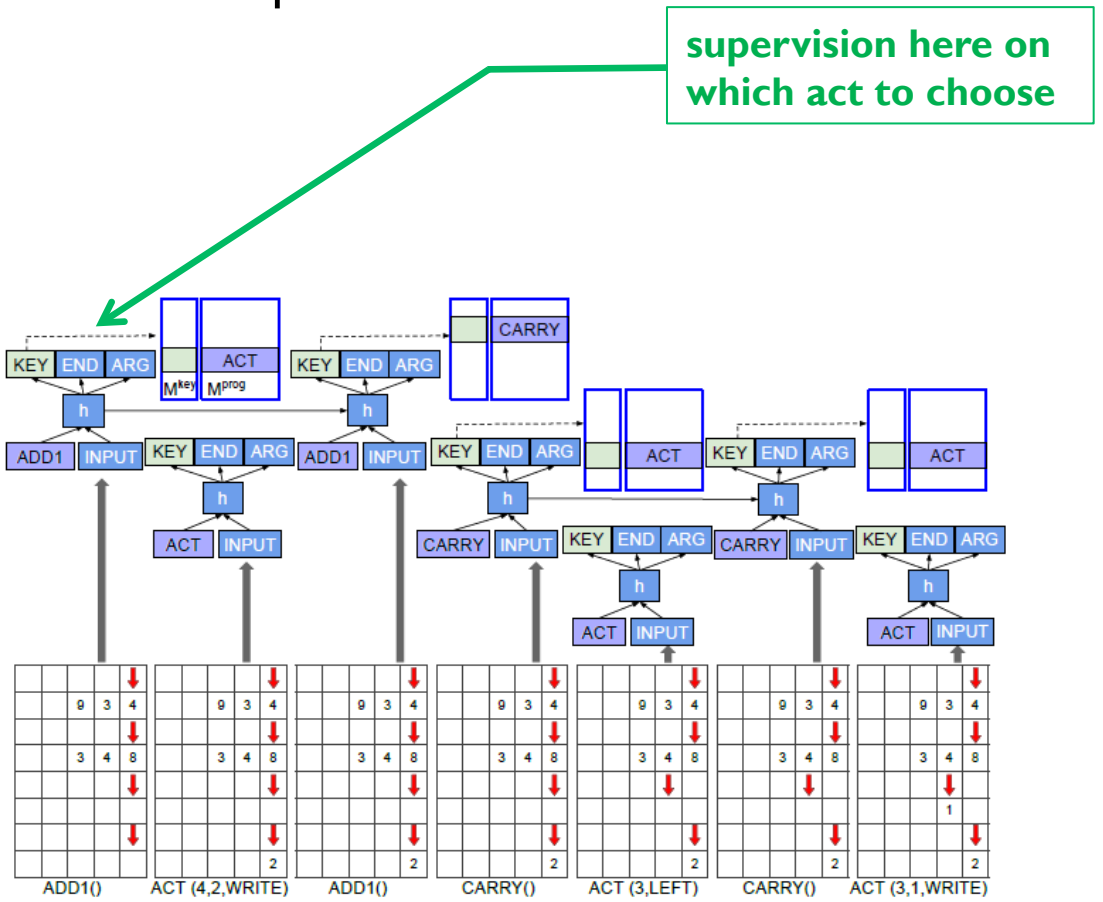
  "Intermediate small rewards are better than the final big reward"

- It is often more expensive to obtain than the "end supervision" (final reward), but sometimes it comes almost for free as byproduct

- A few examples
  ① Neural Programmer-Interpreter (Reed & de Freitas, 2016)
  ② Neural Enquirer (Yin et al., 2016b)
  ③ Memory Network (Weston et al., 2014)

Step-by-step supervision Example (1)

