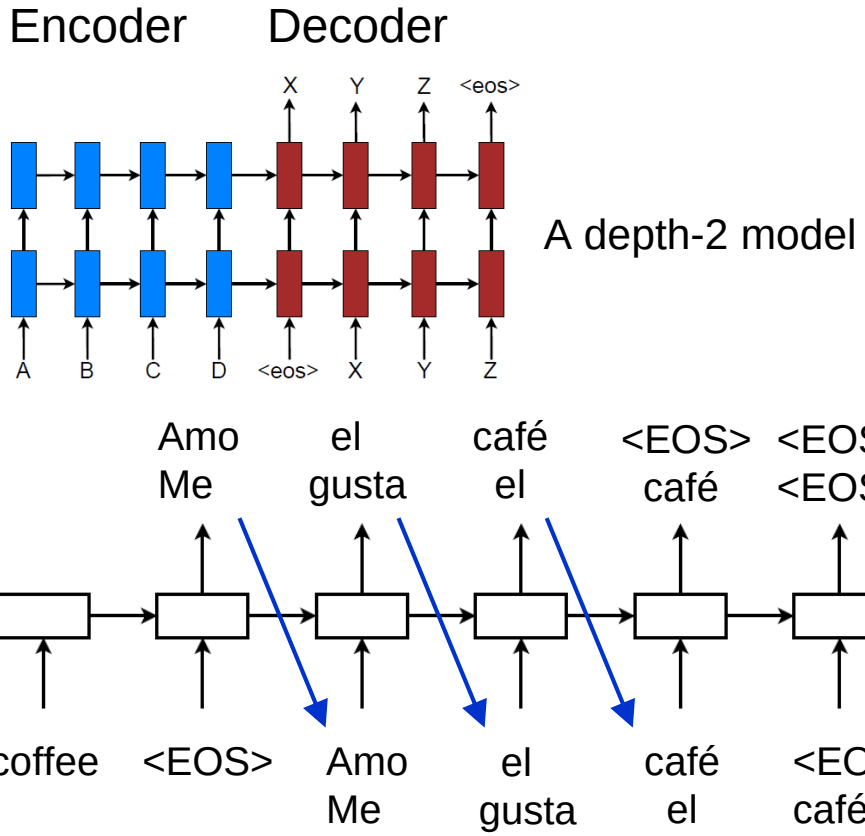
CS194/294-129: Designing, Visualizing and Understanding Deep Neural Networks

John Canny

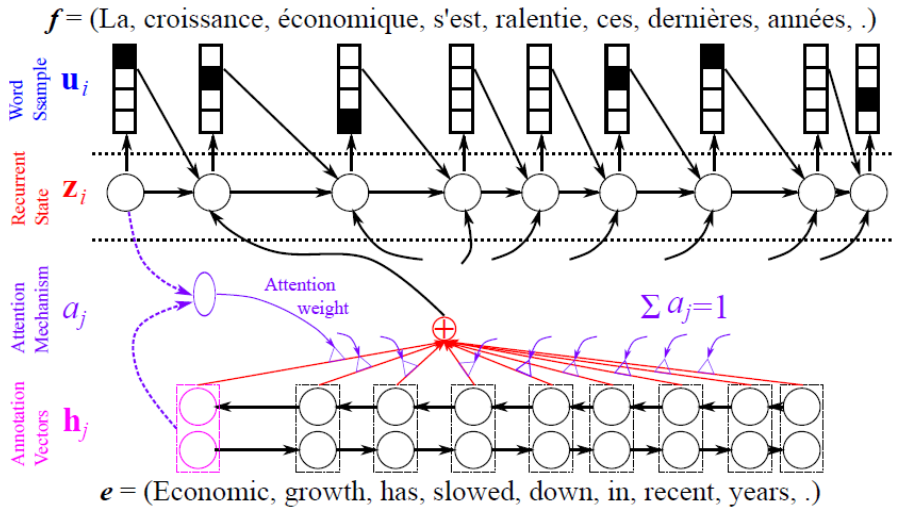
Spring 2018

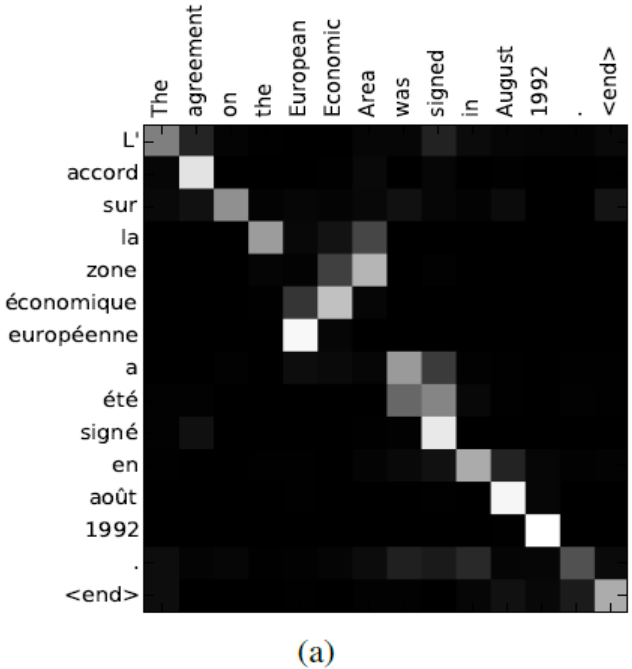
Lecture 15: Memory Networks

Last Time: Sequence-To-Sequence Translation



Last Time: Soft Attention for Translation





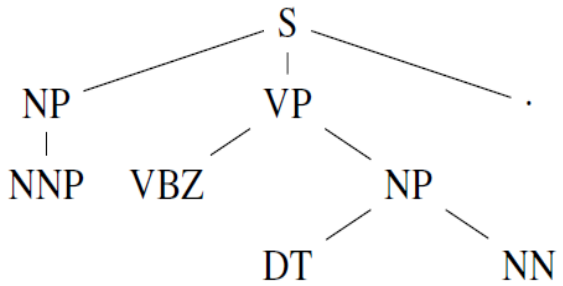
(a)

From Y. Bengio CVPR 2015 Tutorial

Last Time: Parsing as Translation

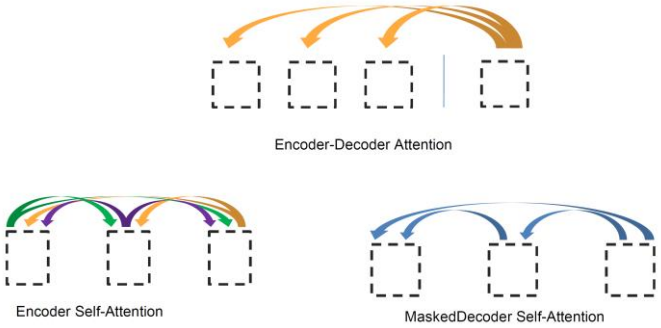
Sequence models generate linear structures, but these can easily encode trees by “closing parens” (prefix tree notation):

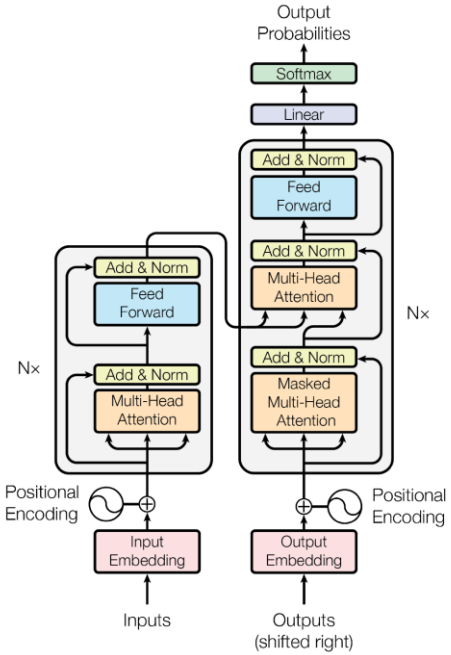
John has a dog . →

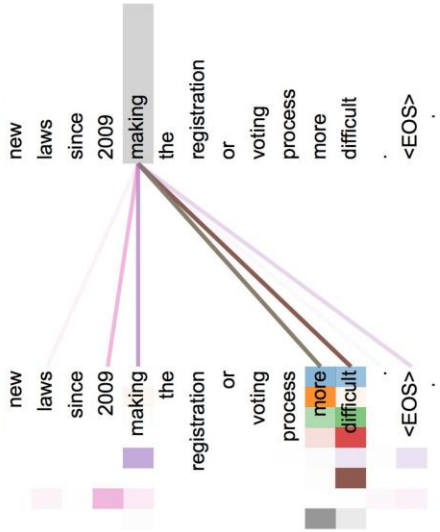


John has a dog . → (S (NP NNP)$_{\text{NP}}$ (VP VBZ (NP DT NN)$_{\text{NP}}$)$_{\text{VP}}$.)$_{\text{S}}$

Last Time: Attention only Models: Transformer







Multi-headed self-attention

image from Lukas Kaiser, Stanford NLP seminar

This Time: Memory Networks

“Hey Google, Explain Memory Networks”

Google: Maybe you should ask a Facebook Agent?

