+

CHULA ENGINEERING
Foundation toward Innovation
COMPUTER

Text Generation & Question Answering

2110594: Natural Language Processing (NLP)

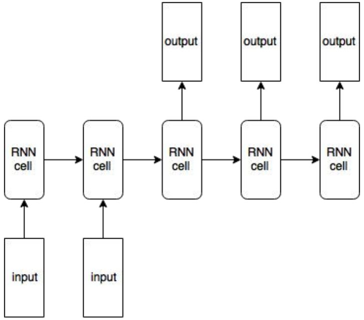
Peerapon Vateekul & Ekapol Chuangsuwanich
Department of Computer Engineering,
Faculty of Engineering, Chulalongkorn University

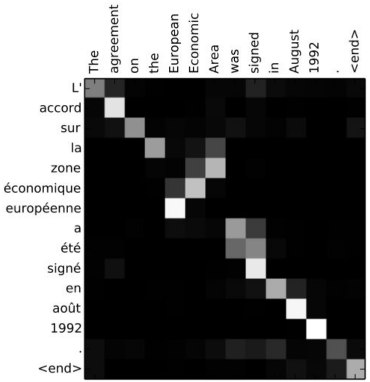
Credit: Can Udomcharoenchaikit & Nattachai Tretasayuth

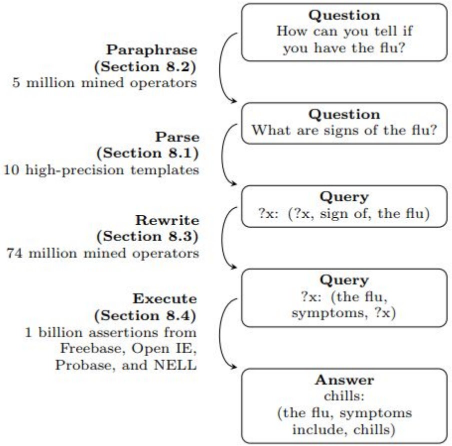
Outline

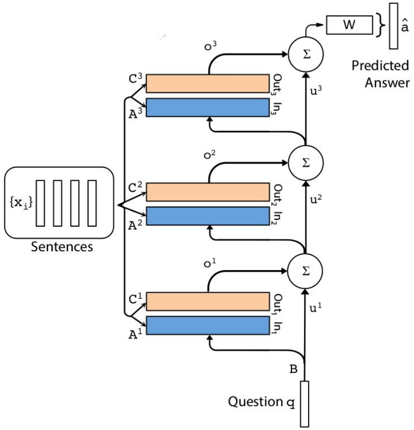
Text Generation & Attention Mechanism

Question Answering and Deep Learning



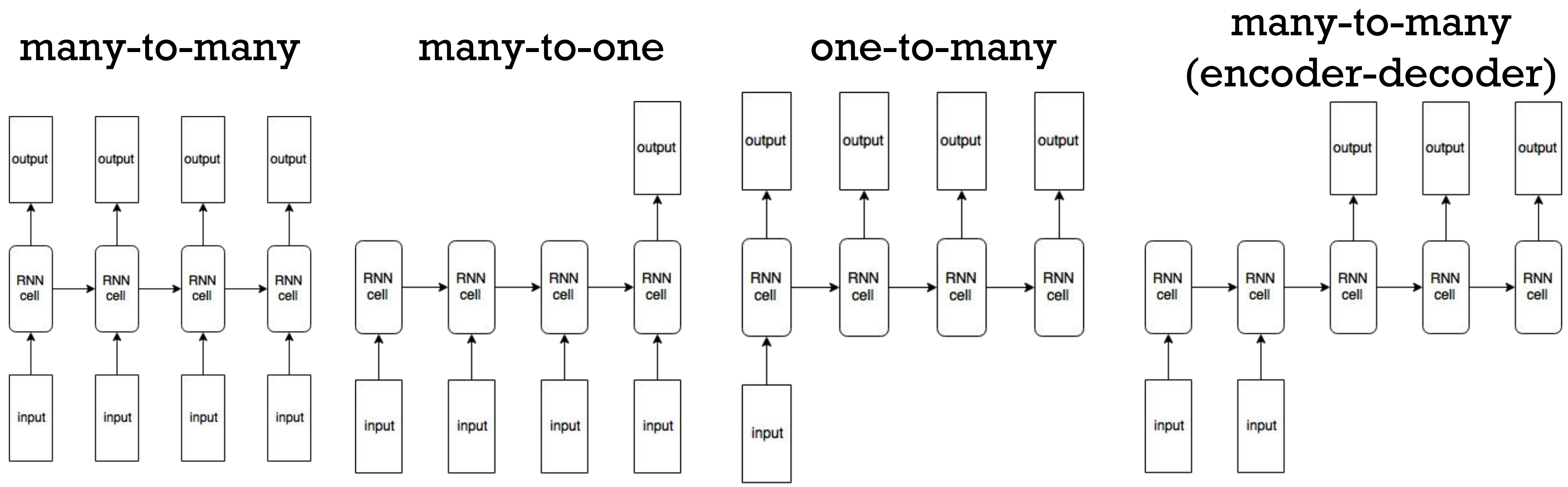






Text Generation & Attention Mechanism

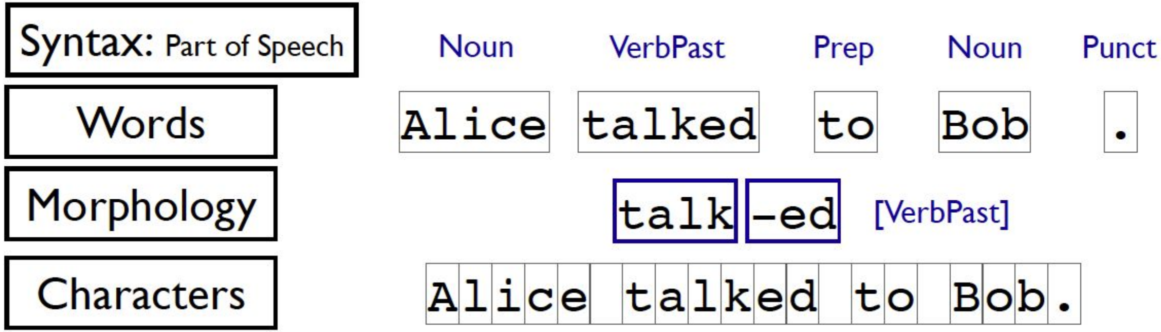
Different types of RNN architectures

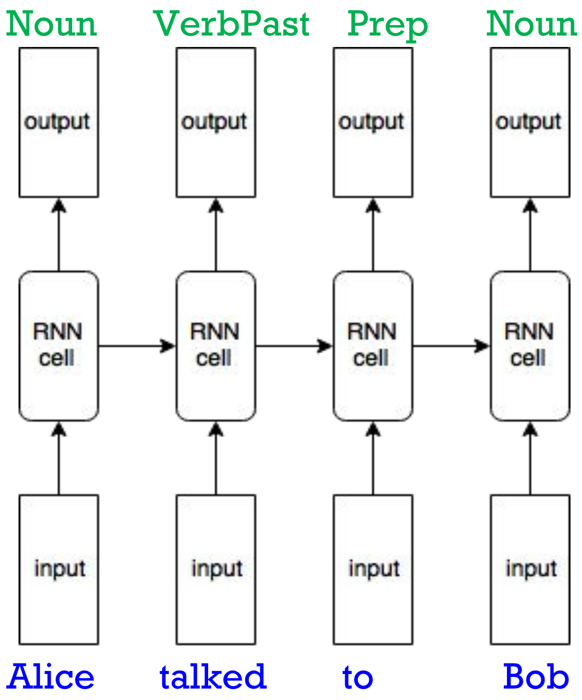


Reference: The Unreasonable Effectiveness of Recurrent Neural Networks, http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Many-to-many