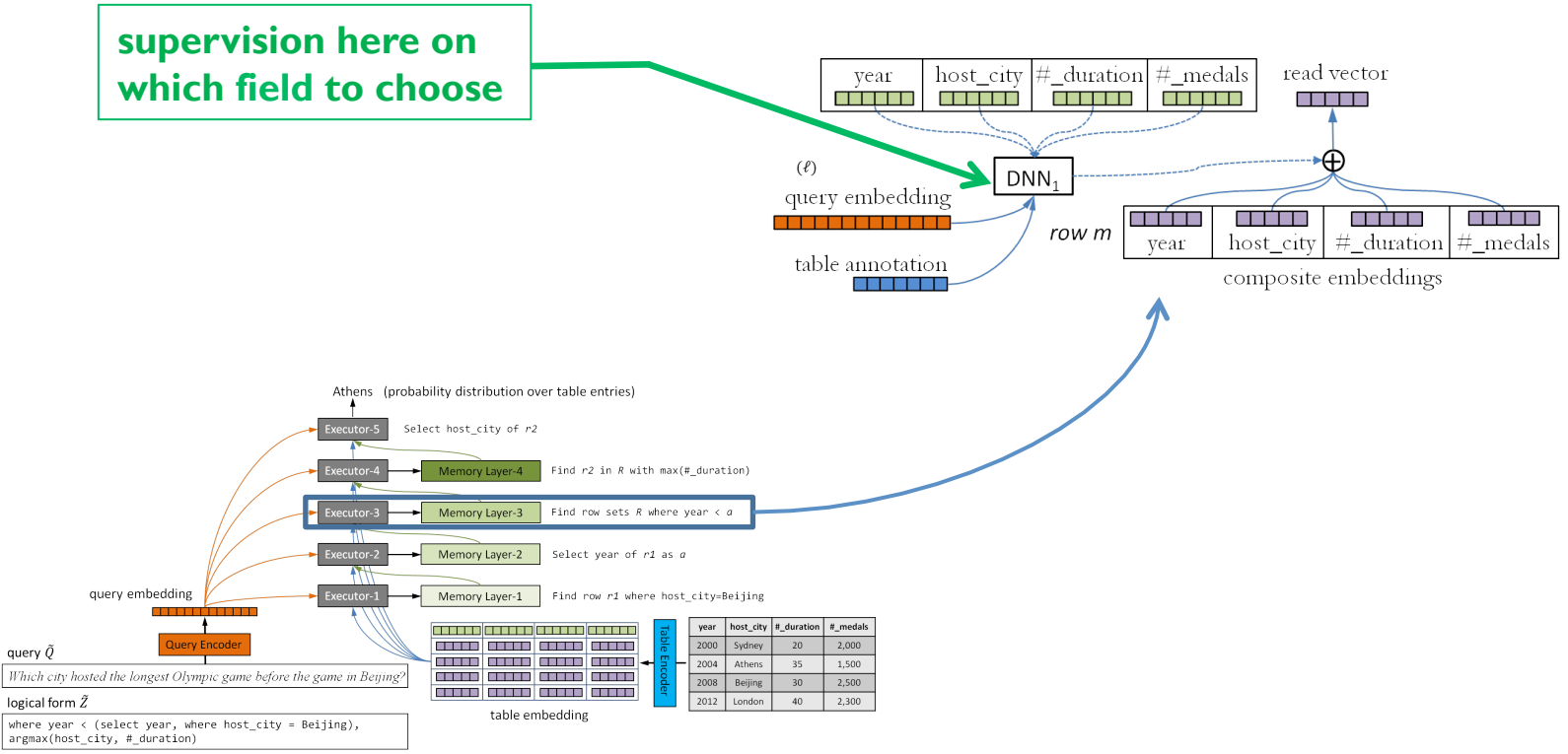
- Neural Programmer-Interpreter



Scott & de Freitas (2016)

Step-by-step supervision Example (2)

- Neural Enquirer can leverage step-by-step supervision in choosing the field to attend to in each execution