Memory Networks

- Convolutional Networks: Activations (content) fully predictable from inputs.

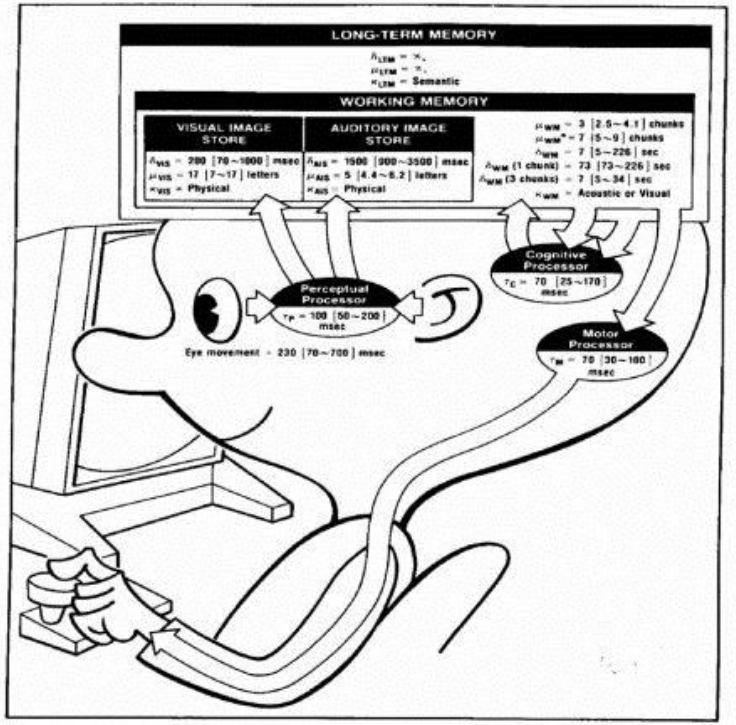
- Attention Models: Activations (content) depends mostly on the input, agent has also dynamic attention. You can think of attention as an analog of pointers or references in traditional programming languages.

- Memory Networks: Provide general purpose memory, pointers (via attention), and read/write capability. Critical for dynamic memory in conversational agents.

Human Memory

- Short-term or Working Memory: Dynamic, ephemeral, over a time scale of seconds.

- Long-Term Memory: Stores Events, Write-Once, Read-Many (WORM). Time frame is minutes to years.



Card, Moran and Simon "*The Model Human Processor: An Engineering Model of Human Performance" 1986*

Memory Networks: Basic Dialog Tasks

Task 1:
1 Mary moved to the bathroom.
2 John went to the hallway.
3 Where is Mary? bathroom 1
4 Daniel went back to the hallway.
5 Sandra moved to the garden.
6 Where is Daniel? hallway 4
7 John moved to the office.
8 Sandra journeyed to the bathroom.
9 Where is Daniel? hallway 4
10 Mary moved to the hallway.
11 Daniel travelled to the office.
12 Where is Daniel? office 11

babI dataset: Facebook research

Memory Networks: babI dataset

Task 3
5 Mary journeyed to the office.
17 Mary journeyed to the bathroom.
23 Mary dropped the football.
25 Where was the football before the bathroom? office 23 17 5

Task 14
2 Julie went to the school this morning.
4 Yesterday Julie went to the office.
5 Where was Julie before the school? Office 2 4

Task 16
6 Julius is a swan.
7 Julius is green.
9 Greg is a swan.
10 What color is Greg? Green 9 6 7

Task 19
2 The kitchen is north of the office.
4 The office is west of the garden.
6 How do you go from the kitchen to the garden? s,e 2 4

Memory Networks

Support several classes of tasks:

- Reading and Comprehension: Read a passage of text and answer questions about it.

- Dialog: To remember previous short- and long-term information during a conversation (what were we talking about?)

- Learning from Dialog: Learn new tasks from conversations with users

- Memory Networks support Reading with Attention over Memory (RAM).

Long- and Short-Term Memory

Long-Term Memories h_i	Shaolin Soccer directed_by Stephen Chow Shaolin Soccer written_by Stephen Chow Shaolin Soccer starred_actors Stephen Chow Shaolin Soccer release_year 2001 Shaolin Soccer has_genre comedy Shaolin Soccer has_tags martial arts, kung fu soccer, stephen chow Kung Fu Hustle directed_by Stephen Chow Kung Fu Hustle written_by Stephen Chow Kung Fu Hustle starred_actors Stephen Chow Kung Fu Hustle has_genre comedy action Kung Fu Hustle has_imdb_votes famous Kung Fu Hustle has_tags comedy, action, martial arts, kung fu, china, soccer, hong kong, stephen chow The God of Cookery directed_by Stephen Chow The God of Cookery written_by Stephen Chow The God of Cookery starred_actors Stephen Chow The God of Cookery has_tags hong kong Stephen Chow From Beijing with Love directed_by Stephen Chow From Beijing with Love written_by Stephen Chow From Beijing with Love starred_actors Stephen Chow, Anita Yuen . . . <and more> . . .
Short-Term Memories c_1^u c_1^r	1) I'm looking a fun comedy to watch tonight, any ideas? 2) Have you seen Shaolin Soccer? That was zany and great.. really funny but in a whacky way.
Input c_2^u	3) Yes! Shaolin Soccer and Kung Fu Hustle are so good I really need to find some more Stephen Chow films I feel like there is more awesomeness out there that I haven't discovered yet ...
Output y	4) God of Cookery is pretty great, one of his mid 90's hong kong martial art comedies.

Note: Neural RAM long-term memory is typically "WORM" – Write Once, Read Many

Memory Network Framework

Four Components:

I: (input feature map) converts input data to internal feature representation.

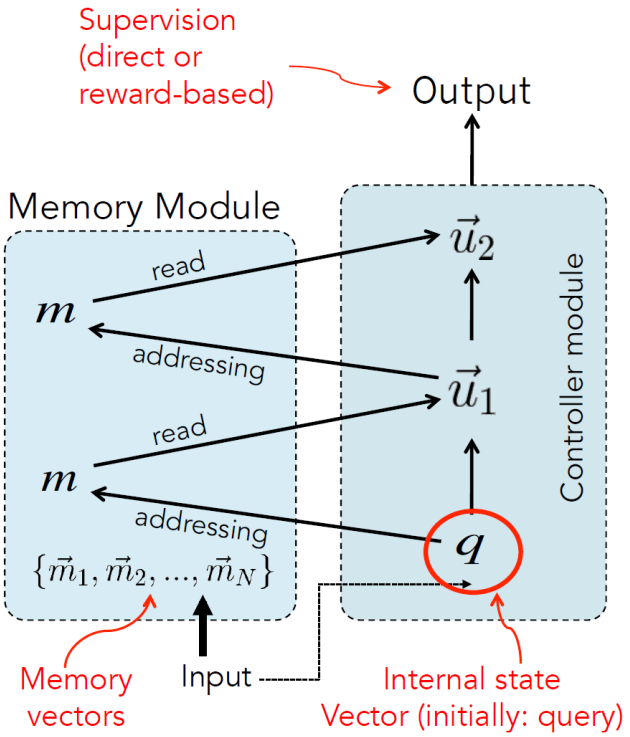
G: (generalization) update memories given new input.

O: produce new output (in feature representation space) given the memories.

R**:** (response) convert output O into a response seen by the outside world.