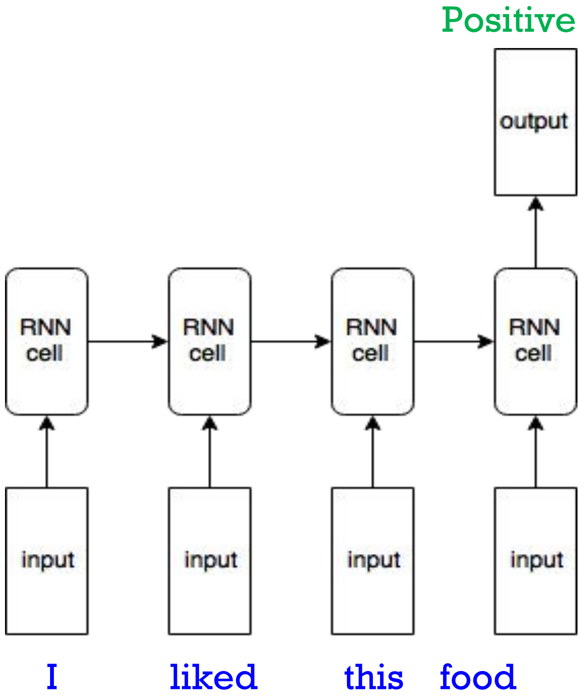
- You have seen and implemented this type of RNN architecture in your homework already.
- E.g. Tokenization, POS tagging
- Sequence Input, Sequence Output





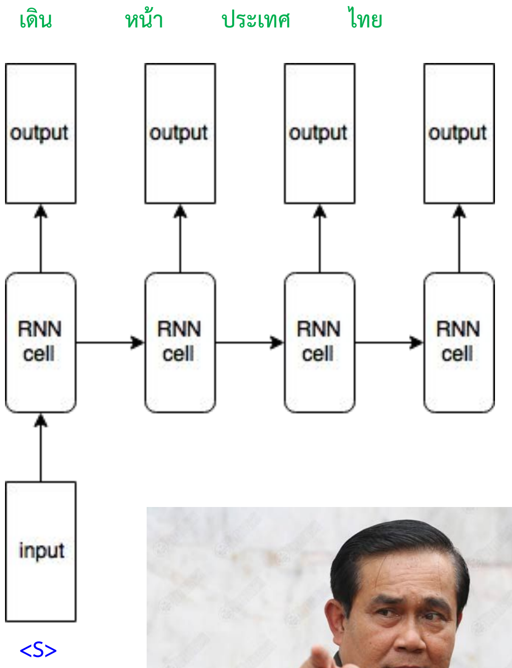
Many-to-one

- You probably just implemented this type of RNN for your take-home exam.
- E.g. Sentiment Analysis, Text classification
- Sequence input



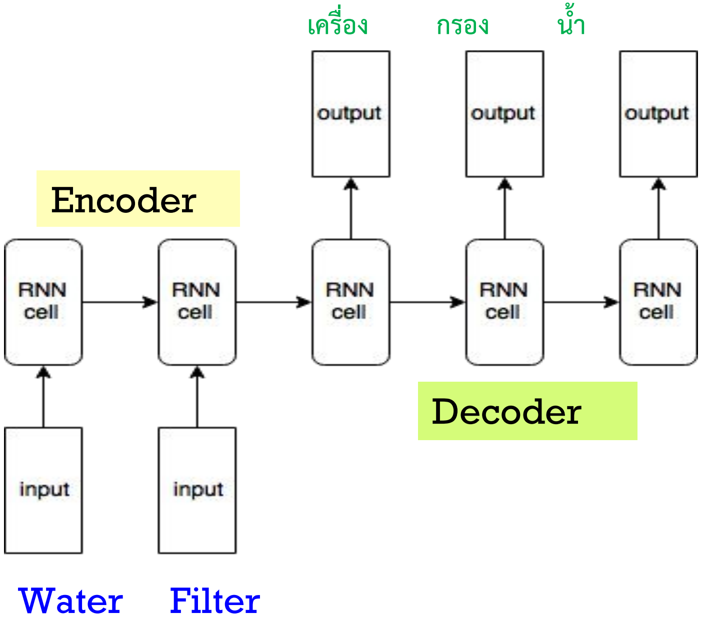
One-to-many

- Sequence output
- E.g. Music Generation, Image caption generation
- Music **generation**
  - Input: Initial seed
  - Output: Sequence of music notes
- Image caption **generation**
  - Input: Image features extracted by CNN
  - Output: Sequence of text



Many-to-many (encoder-decoder)

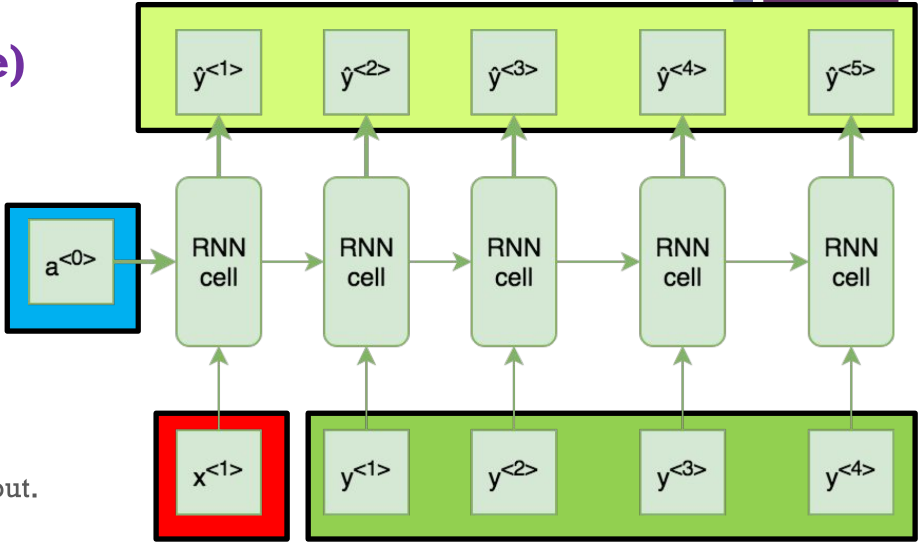
- Sequence Input, Sequence output
- These two sequences can be of different length
- E.g. Machine Translation
  - Input: English Sentence
  - Output: Thai Sentence
- Machine Translation is also a **text generation task**



Text generation model (training)

One-to-Many RNN (autoregressive)

- The only real input is $\mathbf{x}^{<1>}$
- $a^{<0>}$ is the initial hidden state.
- $\hat{y}$ is the predicted output.
- y is an actual output.
- During **the training phase**, instead of using the predicted output to feed into the next time-step, we use the actual output.