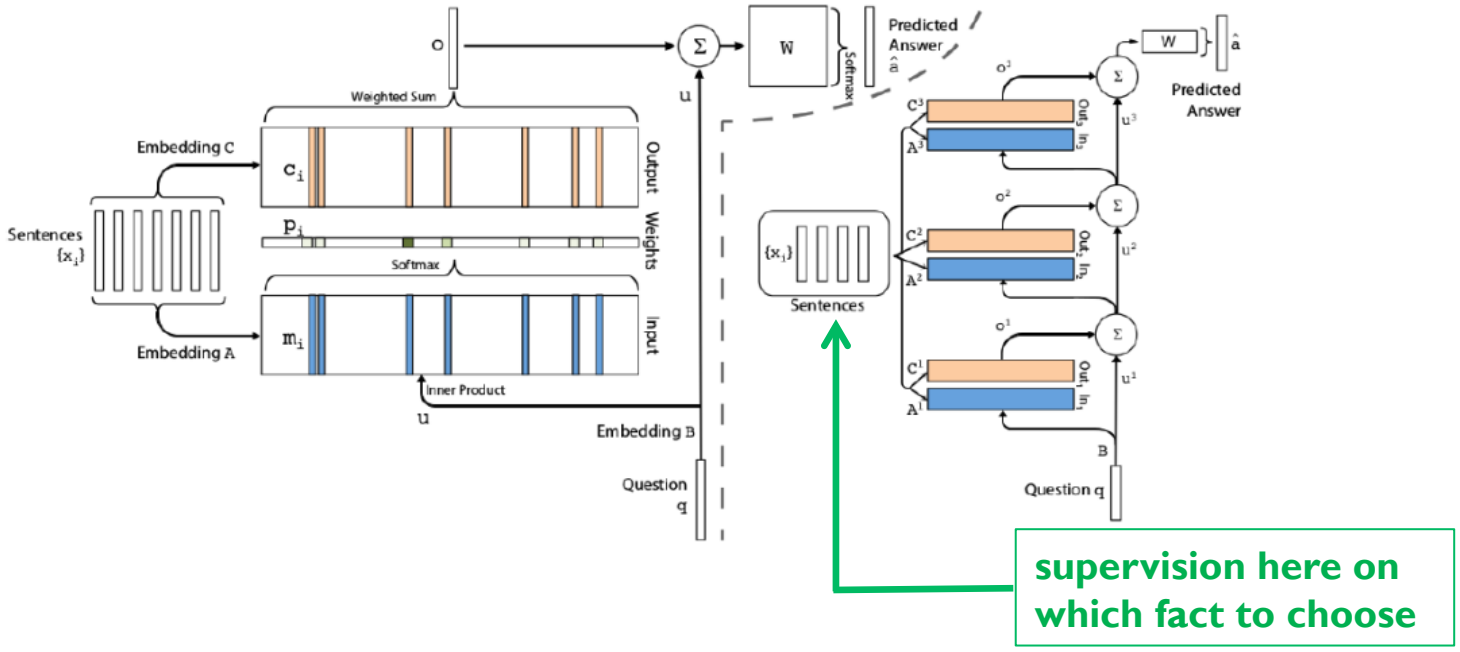


Yin et al. (2016b)

Step-by-step supervision: Memory Network

Memory Net is trained with **extra** step-by-step supervision

- In each reasoning step, the supporting fact is identified



Weston et al. (2014)

End-to-end Learning: pros and cons

Pros:

1. It allows you to focus on the architecture design
2. The model can learn from scratch (often with just BP)
3. The model can learn through its interaction with the world
4. Ideally, it can keep improving itself without any human intervention

Cons:

1. That is not the way human learn language, and probably not the (only) right way to teach machine

   "We don't learn a foreign languages from 4 million parallel sentences"
2. Sometimes the credit assignment is too hard, and we have to resort to some other means

Learning: Outline

- Overview
- End-to-end learning (or not?)
- Dealing with non-differentiability
- Grounding-based learning

When it is hard to do back-propagation

For example,

- Discrete decision has to be made for intermediate represenation
- Objective is not differentiable (BLEU, Rouge, user feedback)

What to do

- In general, we can use reinforcement learning (RL) or other sampling-based method.
- But let's describe some less general (but potentially simpler and more efficient) methods first

Minimum risk training for NMT

Maximizing the BLEU score

training data: $\{\langle \mathbf{x}^{(s)}, \mathbf{y}^{(s)} \rangle\}_{s=1}^{S}$

objective:
$$\mathcal{J}(\theta) = \sum_{s=1}^{S} \sum_{\mathbf{y} \in \mathcal{Y}(\mathbf{x}^{(s)})} P(\mathbf{y}|\mathbf{x}^{(s)}; \theta) \Delta(\mathbf{y}, \mathbf{y}^{(s)})$$
$$= \sum_{s=1}^{S} \mathbb{E}_{\mathbf{y}|\mathbf{x}^{(s)};\theta} \left[\Delta(\mathbf{y}, \mathbf{y}^{(s)}) \right]$$

optimization: $\hat{\theta}_{\text{MRT}} = \text{argmin}_{\theta} \left\{ \mathcal{J}(\theta) \right\}$

Minimum risk training for NMT

Maximizing the BLEU score

training data: $\{\langle \mathbf{x}^{(s)}, \mathbf{y}^{(s)} \rangle\}_{s=1}^{S}$

objective:
$$\mathcal{J}(\theta) = \sum_{s=1}^{S} \sum_{\mathbf{y} \in \mathcal{Y}(\mathbf{x}^{(s)})} P(\mathbf{y}|\mathbf{x}^{(s)};\theta)\Delta(\mathbf{y},\mathbf{y}^{(s)})$$
$$= \sum_{s=1}^{S} \mathbb{E}_{\mathbf{y}|\mathbf{x}^{(s)};\theta}\left[\Delta(\mathbf{y},\mathbf{y}^{(s)})\right]$$

optimization: $\hat{\theta}_{\mathrm{MRT}} = \mathrm{argmin}_{\theta}\left\{\mathcal{J}(\theta)\right\}$

⇩

Challenge

- It is intractable to calculate partial derivatives

$$\frac{\partial \mathcal{J}(\theta)}{\partial \theta_i} = \sum_{s=1}^{S} \mathbb{E}_{\mathbf{y}|\mathbf{x}^{(s)};\theta}\left[\Delta(\mathbf{y},\mathbf{y}^{(s)}) \sum_{n=1}^{N^{(s)}} \frac{\partial P(\mathbf{y}_n|\mathbf{x}^{(s)},\mathbf{y}_{<n};\theta)/\partial\theta_i}{P(\mathbf{y}_n|\mathbf{x}^{(s)},\mathbf{y}_{<n};\theta)}\right]$$

the search space is exponential

the loss function is usually non-decomposable

Minimum risk training for NMT

Maximizing the BLEU score

training data: $\{\langle \mathbf{x}^{(s)}, \mathbf{y}^{(s)} \rangle\}_{s=1}^{S}$

objective:
$$\mathcal{J}(\theta) = \sum_{s=1}^{S} \sum_{\mathbf{y} \in \mathcal{Y}(\mathbf{x}^{(s)})} P(\mathbf{y}|\mathbf{x}^{(s)};\theta)\Delta(\mathbf{y},\mathbf{y}^{(s)})$$
$$= \sum_{s=1}^{S} \mathbb{E}_{\mathbf{y}|\mathbf{x}^{(s)};\theta}\left[\Delta(\mathbf{y},\mathbf{y}^{(s)})\right]$$

optimization: $\hat{\theta}_{\text{MRT}} = \text{argmin}_\theta \left\{\mathcal{J}(\theta)\right\}$