Igor borrowed from Jason Weston's 2016 ICML Tutorial

Memory Iteration



[Figure by Saina Sukhbaatar]

Q&A Memory Network

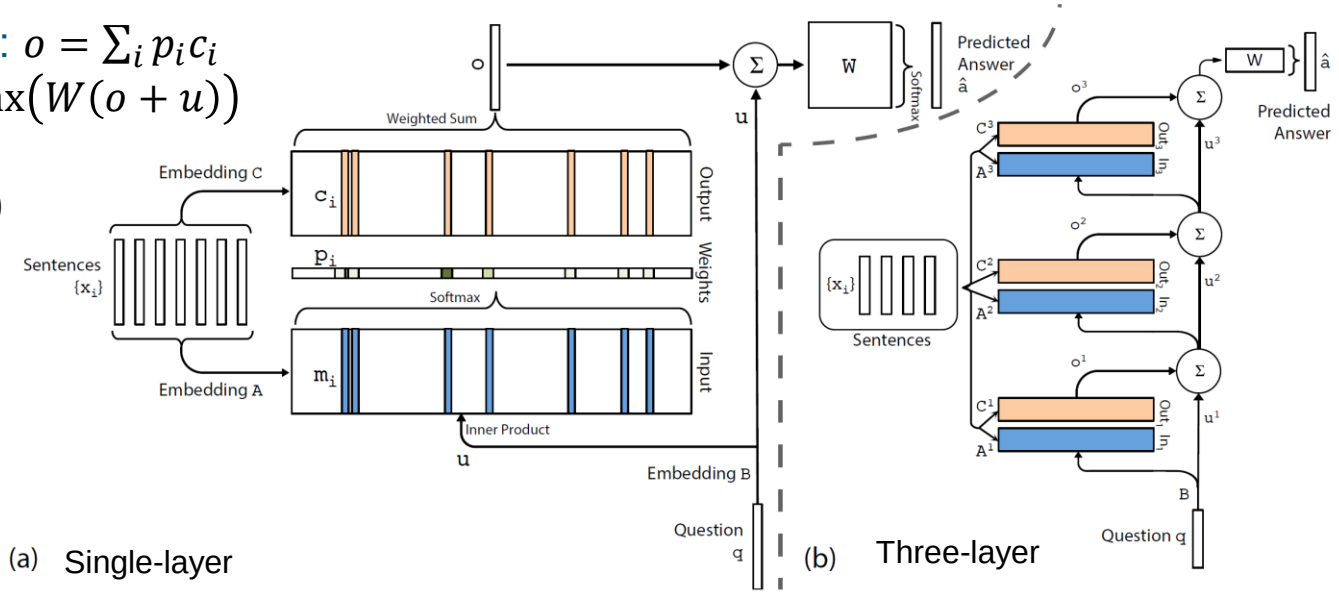
Sukbhaatar et al “End-to-End Memory Networks” 2015

Input data: $m_i = Ax_i$, $c_i = Cx_i$ a key-value store

Query embedding: $u = Bq$
Attention: $p_i = \text{softmax}(u^T m_i)$

Output representation: $o = \sum_i p_i c_i$
Prediction: $\hat{a} = \text{softmax}\big(W(o + u)\big)$

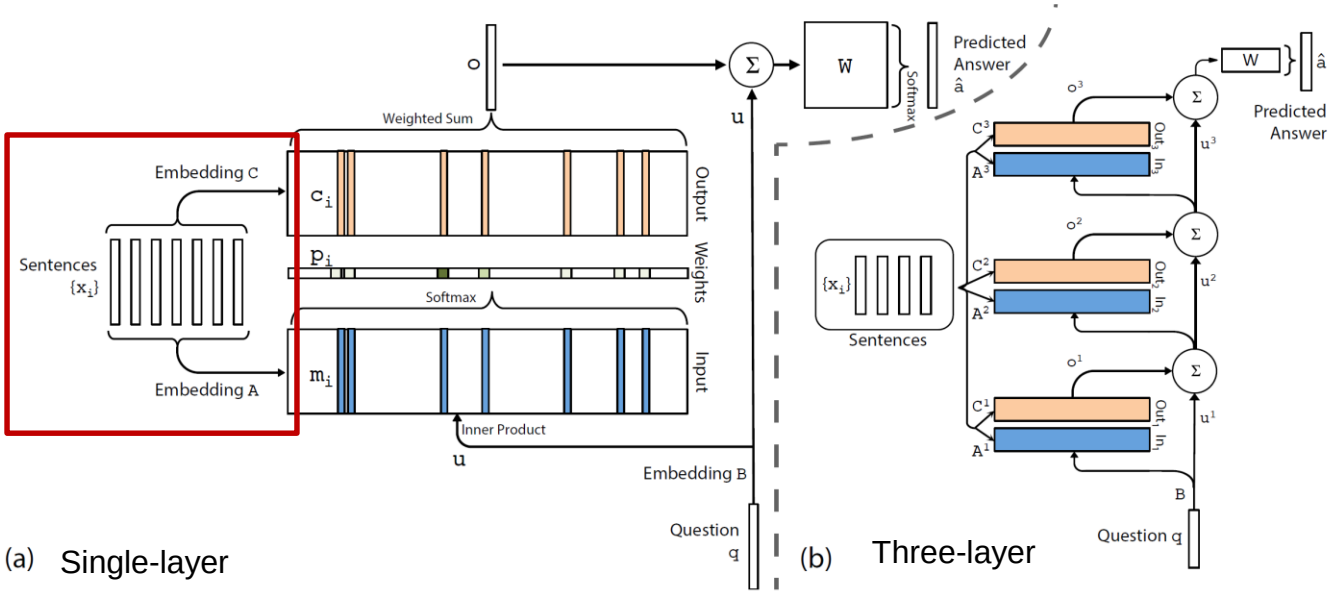
(no memory updating)