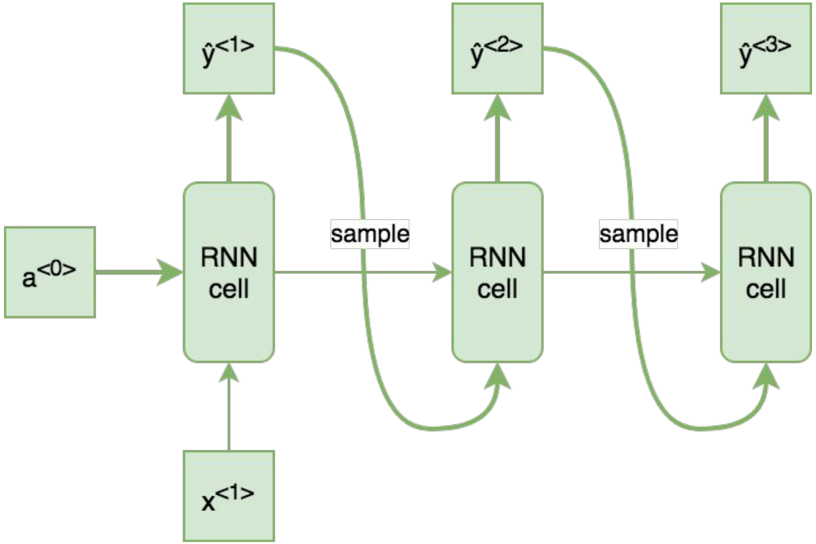
REAL SEQUENCE!!! (y, not ŷ)

$$a^{<t>} = \boxed{Wa^{<t-1>}} + \boxed{Wx^{<t>}} + \boxed{b}$$

Text generation model (inference)

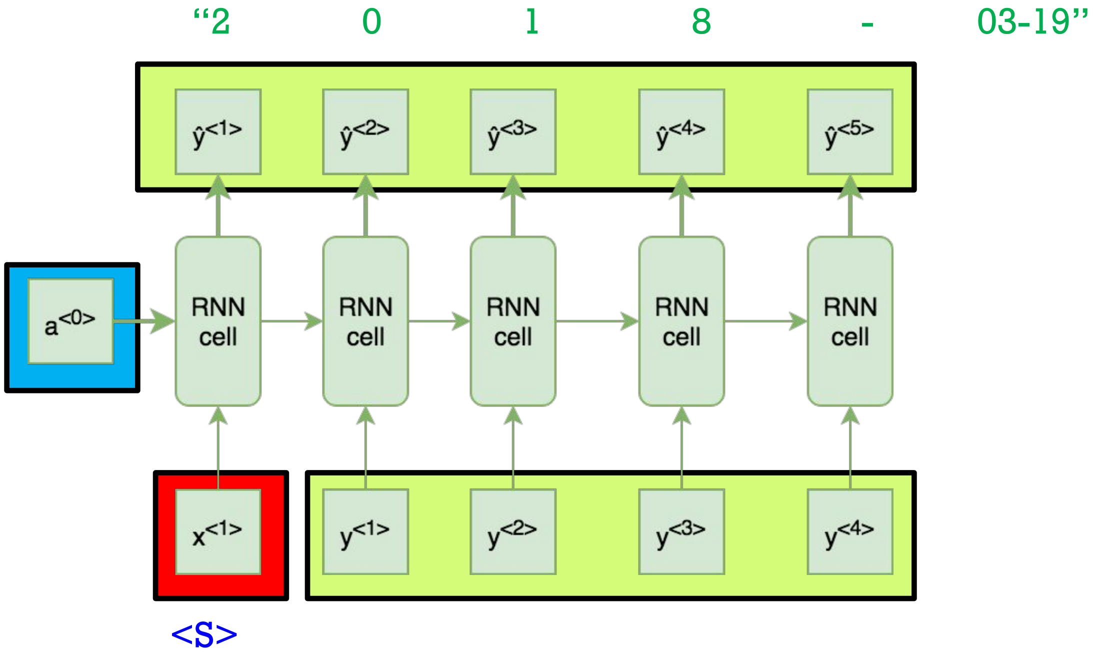
Training → Inference

- To generate a novel sequence, the inference model (testing phase) randomly samples an output from a softmax distribution.



In class demo: Text generation

Simple demo: Generating a piece of text using RNN; Random Date Generation “2018-03-19”

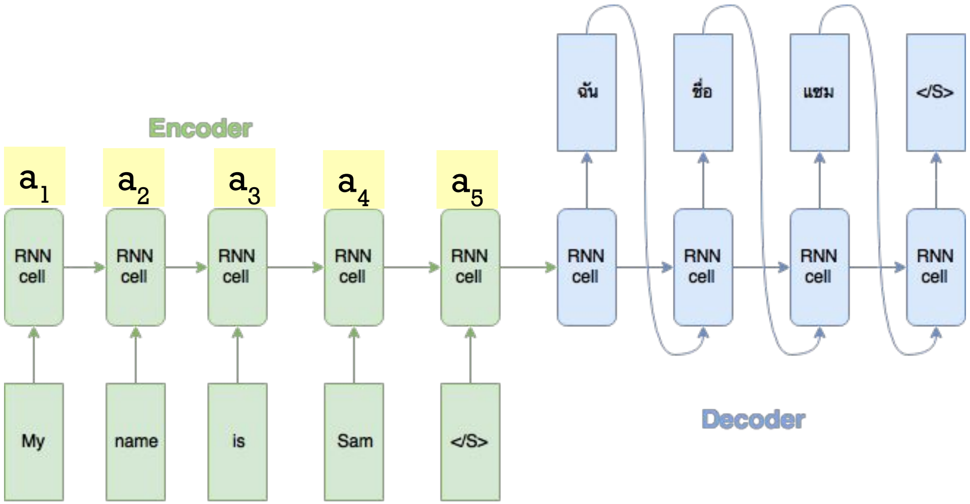


Attention Mechanism (Many-to-Many)

Attention is commonly used in sequence-to-sequence model, it allows **the decoder part** of the network to focus/**attend** on a different part of **the encoder's outputs** for every step of the decoder's own outputs.

Why attention?

This is what we want you to think about: How can information travel from one end to another in neural networks?



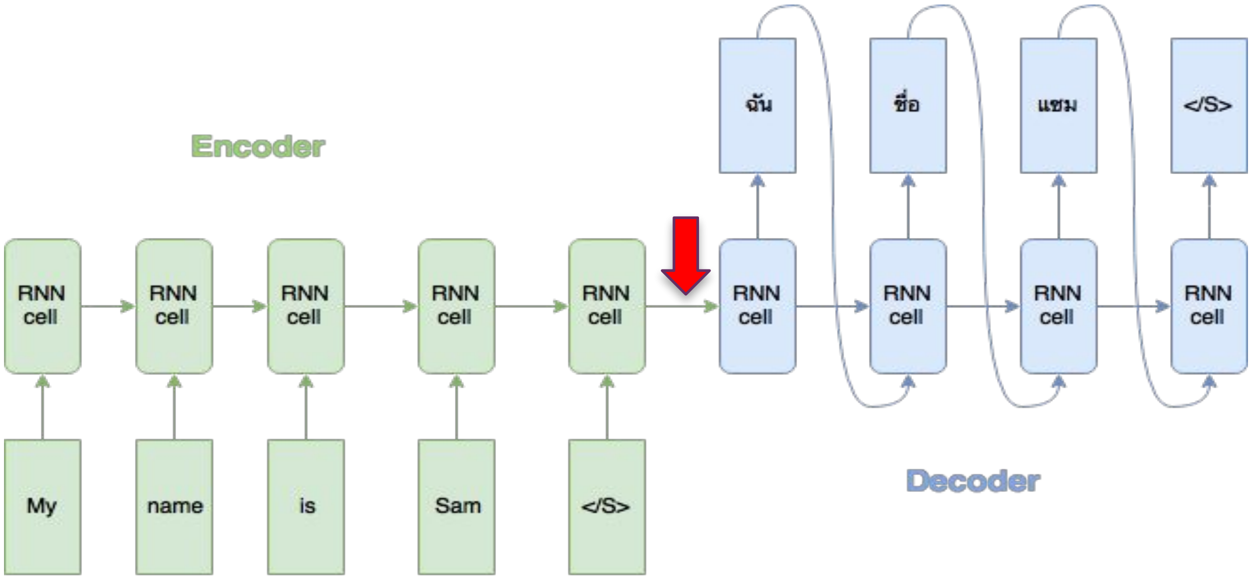
Machine Translation Problem: English to Thai

+

Attention Mechanism (cont.)