⇩

Challenge

- It is intractable to calculate partial derivatives

$$\frac{\partial \mathcal{J}(\theta)}{\partial \theta_i} = \sum_{s=1}^{S} \mathbb{E}_{\mathbf{y}|\mathbf{x}^{(s)};\theta}\left[\Delta(\mathbf{y},\mathbf{y}^{(s)}) \sum_{n=1}^{N^{(s)}} \frac{\partial P(\mathbf{y}_n|\mathbf{x}^{(s)},\mathbf{y}_{<n};\theta)/\partial\theta_i}{P(\mathbf{y}_n|\mathbf{x}^{(s)},\mathbf{y}_{<n};\theta)}\right]$$

- the search space is exponential
- the loss function is usually non-decomposable

⇨

Approximation

- We approximate the true distribution with a sampled space

$$\tilde{\mathcal{J}}(\theta) = \sum_{s=1}^{S} \sum_{\mathbf{y} \in \mathcal{S}(\mathbf{x}^{(s)})} \frac{P(\mathbf{y}|\mathbf{x}^{(s)};\theta)^\alpha}{\sum_{\mathbf{y}' \in \mathcal{S}(\mathbf{x}^{(s)})} P(\mathbf{y}'|\mathbf{x}^{(s)};\theta)^\alpha} \Delta(\mathbf{y},\mathbf{y}^{(s)})$$
$$= \sum_{s=1}^{S} \sum_{\mathbf{y} \in \mathcal{S}(\mathbf{x}^{(s)})} Q(\mathbf{y}|\mathbf{x}^{(s)};\theta,\alpha)\Delta(\mathbf{y},\mathbf{y}^{(s)})$$
$$= \sum_{s=1}^{S} \mathbb{E}_{\mathbf{y}|\mathbf{x}^{(s)};\theta,\alpha}\left[\Delta(\mathbf{y},\mathbf{y}^{(s)})\right]$$

Shen et al. (2016)

Reinforcement Learning

- Reinforcement learning (RL) is a rather general framework (with supervised learning a special case), which can handle
  - ① discrete intermediate decisions

    "In RL, they are just discrete actions"
  - ② Non-differentiable objectives

    "In RL, they are just rewards"

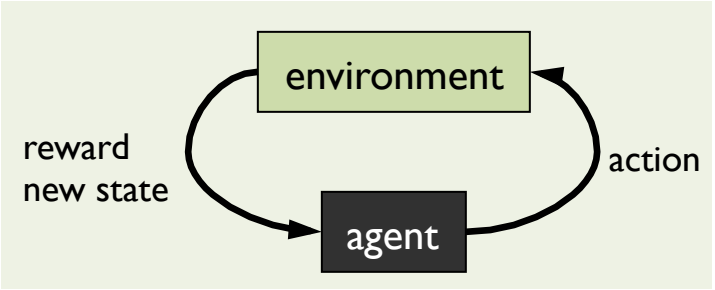
- Could still be end-to-end, although no longer back-propagation
- The "depth" of the neural models sometimes translated to the delay of the rewards, therefore the same credit-assignment problem

- A great tool, but comes at a cost (mainly inefficiency in learning)

RL: general formulation

Markov Decision Process (MDP)

- set of states S, set of actions A, initial state S_0
- transition model $P(s,a,s')$
- reward function $r(s)$



- Goal: maximize cumulative reward in the long run
- Policy: mapping from S to A
  - $\pi(s)$ or $\pi(s,a)$ (deterministic vs. stochastic)
- Reinforcement learning
  - transitions and rewards usually not available
  - how to change the policy based on experience
  - how to explore the environment

Reinforcement Learning for deep NLP models

RL is useful for deep NLP models, when

① The supervision signal comes as delayed reward, e.g. ,

- ❶ in evaluating the entire generated sequence (eg. in NMT)
- ❷ in evaluating the gain of the agent's interaction with the environment

② Representing the entire distribution of operations is difficult, e.g.,

- ❶ altering the structure of sentence as discrete operations
- ❷ discrete structural decision in the intermediate representation

RL Application Examples

General ideas:

① Sequential prediction as decision making

② Pre-defined discrete action for altering the structures

③ Strategy (in dialogue) as discrete action

Real applications:

① Text game

② RL for natural language generation

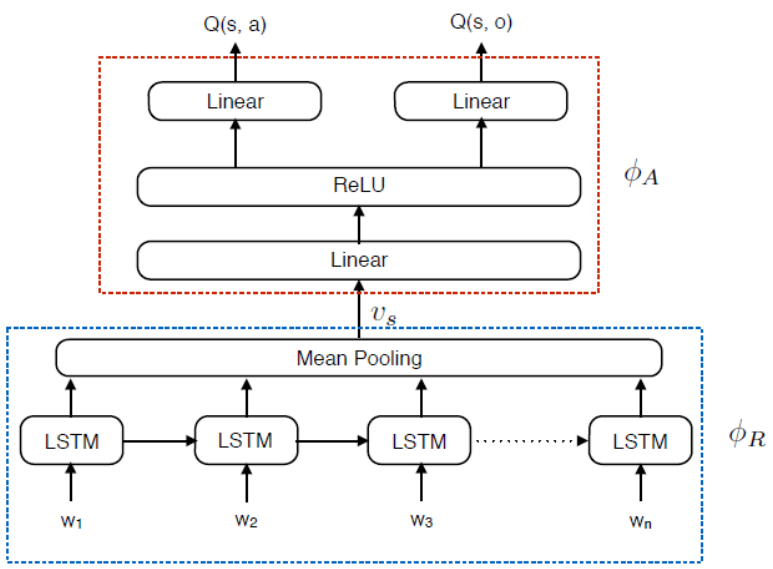
③ RL for policy learning in dialogue system

RL Application : (1)

Real applications:

① Text game: the agent has to make text-based decisions, which will be rewarded later

② Both the word-embedding and composition models are learned indirectly from Q-Learning



State 1: The old bridge
You are standing very close to the bridge's eastern foundation. If you go east you will be back on solid ground ... The bridge sways in the wind.

Command: **Go east**

State 2: Ruined gatehouse
The old gatehouse is near collapse. Part of its northern wall has already fallen down ... East of the gatehouse leads out to a small open area surrounded by the remains of the castle. There is also a standing archway offering passage to a path along the old southern inner wall.
Exits: Standing archway, castle corner, Bridge over the abyss

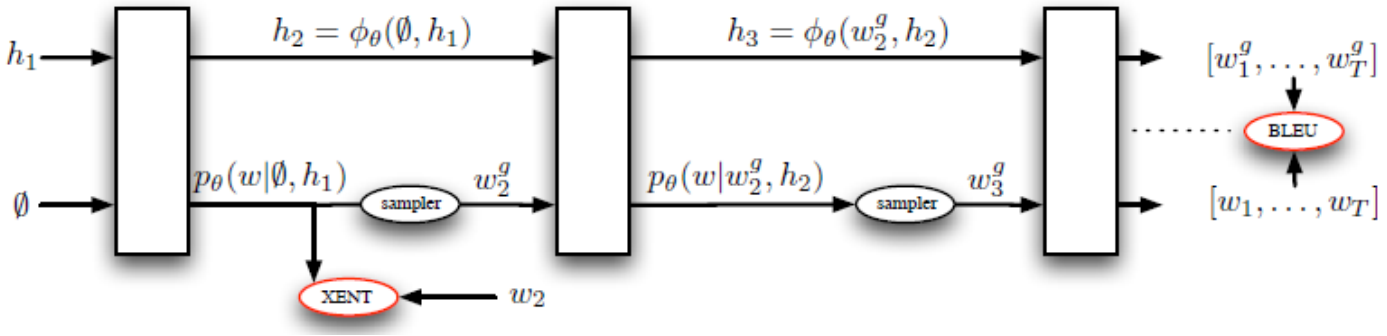
Narashimhan et al. (2015)

RL Application : (2)

Real applications:

① Simple policy gradient to maximize the BLEU

② Intriguingly related to minimum risk training (Shen et al., 2015)



Data: a set of sequences with their corresponding context.
Result: RNN optimized for generation.
Initialize RNN at random and set N^{XENT}, $N^{\text{XE+R}}$ and Δ;
for $s = T, 1, -\Delta$ **do**
  if $s == T$ **then**
    train RNN for N^{XENT} epochs using XENT only;
  else
    train RNN for $N^{\text{XE+R}}$ epochs. Use XENT loss in the first s steps, and REINFORCE (sampling from the model) in the remaining $T - s$ steps;
  end
end

Ranzato et al. (2016)