Q&A Memory Network

Sukbhaatar et al “End-to-End Memory Networks” 2015

Input data: $m_i = Ax_i$, $c_i = Cx_i$ a key-value store

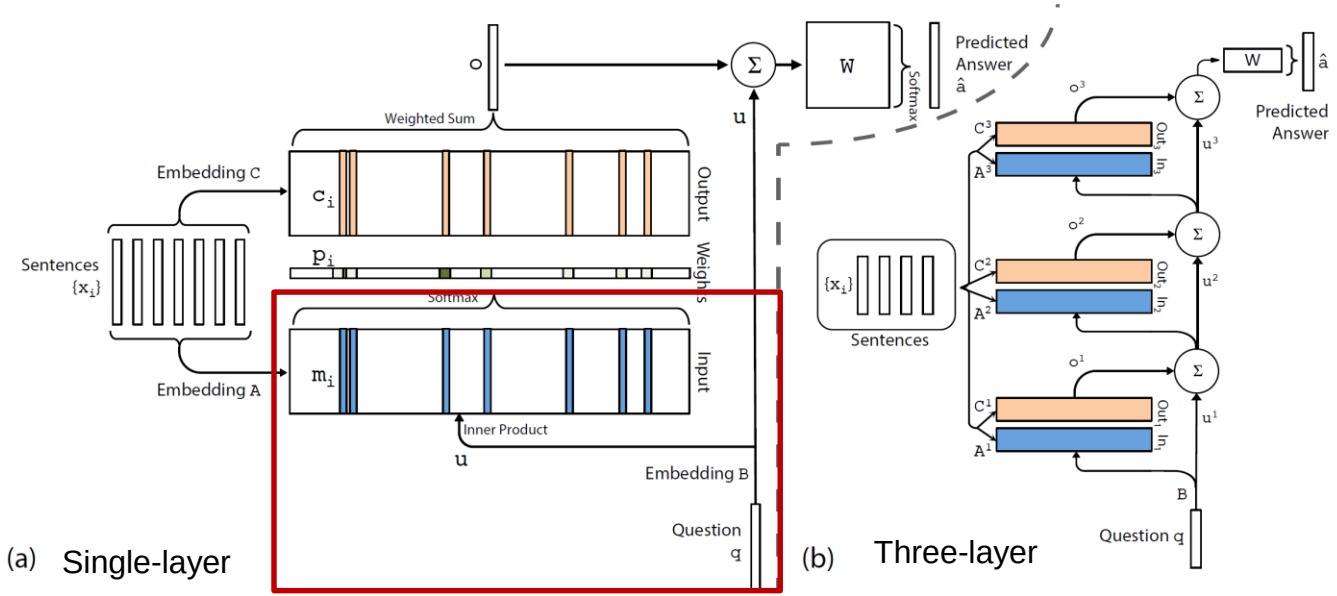


Q&A Memory Network

Sukbhaatar et al “End-to-End Memory Networks” 2015

Query embedding: $u = Bq$

Attention: $p_i = \text{softmax}(u^T m_i)$

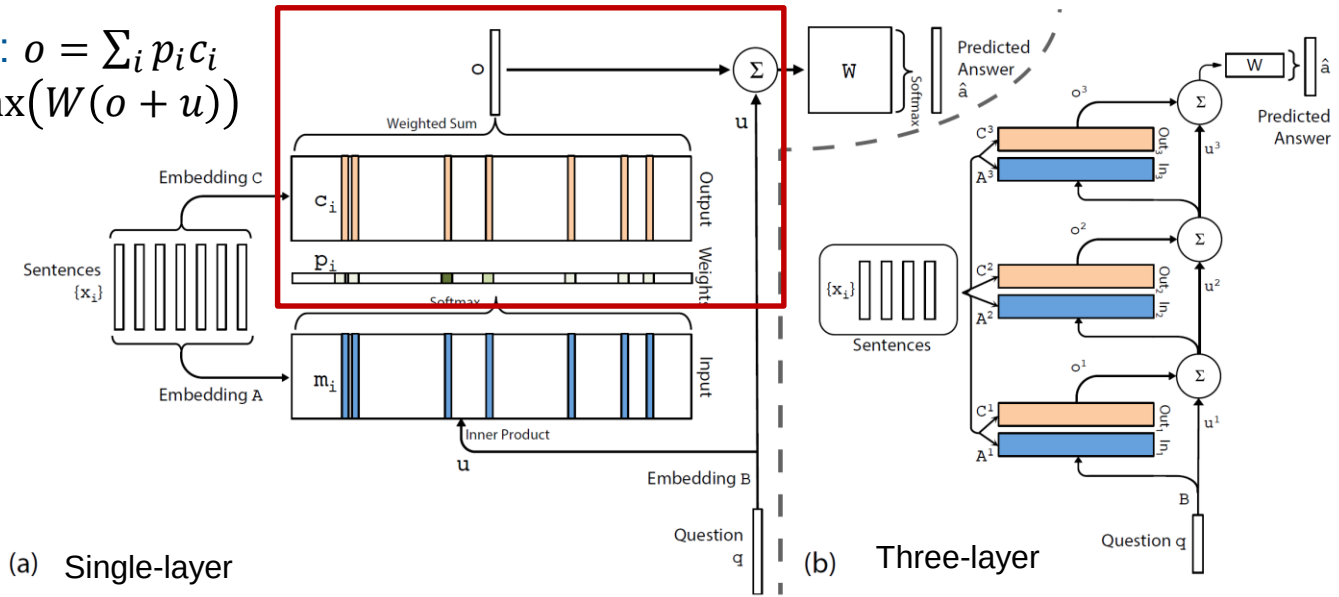


Q&A Memory Network

Sukbhaatar et al “End-to-End Memory Networks” 2015

Output representation: $o = \sum_i p_i c_i$

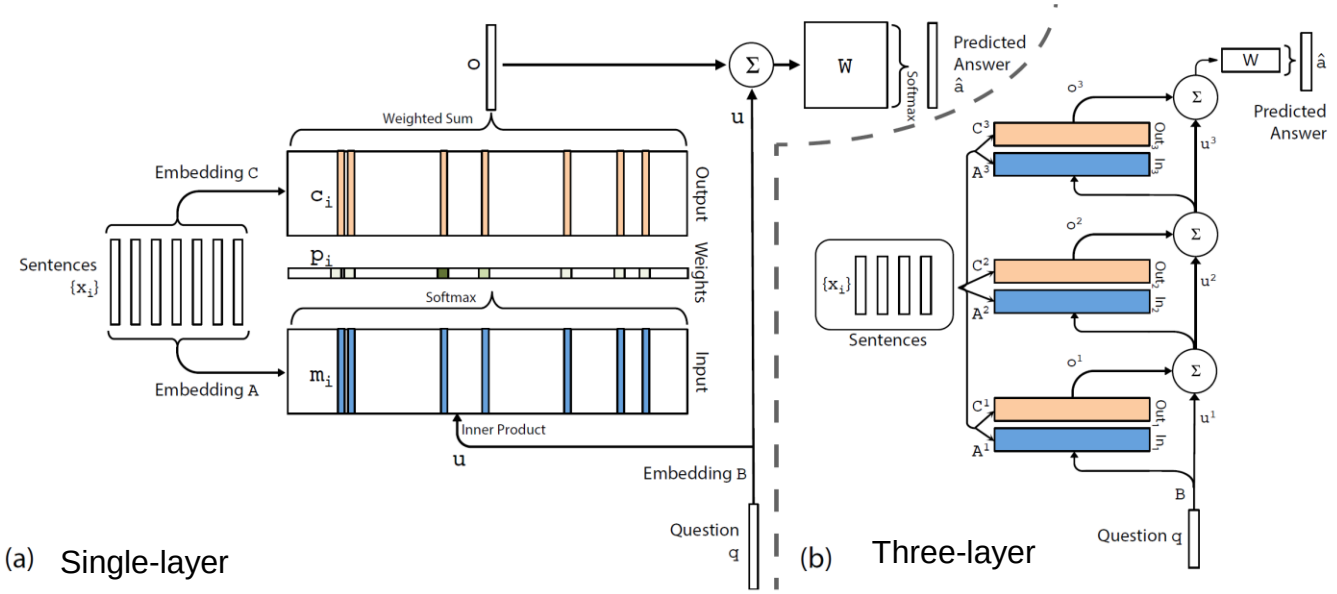
Prediction: $\hat{a} = \text{softmax}\big(W(o + u)\big)$



Q&A Memory Network

Sukbhaatar et al “End-to-End Memory Networks” 2015

Model is trained end-to-end on a series of Assertions and Questions, learns A, B, C and W.



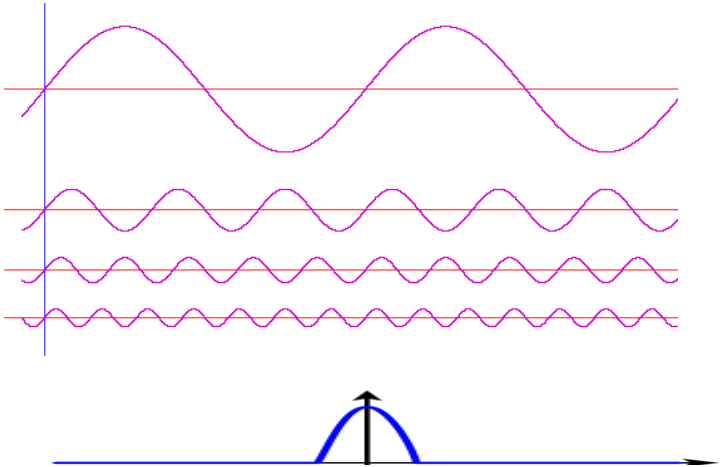
Aside: Position Encoding

Many models, including Transformer and memory networks, use position encoding so that embedded words carry information about their location in the input.

Memory networks multiply input words by a linear function of position.

The Transformer uses a vector of sinusoids which is appended to the input vector.

The advantage of this representation is that the model can learn a linear combination of the sinusoids that is strongest at any particular word position, or a range of positions.