Why attention?

"You can't cram the meaning of a whole %&!$# sentence into a single $&!#* vector!" - Raymond Mooney (2014)



Reference:http://yoavartzi.com/sp14/slides/mooney.sp14.pdf

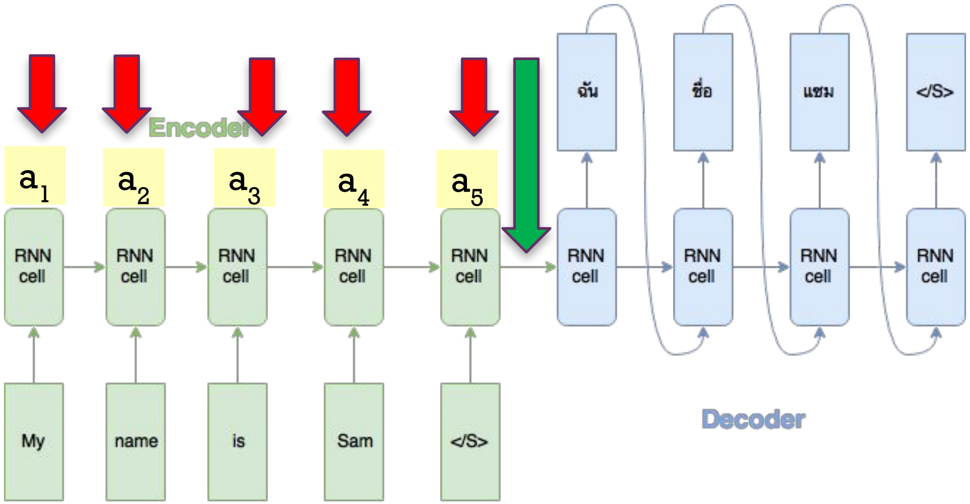
Machine Translation Problem: English to Thai

Attention Mechanism (cont.)

Why attention?

Main idea: We can use multiple vectors based on the length of the sentence instead of one.

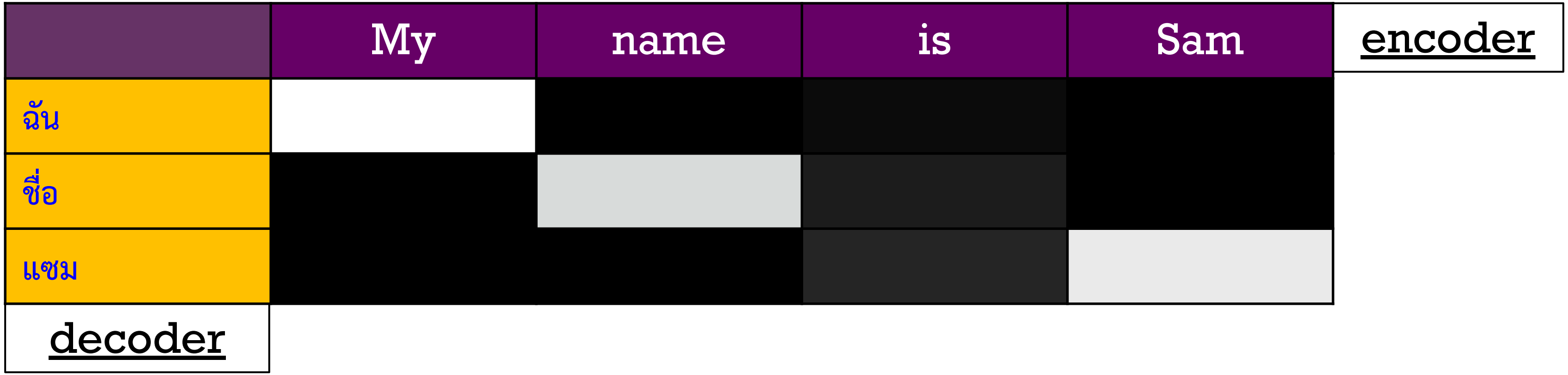
Attention mechanism = Instead of encoding all the information into a fixed-length vector, the decoder gets to decide parts of the input source to pay attention.

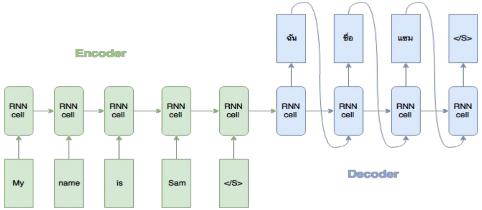


Machine Translation Problem: English to Thai

Graphical Example: English-to-Thai machine translation

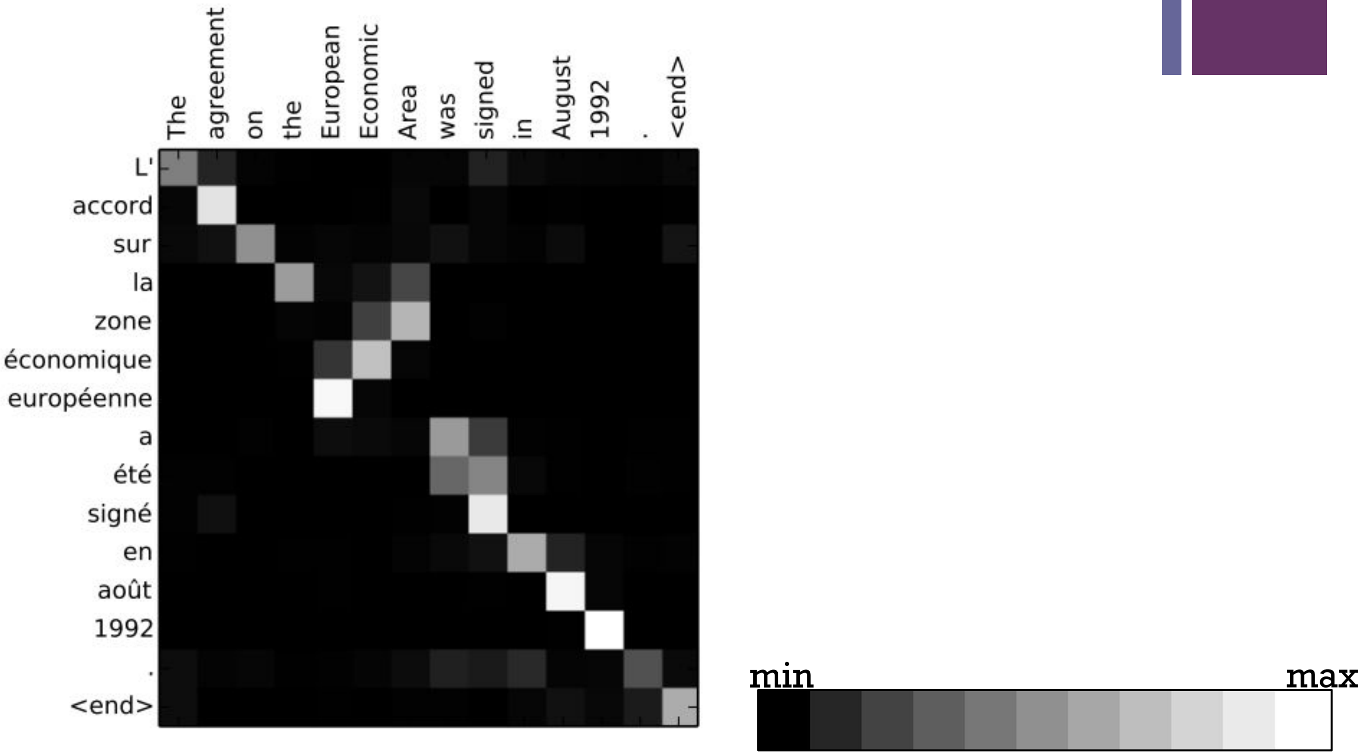
- This is a rough estimate of what might occur for English-to-Thai translation