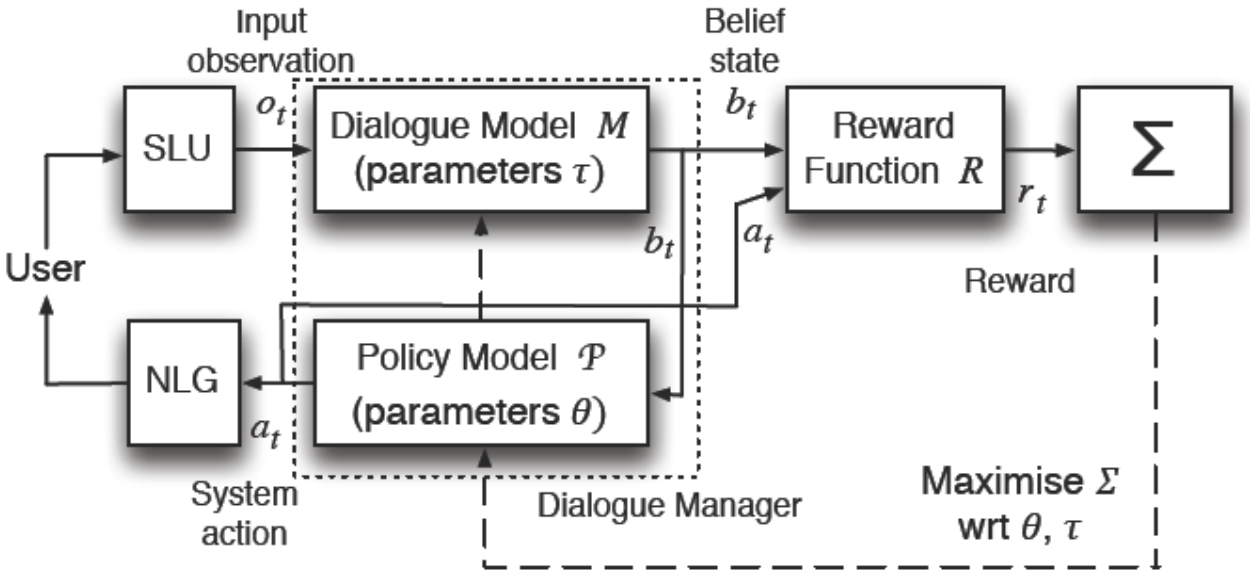
RL Application : (3)

RL for policy learning in dialogue system

① Dialogue Model: representing the transition and observation probability function (“ what is the next state?”)

② Policy Model: “which action to take?”

③ Basic ideas: designed state vector + belief state tracking (POMDP) + RL



Young et al. (2015)

Reinforcement Learning: pros and cons

Pros:

- More modeling flexibility (discrete decision allowed)
- Richer form of supervision (delayed reward is much broader than labels)

Cons:

- Generally still sampling methods, and low in efficiency (many works trying to fix that)

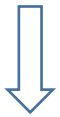
Reinforcement Learning: pros and cons

Pros:

- More modeling flexibility (discrete decision allowed)
- Richer form of supervision (delayed reward is much broader than labels)

Cons:

- Generally still sampling methods, and low in efficiency (many works trying to fix that)



Better be combined with more efficient learning paradigm (e.g., supervised learning) to work

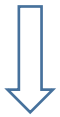
Reinforcement Learning: pros and cons

Pros:

- More modeling flexibility (discrete decision allowed)
- Richer form of supervision (delayed reward is much broader than labels)

Cons:

- Generally still sampling methods, and low in efficiency (many works trying to fix that)



Better be combined with more efficient learning paradigm (e.g., supervised learning) to work

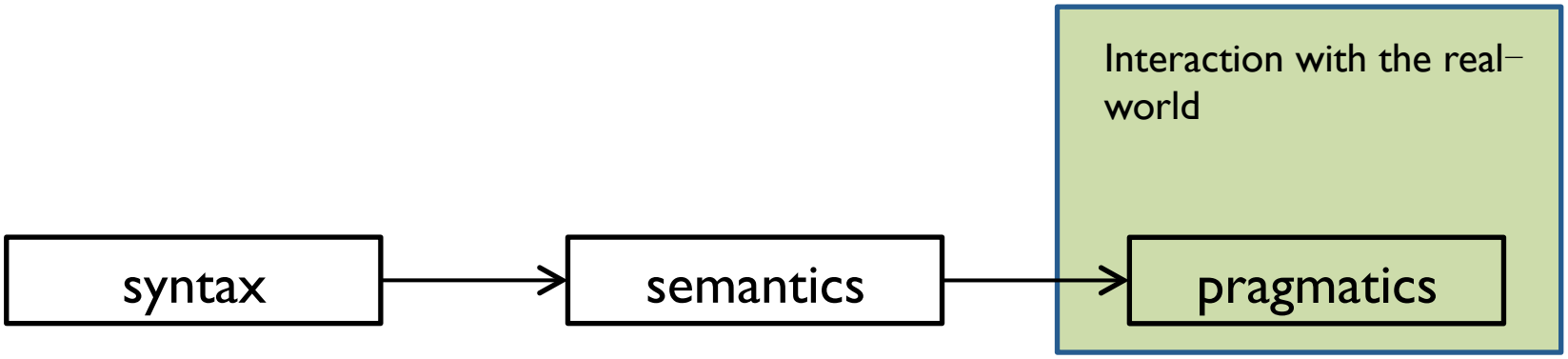
Several recent effort in dialogue model and control