Multi-Hop Inference

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				

Note: Answers are single-word, predicted by the output softmax

Performance on Q&A (babI) tasks

	Baseline			MemN2N								
Task	Strongly Supervised MemNN [22]	LSTM [22]	MemNN WSH	BoW	PE	PE LS	PE LS RN	1 hop PE LS joint	2 hops PE LS joint	3 hops PE LS joint	PE LS RN joint	PE LS LW joint
1: 1 supporting fact	0.0	50.0	0.1	0.6	0.1	0.2	0.0	0.8	0.0	0.1	0.0	0.1
2: 2 supporting facts	0.0	80.0	42.8	17.6	21.6	12.8	8.3	62.0	15.6	14.0	11.4	18.8
3: 3 supporting facts	0.0	80.0	76.4	71.0	64.2	58.8	40.3	76.9	31.6	33.1	21.9	31.7
4: 2 argument relations	0.0	39.0	40.3	32.0	3.8	11.6	2.8	22.8	2.2	5.7	13.4	17.5
5: 3 argument relations	2.0	30.0	16.3	18.3	14.1	15.7	13.1	11.0	13.4	14.8	14.4	12.9
6: yes/no questions	0.0	52.0	51.0	8.7	7.9	8.7	7.6	7.2	2.3	3.3	2.8	2.0
7: counting	15.0	51.0	36.1	23.5	21.6	20.3	17.3	15.9	25.4	17.9	18.3	10.1
8: lists/sets	9.0	55.0	37.8	11.4	12.6	12.7	10.0	13.2	11.7	10.1	9.3	6.1
9: simple negation	0.0	36.0	35.9	21.1	23.3	17.0	13.2	5.1	2.0	3.1	1.9	1.5
10: indefinite knowledge	2.0	56.0	68.7	22.8	17.4	18.6	15.1	10.6	5.0	6.6	6.5	2.6
11: basic coreference	0.0	38.0	30.0	4.1	4.3	0.0	0.9	8.4	1.2	0.9	0.3	3.3
12: conjunction	0.0	26.0	10.1	0.3	0.3	0.1	0.2	0.4	0.0	0.3	0.1	0.0
13: compound coreference	0.0	6.0	19.7	10.5	9.9	0.3	0.4	6.3	0.2	1.4	0.2	0.5
14: time reasoning	1.0	73.0	18.3	1.3	1.8	2.0	1.7	36.9	8.1	8.2	6.9	2.0
15: basic deduction	0.0	79.0	64.8	24.3	0.0	0.0	0.0	46.4	0.5	0.0	0.0	1.8
16: basic induction	0.0	77.0	50.5	52.0	52.1	1.6	1.3	47.4	51.3	3.5	2.7	51.0
17: positional reasoning	35.0	49.0	50.9	45.4	50.1	49.0	51.0	44.4	41.2	44.5	40.4	42.6
18: size reasoning	5.0	48.0	51.3	48.1	13.6	10.1	11.1	9.6	10.3	9.2	9.4	9.2
19: path finding	64.0	92.0	100.0	89.7	87.4	85.6	82.8	90.7	89.9	90.2	88.0	90.6
20: agent's motivation	0.0	9.0	3.6	0.1	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.2
Mean error (%)	6.7	51.3	40.2	25.1	20.3	16.3	13.9	25.8	15.6	13.3	12.4	15.2
Failed tasks (err. $>$ 5%)	4	20	18	15	13	12	11	17	11	11	11	10
On 10k training data												
Mean error (%)	3.2	36.4	39.2	15.4	9.4	7.2	6.6	24.5	10.9	7.9	7.5	11.0
Failed tasks (err. $>$ 5%)	2	16	17	9	6	4	4	16	7	6	6	6

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

Performance Improves with Number of Hops

	Baseline			MemN2N								
Task	Strongly Supervised MemNN [22]	LSTM [22]	MemNN WSH	BoW	PE	PE LS	PE LS RN	1 hop PE LS joint	2 hops PE LS joint	3 hops PE LS joint	PE LS RN joint	PE LS LW joint
1: 1 supporting fact	0.0	50.0	0.1	0.6	0.1	0.2	0.0	0.8	0.0	0.1	0.0	0.1
2: 2 supporting facts	0.0	80.0	42.8	17.6	21.6	12.8	8.3	62.0	15.6	14.0	11.4	18.8
3: 3 supporting facts	0.0	80.0	76.4	71.0	64.2	58.8	40.3	76.9	31.6	33.1	21.9	31.7
4: 2 argument relations	0.0	39.0	40.3	32.0	3.8	11.6	2.8	22.8	2.2	5.7	13.4	17.5
5: 3 argument relations	2.0	30.0	16.3	18.3	14.1	15.7	13.1	11.0	13.4	14.8	14.4	12.9
6: yes/no questions	0.0	52.0	51.0	8.7	7.9	8.7	7.6	7.2	2.3	3.3	2.8	2.0
7: counting	15.0	51.0	36.1	23.5	21.6	20.3	17.3	15.9	25.4	17.9	18.3	10.1
8: lists/sets	9.0	55.0	37.8	11.4	12.6	12.7	10.0	13.2	11.7	10.1	9.3	6.1
9: simple negation	0.0	36.0	35.9	21.1	23.3	17.0	13.2	5.1	2.0	3.1	1.9	1.5
10: indefinite knowledge	2.0	56.0	68.7	22.8	17.4	18.6	15.1	10.6	5.0	6.6	6.5	2.6
11: basic coreference	0.0	38.0	30.0	4.1	4.3	0.0	0.9	8.4	1.2	0.9	0.3	3.3
12: conjunction	0.0	26.0	10.1	0.3	0.3	0.1	0.2	0.4	0.0	0.3	0.1	0.0
13: compound coreference	0.0	6.0	19.7	10.5	9.9	0.3	0.4	6.3	0.2	1.4	0.2	0.5
14: time reasoning	1.0	73.0	18.3	1.3	1.8	2.0	1.7	36.9	8.1	8.2	6.9	2.0
15: basic deduction	0.0	79.0	64.8	24.3	0.0	0.0	0.0	46.4	0.5	0.0	0.0	1.8
16: basic induction	0.0	77.0	50.5	52.0	52.1	1.6	1.3	47.4	51.3	3.5	2.7	51.0
17: positional reasoning	35.0	49.0	50.9	45.4	50.1	49.0	51.0	44.4	41.2	44.5	40.4	42.6
18: size reasoning	5.0	48.0	51.3	48.1	13.6	10.1	11.1	9.6	10.3	9.2	9.4	9.2
19: path finding	64.0	92.0	100.0	89.7	87.4	85.6	82.8	90.7	89.9	90.2	88.0	90.6
20: agent's motivation	0.0	9.0	3.6	0.1	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.2
Mean error (%)	6.7	51.3	40.2	25.1	20.3	16.3	13.9	25.8	15.6	13.3	12.4	15.2
Failed tasks (err. $>$ 5%)	4	20	18	15	13	12	11	17	11	11	11	10
On 10k training data												
Mean error (%)	3.2	36.4	39.2	15.4	9.4	7.2	6.6	24.5	10.9	7.9	7.5	11.0
Failed tasks (err. $>$ 5%)	2	16	17	9	6	4	4	16	7	6	6	6

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

Memory Networks and Natural Language

The memory network considered so far is designed to work with structured text documents (a knowledge base or KB).

It can be extended to deal with Natural language text, and performance on the two types of data source can be compared.

The domain is movie knowledge.

Miller et al. “Key-Value Memory Networks for Directly Reading Documents” 2016

Memory Networks and Natural Language

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, . . .

After running an IE (Information Extraction) pipeline on the full text we build this table:

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

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?

Miller et al. "Key-Value Memory Networks for Directly Reading Documents" 2016

Memory Network Key-Value Store

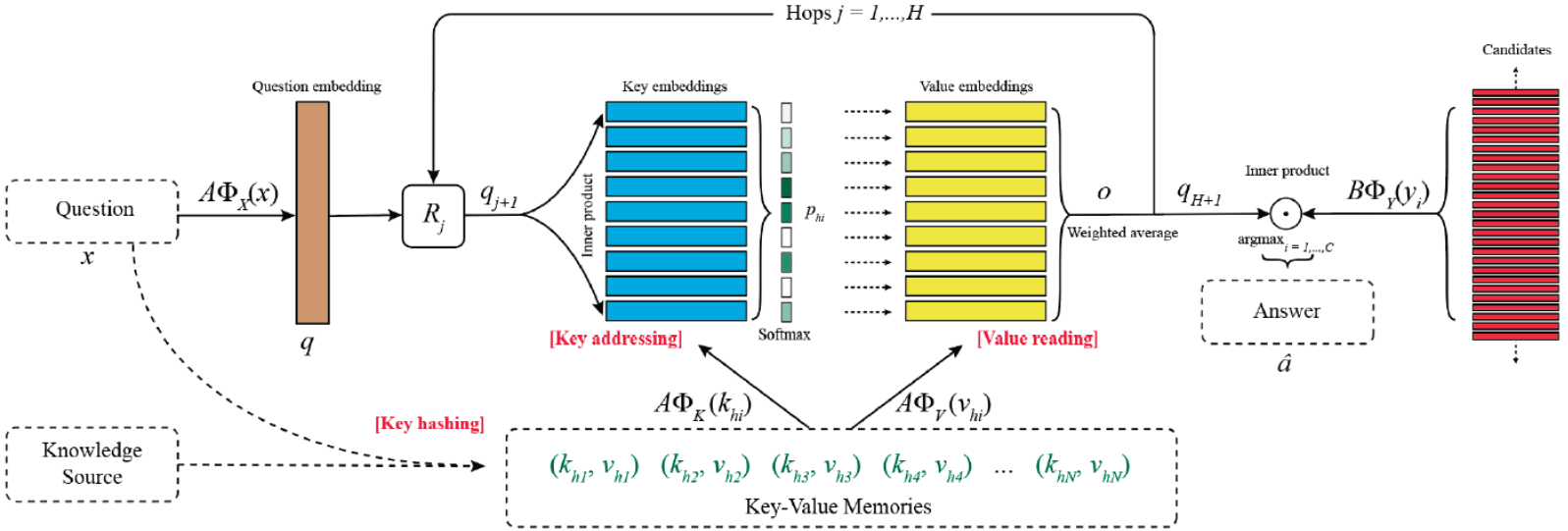
Input feature maps: $\Phi_K(k_i)$ and $\Phi_V(v_i)$ - key, value embeddings (k_i, v_i different this time)

Query: x

Attention: $p_i = \text{softmax}(A\Phi_X(x) \cdot A\Phi_K(k_{h_i}))$

Output representation: $o = \sum_i p_{h_i} A\Phi\_V(v_{h_i})$

Prediction: $\hat{a} = \text{argmax}_{i=1,\ldots,C} \text{Softmax}(q_{H+1}^\top B\Phi_Y(y_i))$



Memory Networks: Using the Key-Value Store

Sentence-Level Encoding: Free-text input is broken into sentences, and each is encoded in BoW as key and value – equivalent to standard MemNN.