Machine Translation Problem: English to Thai

Graphical Example: English-to-French machine translation



Reference: Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." ICLR(2015).

Attention Mechanism: Recap Basic Idea

- Encode each word in the sequence into a vector
- When DECODING, perform a linear combination of these encoded vectors from the encoding step with their corresponding "attention weights".
  - (scalar 1)(encoded vector1) + (scalar 2)(encoded vector 2) + (scalar 3)(encoded vector 3)

$$\mathbf{c}_i = \sum_j a_{ij} \mathbf{h}_j$$

- A vector formed by this linear combination is called "context vector"
- Use context vectors as inputs for the decoding step

Reference: Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." ICLR(2015).

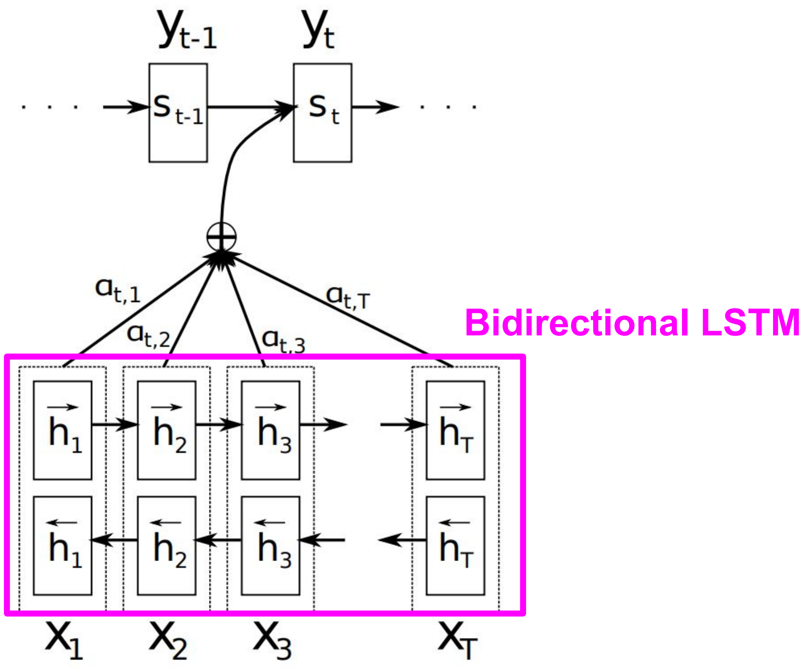


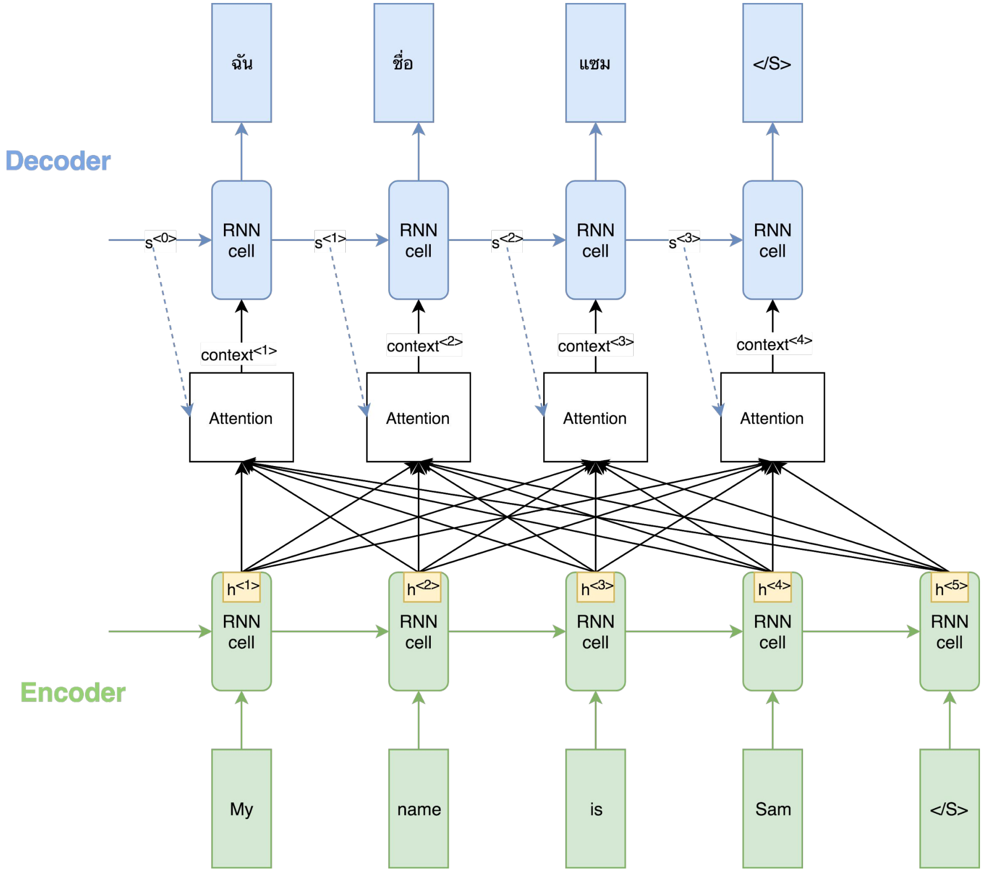
Figure 1: The graphical illustration of the proposed model trying to generate the t-th target word y_t given a source sentence $(x_1, x_2, \ldots, x_T)$.

source = encoder

target word = decoder

Reference: Bahdanau, D., Cho, K., & Bengio, Y.. Neural Machine Translation by Jointly Learning to Align and Translate. ICLR 2015

RNN and attention mechanism



Attention Mechanism (1): C_i

We want to calculate a context vector ***c*** based on hidden states $\mathbf{s}_0 \ldots \mathbf{s}_{m-1}$ that can be used with the current state $\mathbf{h}_j$ for prediction. The context vector $\mathbf{c}_i$ at position "i" is calculated as an average of the previous states weighted with the attention scores $\mathbf{a}_i$.

$$\boxed{\mathbf{c}_i} = \sum_j \boxed{a_{ij}}\boxed{\mathbf{h}_j}$$

context vector

attention score

encoder state at index j

i = decoder index
j = encoder index

$$\boxed{a_{ij}} = softmax(f_{att}(\boxed{\mathbf{s}_{i-1}}, \boxed{\mathbf{h}_j}))$$

previous hidden state/
decoder state

attention score(weight vector)
of encoder state at index j

encoder state at index j

Reference: http://ruder.io/deep-learning-nlp-best-practices/index.html#attention

Attention Mechanism (2): f_{att}

The attention function $f_{att}(\mathbf{s}_{i-1}, \mathbf{h}_j)$ calculates an unnormalized alignment score between the current hidden state $\mathbf{s}_{i-1}$ and the previous hidden state $\mathbf{h}_j$.