Learning: Outline

- Overview
- End-to-end learning (or not?)
- Dealing with non-differentiability
- Grounding-based learning

Grounding-based Learning

- Text, e.g., a sentence, interacts with the world and receives supervision signal from there



Grounding-based Learning: Examples

Grounded on its interaction with other agent

- Example: Text game

Grounded on other modality

- Example: Image/video captioning and image QA

Grounded on its interaction with KB

- Example: Table querying

Grounding-based Learning: Examples

Grounded on its interaction with other agent

- Example: Text game

Grounded on other modality

- Example: Image/video captioning and image QA

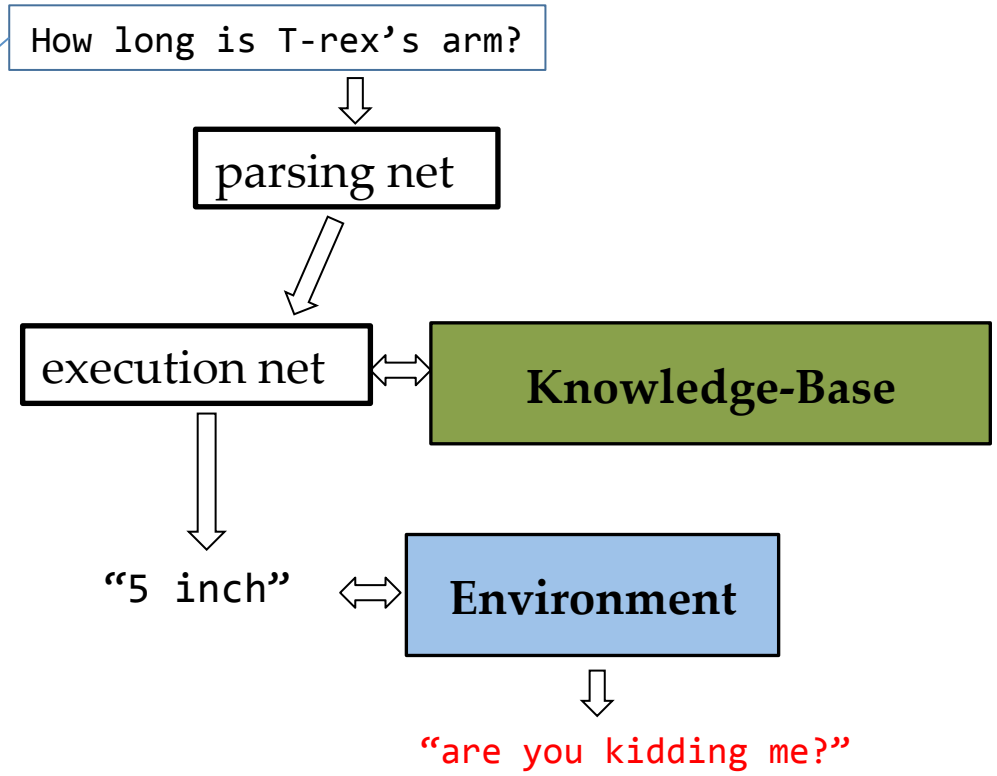
Grounded on its interaction with KB

- Example: Table querying

Grounding-based Learning: QA with KB

Grounded on its interaction with a knowledge-base

① 1. Table querying

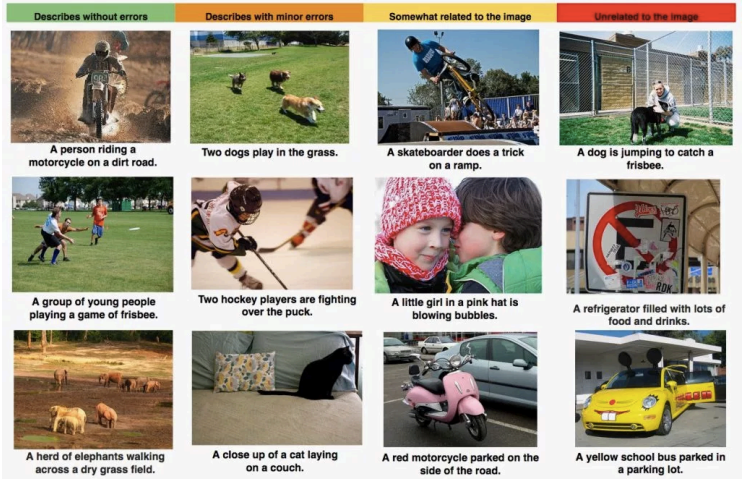


Grounding-based Learning : other modality

Automatic Image/Video Captioning

① Generate captions for images, so ground texts to images (through a shared representation of image and text)