Window-Level: Encode a window of W words in BOW as the key. Use the center word as the value.

KB-Triple: Typically have the form “subject relation object,” key is subject-relation pair, value is the object.
For better retrieval relations are typically encoded twice, e.g.:

Blade Runner directed_by Ridley Scott
Ridley Scott !directed_by Blade Runner

Question Standardization

Original natural language questions were standardized using the SimpleQuestions dataset.

“What movies did Harrison Ford star in?”

➔

Instance of the pattern “What movies did [@actor] star in?”

Created 100k training pairs.

Scaling Up

Its impractical to test queries against the entire database.

Instead the query text is used to perform full-text search across the database.

Only document that are similar enough to the query (e.g. contain at least one query word) are actually considered.

Results!

Question Type	KB	IE	Doc
Writer to Movie	97	72	91
Tag to Movie	85	35	49
Movie to Year	95	75	89
Movie to Writer	95	61	64
Movie to Tags	94	47	48
Movie to Language	96	62	84
Movie to IMDb Votes	92	92	92
Movie to IMDb Rating	94	75	92
Movie to Genre	97	84	86
Movie to Director	93	76	79
Movie to Actors	91	64	64
Director to Movie	90	78	91
Actor to Movie	93	66	83

Table 4: Breakdown of test results (% hits@1) on WIKIMOVIES for Key-Value Memory Networks using different knowledge representations.

Results!

Reading raw docs (Doc column) usually does much better than the doc-extracted KB (IE column).

Structed KBs (KB column) often better than the document generated answer.

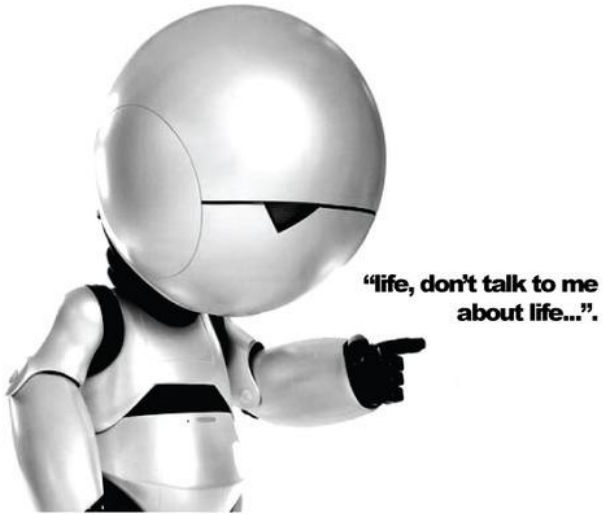
The experiment was done on a subset of questions that were in the KB. Many possible questions are not.

Moral: Use KB when possible, fall back on free text.

Question Type	KB	IE	Doc
Writer to Movie	97	72	91
Tag to Movie	85	35	49
Movie to Year	95	75	89
Movie to Writer	95	61	64
Movie to Tags	94	47	48
Movie to Language	96	62	84
Movie to IMDb Votes	92	92	92
Movie to IMDb Rating	94	75	92
Movie to Genre	97	84	86
Movie to Director	93	76	79
Movie to Actors	91	64	64
Director to Movie	90	78	91
Actor to Movie	93	66	83

Table 4: Breakdown of test results (% hits@1) on WIKIMOVIES for Key-Value Memory Networks using different knowledge representations.

Dialog



Goal-Directed Dialog

Goal-Directed Dialog Systems, in contrast to chatbots, aim to not only engage the user, but help the user with goal-directed tasks.

Traditional dialog systems use slot-filling:
“We’d like a table for two at 8pm, outside if possible”

Fills slots for
- Number of people
- Time
- Location preference

The interaction may need several turns to:
- Clarify users intention (slot doesn’t match)
- Ask for a different option (request can’t be met)
- Fill in missing slots

Machine Learning Goal-Directed Dialog

The idea is to learn from sample dialog how to respond to user queries.

Task Summary

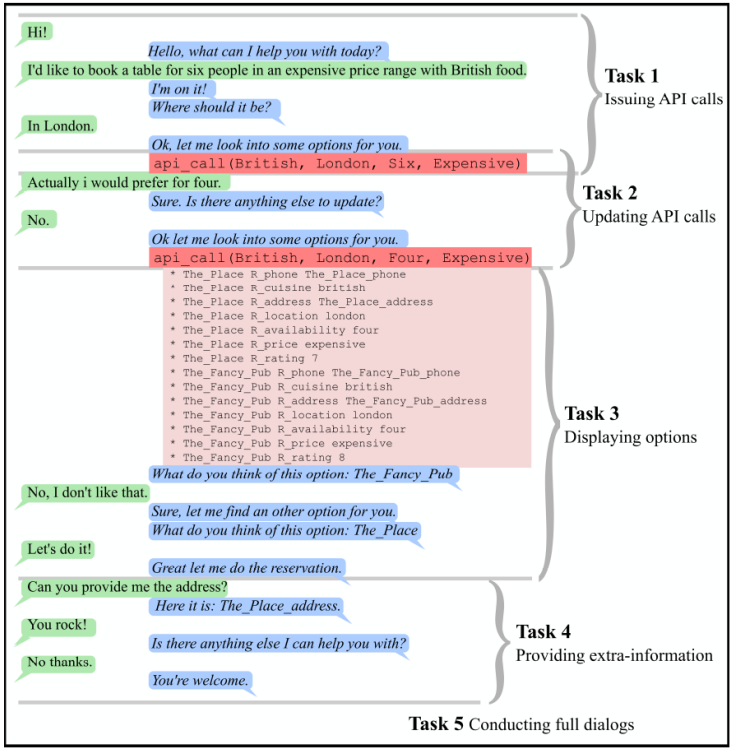
Task 1: Issuing API calls. A user query contains from 0 to 4 of the required fields. Agent must ask questions for filling the missing.

Task 2: Updating API calls. After an API call fromTask 1, user asks to update their requests. The agent ask user if they are done and issue the updated API call.

Task 3: Displaying options. Given a user request, we query the KB to get possible responses.

Task 4: Providing extra information. Users then ask for the phone number of the restaurant, its address or both.

Task 5: Conducting full dialogs We combine Tasks 1-4 to generate full dialogs just as in Figure 1.



Bordes et al “LEARNING END-TO-END GOAL-ORIENTED DIALOG” ICML 2017

Datasets

Restaurant Reservations: Contains two KBs of 4,200 facts and 600 restaurants each (5 types of cuisine 5 locations 3 price ranges 8 ratings). Use one of the KBs to generate the standard training, validation and test dialogs, and use the other KB only to generate test dialogs, termed Out-Of-Vocabulary (OOV) test sets.

Dialog State Tracking Challenge: Another restaurant booking dataset, but using data from real users. We use data from DSTC2 (Henderson et al., 2014a), which was designed for dialog state tracking hence every dialog turn is labeled with a state (a user intent + slots) to be predicted.

Online Concierge Service: Data extracted from a real online concierge service: users make requests through a text-based chat interface that are handled by human operators who can make API calls.