There are many variants of the attention function f_{att}.

$$\boxed{\mathbf{c}_i} = \sum_j \boxed{a_{ij}}\boxed{\mathbf{h}_j}$$

context vector

attention score

encoder state at index j

i = decoder index
j = encoder index

$$\boxed{a_{ij}} = softmax(f_{att}(\boxed{\mathbf{s}_{i-1}}, \boxed{\mathbf{h}_j}))$$

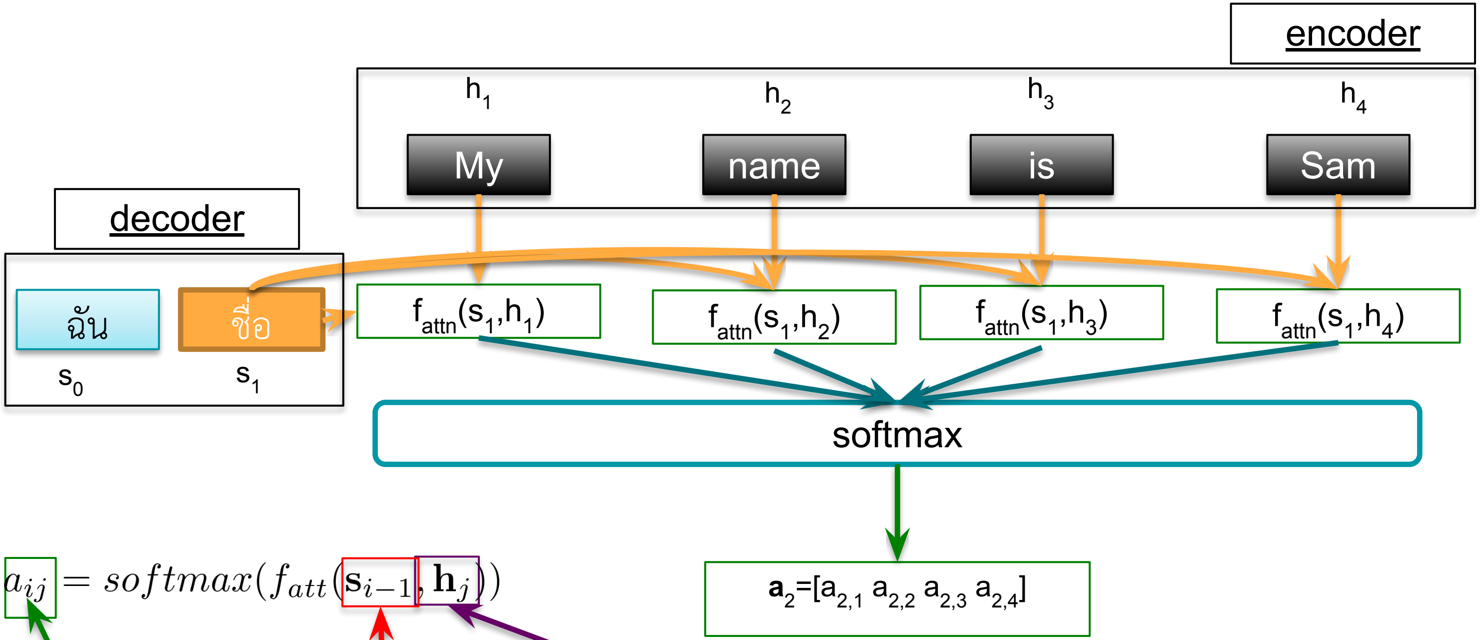
previous hidden state/
decoder state

attention score(weight vector)
of encoder state at index j

encoder state at index j

Reference: http://ruder.io/deep-learning-nlp-best-practices/index.html#attention

Attention Calculation Example (1): Attention Scores



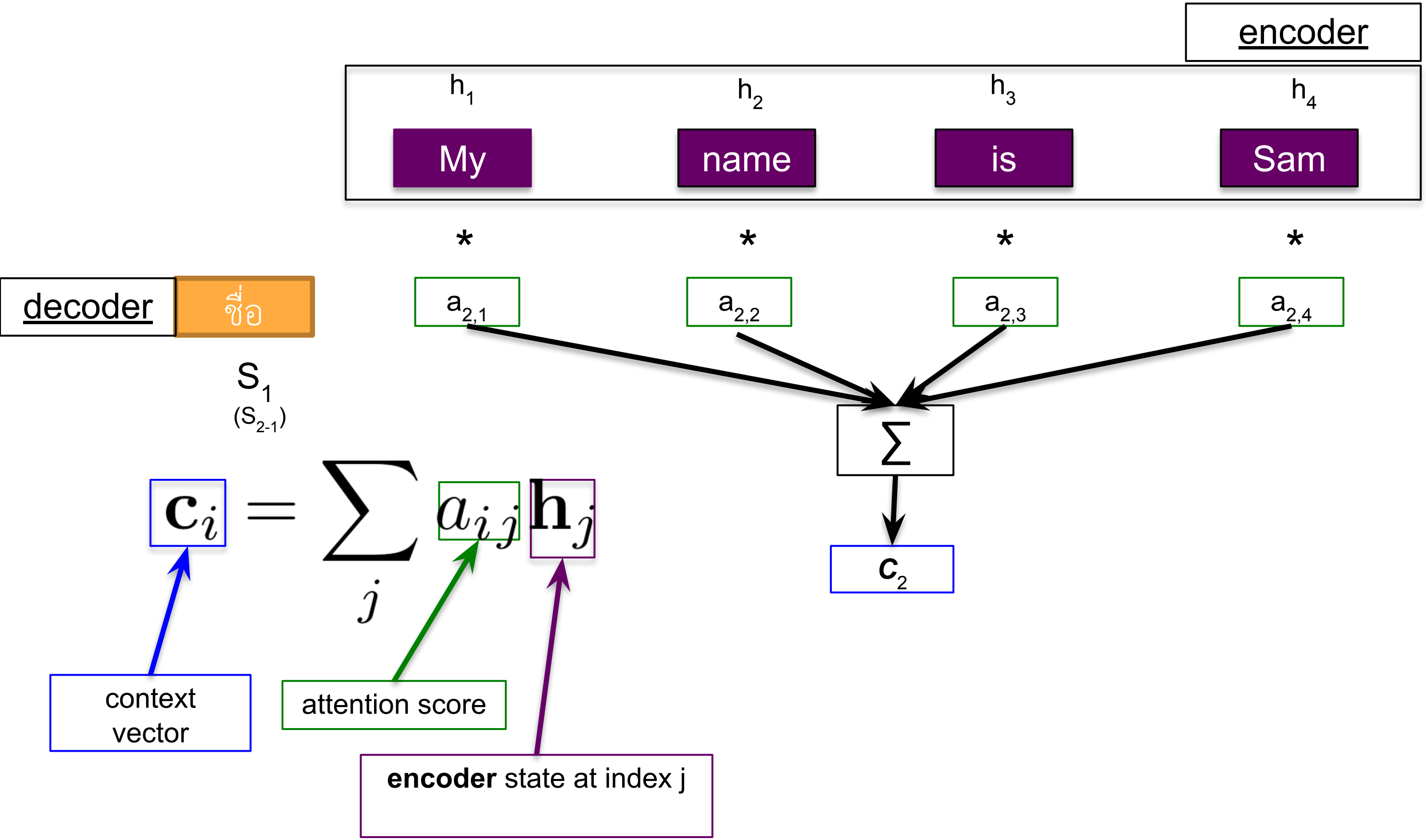
$$\boxed{a_{ij}} = softmax(f_{att}(\boxed{\mathbf{s}_{i-1}}, \boxed{\mathbf{h}_j}))$$