② Supervised learning in end-to-end fashion

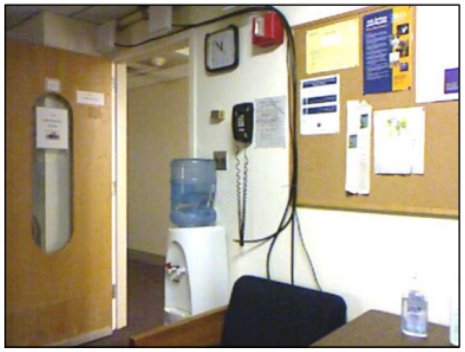


Vinyals et al. (2014)
and many more

Image Question-Answering

① Neural network models to answer natural language questions concerning a given image

② Ground text (both question and answers) to the images through the neural network answerer



Question: what is the largest blue object in this picture?
Ground truth: water carboy
Proposed CNN: water carboy

Question: How many pieces does the curtain have?
Ground truth: 2
Proposed CNN: 2

Ma et al. (2016)

Reference (part III)

- [Guo, 2015] Generating Text with Deep Reinforcement Learning. Hongyu Guo
- [Xu et al.,2015] Show, Attend and Tell: Neural Image Caption Generation with Visual Attention
  Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, Yoshua Bengio
- [Reed & de Freitas, 2016] Neural Programmer-Interpreters. Scott Reed, Nando de Freitas
- [Williams & Zweigs, 2016] End-to-end LSTM-based dialog control optimized with supervised and reinforcement learning. Jason D. Williams and Geoffrey Zweig
- [Kurachet al., 2015] Neural Random-Access Machines Karol Kurach, Marcin Andrychowicz, Ilya Sutskever
- [Bordes & Weston, 2016] Learning End-to-End Goal-Oriented Dialog. Antoine Bordes, Jason Weston
- [Gülçehre & Bengio, 2015] Knowledge Matters: Importance of Prior Information for Optimization. Çağlar Gülçehre, Yoshua Bengio
- [Yin et al., 2016b] Neural Enquirer: Learning to Query Tables with Natural Language. Pengcheng Yin, Zhengdong Lu, Hang Li, Ben Kao
- [Wen et al., 2016] A Network-based End-to-End Trainable Task-oriented Dialogue System Tsung-HsienWen, Milica Gaši´c, Nikola Mrkši´c, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, David Vandyke, Steve Young
- [Weston et al., 2015] Memory Networks. JasonWeston, Sumit Chopra & Antoine Bordes
- [Ranzato et al., 2016] Sequence level training with recurrent neural networks. Marc Aurelio Ranzato, Sumit Chopra, Michael Auli, Wojciech Zaremba
- [Shen et al., 2015] Minimum risk training for neural machine translation. Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu.
- [Vinyals et al., 2014] Sequence to sequence learning with neural networks. Ilya Sutskever, Oriol Vinyals, and Quoc V Le.
- [Vinyals et al., 2015] Show and tell: A neural image caption. O. Vinyals, A. Toshev, S. Bengio and D. Erhan
- [Sukhbaatar et al., 2015] End-to-end memory networks. S. Sukhbaatar, J. Weston, R. Fergus.
- [Peng et al., 2015] Towards Neural Network-based Reasoning Baolin Peng, Zhengdong Lu, Hang Li, Kam-Fai Wong
- [Narasimhan et al., 2015] Language Understanding for Text-based Games using Deep Reinforcement Learning. Karthik Narasimhan, Tejas Kulkarni, Regina Barzilay.
- [Ma et al., 2015] Learning to Answer Questions From Image Using Convolutional Neural Network. Lin Ma, Zhengdong Lu, Hang Li
- [Luong et al. 2015] Effective approaches to attention-based neural machine translation. Minh-Thang Luong, Hieu Pham, and Christopher D Manning.
- [Bahdanau et al.2014] Neural machine translation by jointly learning to align and translate. Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio.

Part-IV: Conclusion

Conclusion (slide 1)

Exciting progress in deep learning for NLP

Models

❶ Sequence-to-sequence learning
❷ Attentive models
❸ External memory

Tasks *:*

❶ Neural machine translation
❷ Neural dialogue models
❸ Neural net-based parsing
❹ Neural net-based reasoning

Conclusion (slide 2)

- Differentiable data-structure is powerful

Memory

❶ Shor-term, intermediate, long-term memory

❷ Fixed, linear and layered arrangement of memory

Addressing

❶ Location-based addressing, and how it can be made differentiable

❷ Content-based addressing, and different implementation

❸ Hybrid addressing and different ways of doing that

- It is not the entire world, and we may need to give up the differentiability when needed

Conclusion (slide 3)

- We are creating new learning paradigms, but more are needed
  - ❶ End-to-end learning is powerful and surprisingly effective
  - ❷ Sometimes, step-by-step supervision is needed
  - ❸ Non-differentiable objective can be partially tackled
  - ❹ R.L. enables us to learn discrete action and/or from delayed rewards

- We need more learning paradigms
  - ❶ To combine supervised learning and R.L.
  - ❷ To take richer form of supervision

Thank you !

We are hiring!

Please contact us at

hr@deeplycurious.ai