Bordes et al “LEARNING END-TO-END GOAL-ORIENTED DIALOG” ICML 2017

Experiments

Task	Rule-based Systems	TF-IDF Match no type	TF-IDF Match + type	Nearest Neighbor	Supervised Embeddings	Memory Networks no match type	Memory Networks + match type
T1: Issuing API calls	*100 (100)*	5.6 (0)	22.4 (0)	55.1 (0)	**100** (100)	**99.9** (99.6)	**100** (100)
T2: Updating API calls	*100 (100)*	3.4 (0)	16.4 (0)	68.3 (0)	68.4 (0)	**100** (100)	98.3 (83.9)
T3: Displaying options	*100 (100)*	8.0 (0)	8.0 (0)	58.8 (0)	64.9 (0)	**74.9** (2.0)	**74.9** (0)
T4: Providing information	*100 (100)*	9.5 (0)	17.8 (0)	28.6 (0)	57.2 (0)	59.5 (3.0)	**100** (100)
T5: Full dialogs	*100 (100)*	4.6 (0)	8.1 (0)	57.1 (0)	75.4 (0)	**96.1** (49.4)	93.4 (19.7)
T1(OOV): Issuing API calls	*100 (100)*	5.8 (0)	22.4 (0)	44.1 (0)	60.0 (0)	72.3 (0)	**96.5** (82.7)
T2(OOV): Updating API calls	*100 (100)*	3.5 (0)	16.8 (0)	68.3 (0)	68.3 (0)	78.9 (0)	**94.5** (48.4)
T3(OOV): Displaying options	*100 (100)*	8.3 (0)	8.3 (0)	58.8 (0)	65.0 (0)	74.4 (0)	**75.2** (0)
T4(OOV): Providing inform.	*100 (100)*	9.8 (0)	17.2 (0)	28.6 (0)	57.0 (0)	57.6 (0)	**100** (100)
T5(OOV): Full dialogs	*100 (100)*	4.6 (0)	9.0 (0)	48.4 (0)	58.2 (0)	65.5 (0)	**77.7** (0)
T6: Dialog state tracking 2	33.3 (0)	1.6 (0)	1.6 (0)	21.9 (0)	22.6 (0)	**41.1** (0)	**41.0** (0)
Concierge(*)	n/a	1.1 (0.2)	n/a	13.4 (0.5)	14.6 (0.5)	**16.7** (1.2)	n/a(†)

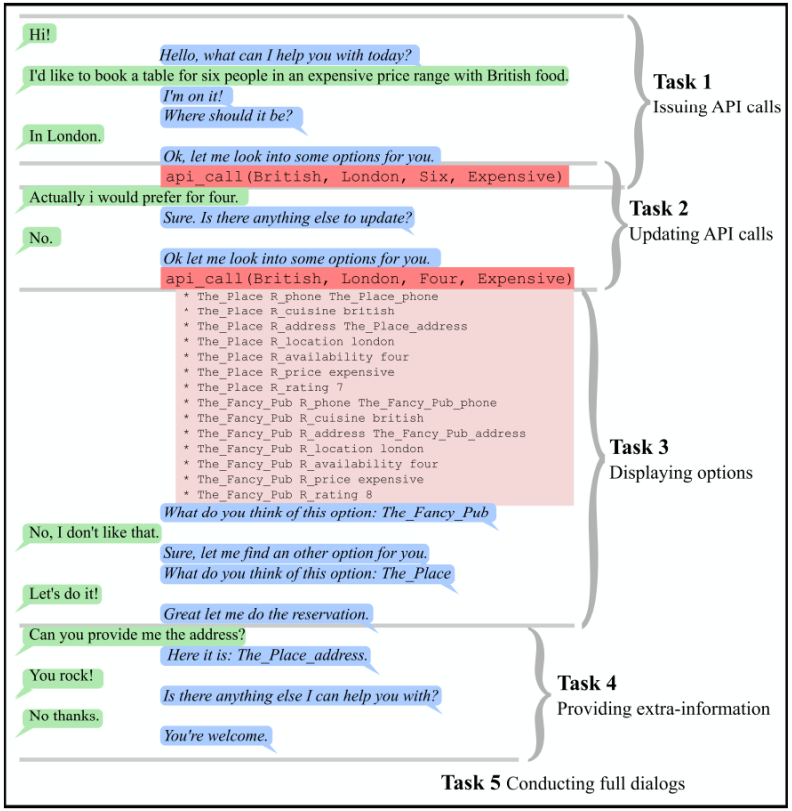
Match type = extend entity descriptions with their type (cuisine type, location, price range, party size, rating, phone number and address) to help match OOV items.
Supervised embedding = task-specific word embedding, using a margin loss on a prediction task.

Bordes et al “LEARNING END-TO-END GOAL-ORIENTED DIALOG” ICML 2017

Experiments

Ground Truth = held out transcripts.

Note “display options” task includes user feedback:



Dynamic Memory Networks

Similarities:

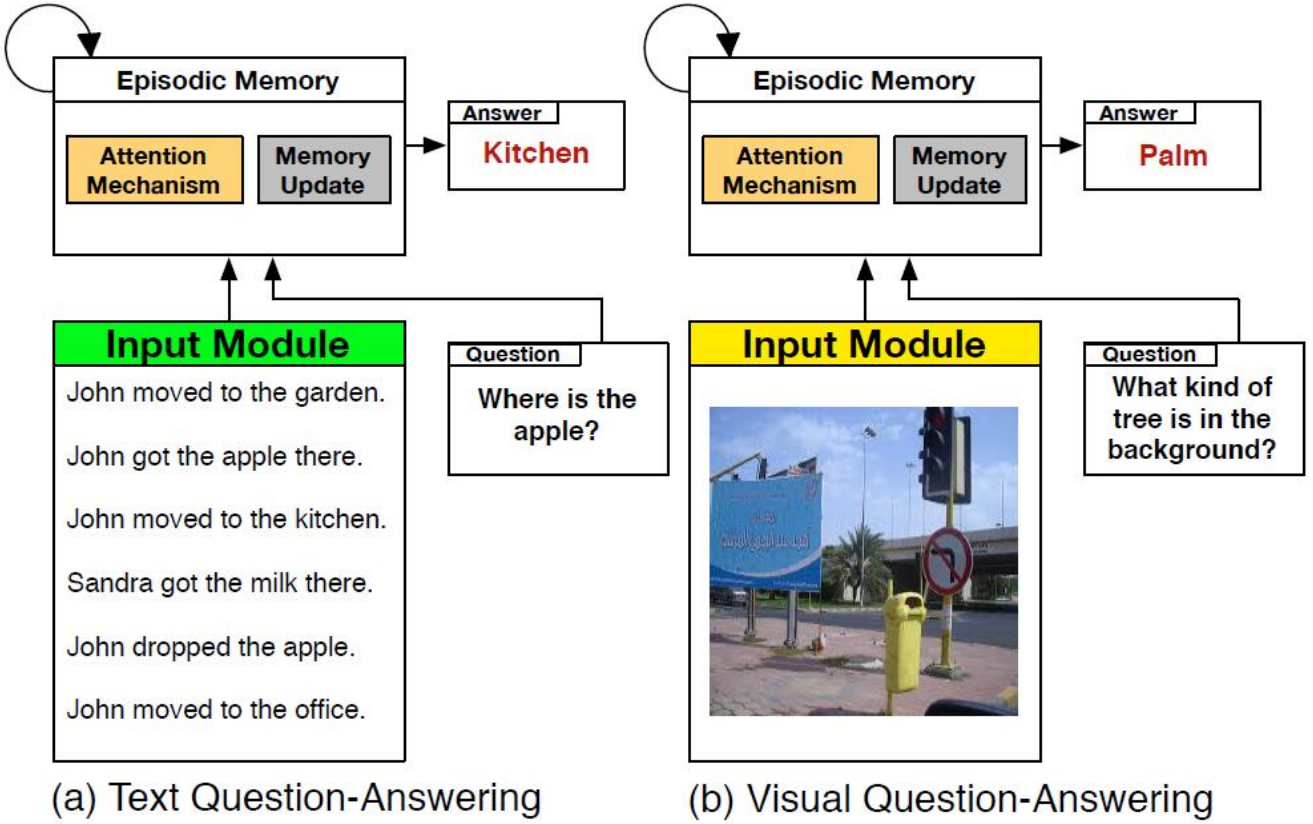
- MemNets and DMNs have input, scoring, attention and response mechanisms.