attention scores

previous hidden state/ decoder state

encoder state at index j

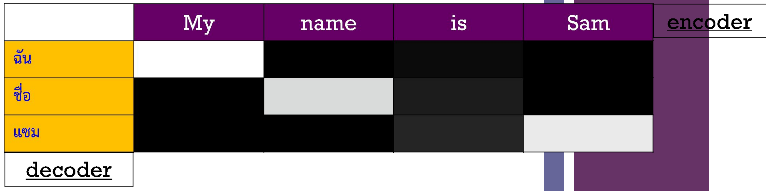
Attention Calculation Example (2): Context Vector



$$\mathbf{c}_i = \sum_j a_{ij} \mathbf{h}_j$$

$$\boxed{a_{ij} = softmax(f_{att}(\mathbf{s}_{i-1}, \mathbf{h}_j))}$$

Type of Attention mechanisms



(Remember that there are many variants of attention function f_{attn})

Additive attention: The original attention mechanism (Bahdanau et al., 2015) uses a one-hidden layer feed-forward network to calculate the attention alignment:

$$f_{attn}(\mathbf{s}_{i-1}, \mathbf{h}_j) = tanh(\mathbf{W}_a[\mathbf{s}_{i-1}; \mathbf{h}_j])$$

Multiplicative attention: Multiplicative attention (Luong et al., 2015) simplifies the attention operation by calculating the following function:

$$f_{attn}(\mathbf{s}_{i-1}, \mathbf{h}_j) = \mathbf{s}_{i-1}^\top \mathbf{W}_a \mathbf{h}_j$$

Self-attention: Without any additional information, however, we can still extract relevant aspects from the sentence by allowing it to attend to itself using self-attention (Lin et al., 2017)

$$\mathbf{a} = softmax\boxed{(\mathbf{w}_{s_2} tanh(\mathbf{W}_{s_1}\mathbf{H}^T))}$$

Key-value attention: key-value attention (Daniluk et al., 2017) is a recent attention variant that separates form from function by keeping separate vectors for the attention calculation.

Reference: http://ruder.io/deep-learning-nlp-best-practices/index.html#attention

$$\boxed{a_{ij} = softmax(f_{att}(\mathbf{s}_{i-1}, \mathbf{h}_j))}$$

+ Additive Attention

	My	name	is	Sam	encoder
ฉัน					
ชื่อ					
แซม					
decoder					

- The original attention mechanism (Bahdanau et al., 2015) uses a one-hidden layer feed-forward network to calculate the attention alignment:

concatenation

$$f_{attn}(\mathbf{s}_{i-1}, \mathbf{h}_j) = tanh(\mathbf{W}_a[\mathbf{s}_{i-1}; \mathbf{h}_j])$$

One-hidden layer (Dense)

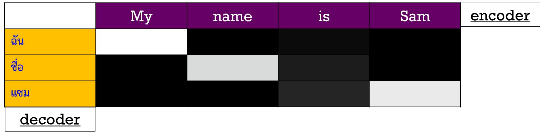
- Where $\mathbf{W}_a$ are learned attention parameters. Analogously, we can also use matrices $\mathbf{W}_1$ and $\mathbf{W}_2$ to learn separate transformations for $\mathbf{s}_{i-1}$ and $\mathbf{h}_j$ respectively, which are then summed (hence the name additive):

$$f_{attn}(\mathbf{s}_{i-1}, \mathbf{h}_j) = tanh(\mathbf{W}_1\mathbf{s}_{i-1} + \mathbf{W}_2\mathbf{h}_j)$$

Reference: http://ruder.io/deep-learning-nlp-best-practices/index.html#attention

$$a_{ij} = softmax(f_{att}(\mathbf{s}_{i-1}, \mathbf{h}_j))$$

+ Multiplicative Attention



- Multiplicative attention (Luong et al., 2015) [16] simplifies the attention operation by calculating the following function:

$$f_{attn}(\mathbf{s}_{i-1}, \mathbf{h}_j) = \mathbf{s}_{i-1}^{\top} \mathbf{W}_a \mathbf{h}_j$$

- Faster, more efficient than additive attention BUT additive attention performs better for larger dimensions

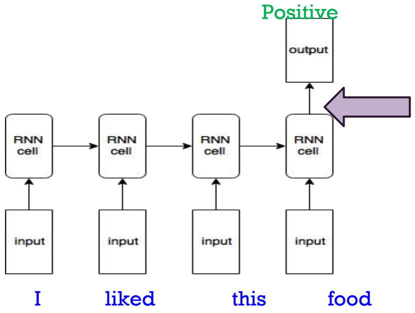
- One way to mitigate this is to scale f_{attn} by

$$\frac{1}{\sqrt{d_s}}$$

d_s = #dimensions of hidden states in LSTM (context vector; latent factors)

$$\boxed{a_{ij} = softmax(f_{att}(\mathbf{s}_{i-1}, \mathbf{h}_j))}$$

+ Self Attention (1)



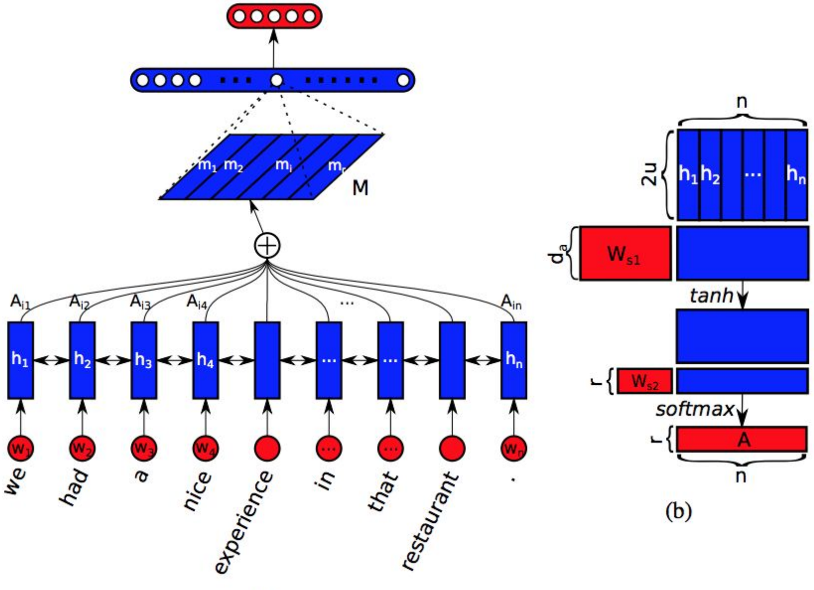
- Without any additional information, we can still extract relevant aspects from the sentence by allowing it to attend to itself using self-attention (Lin et al., 2017)

$$H = (\mathbf{h_1}, \mathbf{h_2}, \cdots \mathbf{h_n})$$

$$\mathbf{a} = \overbrace{softmax(\mathbf{w}_{s_2}}^{\text{Fully connected layer}} {\underbrace{tanh(\mathbf{W}}_{\text{One-hidden layer (Dense)}}}_{s_1} \mathbf{H}^T)$$