Differences:

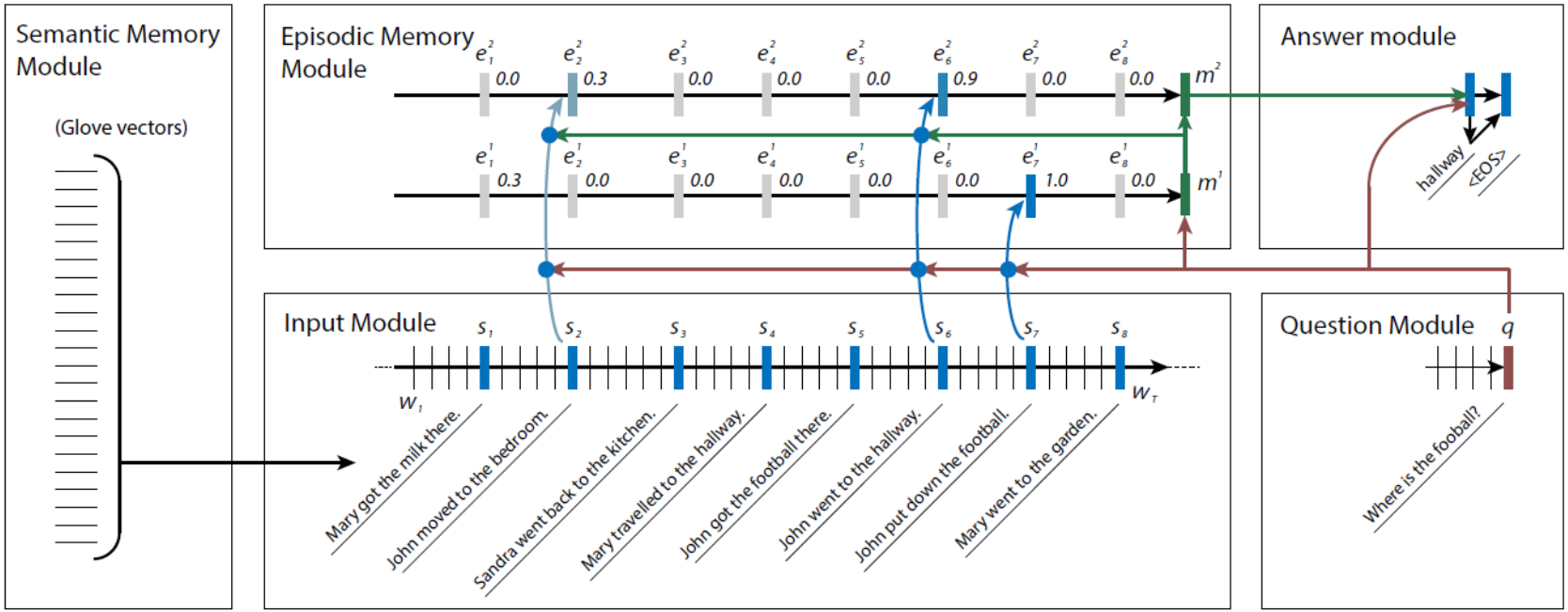
- For input representations MemNets use bag of word, nonlinear or linear embeddings that explicitly encode position.
- MemNets iteratively run functions for attention and response.

- Dynamic Memory Networks use recurrent networks for input encoding, attention and response generation.

Dynamic Memory Networks

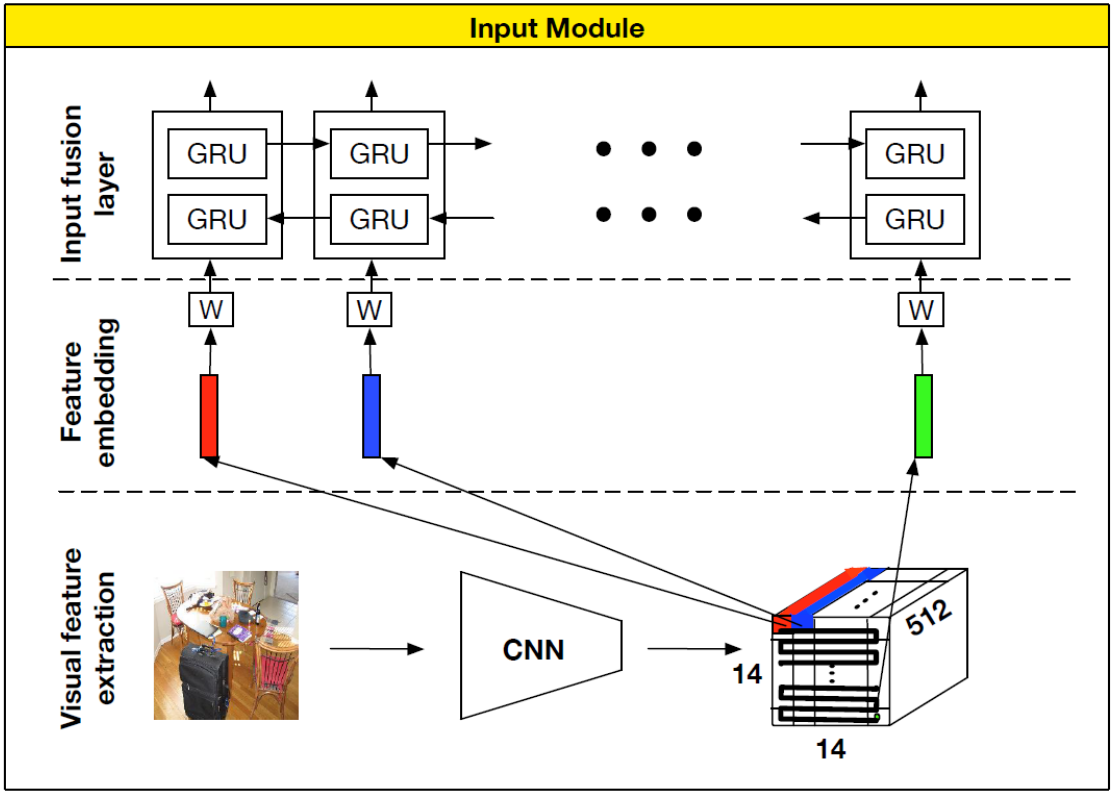


Dynamic Memory Networks



Really very similar to MemNets, except that hops are managed by a recurrent network, instead of fixed logic.

Dynamic Memory Networks: Image Input



Dynamic Memory Network Attention

What is the main color on the bus ?

Answer: blue

What type of trees are in the background ?

Answer: pine

How many pink flags are there ?

Answer: 2

Is this in the wild ?

Answer: no

Dynamic Memory Network Attention

Which man is dressed more flamboyantly ?

Answer: right

Who is on both photos ?

Answer: girl

What time of day was this picture taken ?

Answer: night

What is the boy holding ?

Answer: surfboard

Dynamic Memory Network Attention

What is this sculpture made out of ?

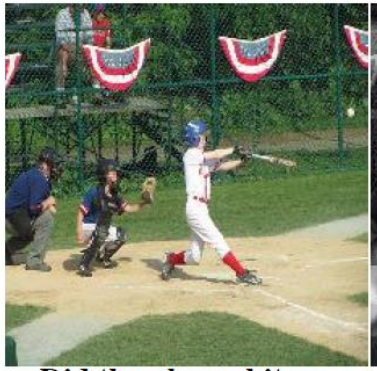
Answer: metal

What color are the bananas ?

Answer: green

What is the pattern on the cat ' s fur on its tail ?

Answer: stripes



Did the player hit the ball ?

Answer: yes

Dynamic Memory Network Results

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

Textual Question Answering (babI dataset)

	test-dev				test-std
Method	All	Y/N	Other	Num	All
VQA					
Image	28.1	64.0	3.8	0.4	-
Question	48.1	75.7	27.1	36.7	-
Q+I	52.6	75.6	37.4	33.7	-
LSTM Q+I	53.7	78.9	36.4	35.2	54.1
ACK	55.7	79.2	40.1	36.1	56.0
iBOWIMG	55.7	76.5	42.6	35.0	55.9
DPPnet	57.2	80.7	41.7	37.2	57.4
D-NMN	57.9	80.5	43.1	37.4	58.0
SAN	58.7	79.3	46.1	36.6	58.9
DMN+	**60.3**	80.5	48.3	36.8	**60.4**

Visual Question Answering

NLP Question Answering

“All NLP/AI tasks can be reduced to Question Answering” – R. Socher

NLP Question Answering

“All NLP/AI tasks can be reduced to Question Answering” – R. Socher

Makes sense if:

- In answering a question “yes”, an agent does what you ask it to…
- You don’t have to interact with the physical world…

But basically, in the world of information, most of our interaction is through speech acts with other actors…

Memory Network Take-aways

- Memory nets include short-term and long-term memory with an indexing (attention) mechanism over long-term (WORM) memory.

- MemNN is like an associative key-value story. Its supports multiple hops to follow inference chains or get up-to-date results.