- w_{s1} is a weight matrix, w_{s2} is a vector of parameters. Note that these parameters are tuned by the neural networks.

- **The objective is to improve a quality of embedding vector by adding context information.**



(a)

(b)

+ Self-attention (2)

- if I can give this restaurant a 0 I will we be just ask our waitress leave because someone with a reservation be wait for our table my father and father-in-law be still finish up their coffee and we have not yet finish our dessert I have never be so humiliated do not go to this restaurant their food be mediocre at best if you want excellent Italian in a small intimate restaurant go to dish on the South Side I will not be go back
- this place suck the food be gross and taste like grease I will never go here again ever sure the entrance look cool and the waiter can be very nice but the food simply be gross taste like cheap 99cent food do not go here the food shot out of me quick then it go in
- everything be pre cook and dry its crazy most Filipino people be used to very cheap ingredient and they do not know quality the food be disgusting I have eat at least 20 different Filipino family home this not even mediocre
- seriously f *** this place disgust food and shitty service ambience be great if you like dine in a hot cellar engulf in stagnate air truly it be over rate over price and they just under deliver forget try order a drink here it will take forever get and when it finally do arrive you will be ready pass out from heat exhaustion and lack of oxygen how be that a head change you do not even have pay for it I will not disgust you with the detailed review of everything I have try here but make it simple it all suck and after you get the bill you will be walk out with a sore ass save your money and spare your self the disappointment
- i be so angry about my horrible experience at Medusa today my previous visit be amaze 5/5 however my go to out of town and I land an appointment with Stephanie I go in with a picture of roughly what I want and come out look absolutely nothing like it my hair be a horrible ashy blonde not anywhere close to the platinum blonde I request she will not do any of the pop of colour I want and even after specifically tell her I do not like blunt cut my hair have lot of straight edge she do not listen to a single thing I want and when I tell her I be unhappy with the colour she basically tell me I be wrong and I have do it this way no no I do not if I can go from Little Mermaid red to golden blonde in 1 sitting that leave my hair fine I shall be able go from golden blonde to a shade of platinum blonde in 1 sitting thanks for ruin my New Year's with 1 the bad hair job I have ever have

(a) 1 star reviews

- I really enjoy Ashley and Ami salon she do a great job be friendly and professional I usually get my hair do when I go to MI because of the quality of the highlight and the price the price be very affordable the highlight fantastic thank Ashley i highly recommend you and ill be back
- love this place it really be my favorite restaurant in Charlotte they use charcoal for their grill and you can taste it steak with chimichurri be always perfect Fried yucca cilantro rice pork sandwich and the good tres lech I have had.The desert be all incredible if you do not like it you be a mutant if you will like diabeetus try the Inca Cola
- this place be so much fun I have never go at night because it seem a little too busy for my taste but that just prove how great this restaurant be they have amazing food and the staff definitely remember us every time we be in town I love when a waitress or waiter come over and ask if you want the cab or the Pinot even when there be a rush and the staff be run around like crazy whenever I grab someone they instantly smile acknowlegde us the food be also killer I love when everyone know the special and can tell you they have try them all and what they pair well with this be a first last stop whenever we be in Charlotte and I highly recommend them
- great food and good service .... what else can you ask for everything that I have ever try here have be great
- first off I hardly remember waiter name because its rare you have an unforgettable experience the day I go I be celebrate my birthday and let me say I leave feel extra special our waiter be the best ever Carlos and the staff as well I be with a party of 4 and we order the potato salad shrimp cocktail lobster amongst other thing and boy be the food great the lobster be the good lobster I have ever eat if you eat a dessert I will recommend the cheese cake that be also the good I have ever have it be expensive but so worth every penny I will definitely be back there go again for the second time in a week and it be even good ...... this place be amazing

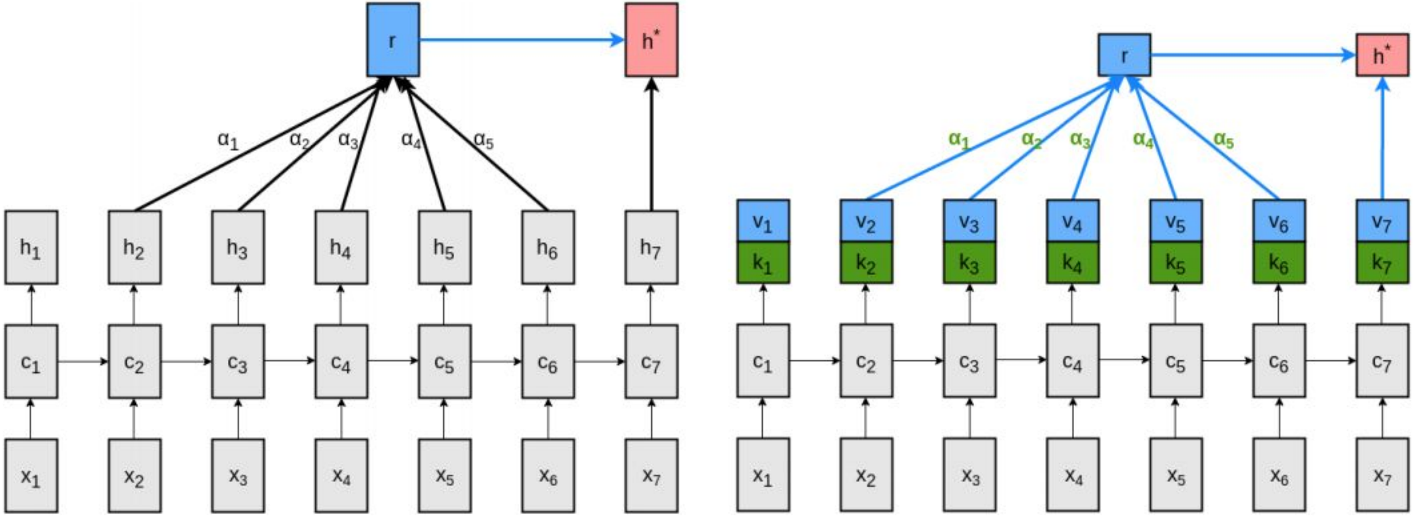
(b) 5 star reviews

Figure 2: Heatmap of Yelp reviews with the two extreme score.

$$a_{ij} = softmax(f_{att}(\mathbf{s}_{i-1}, \mathbf{h}_j))$$

Key-value attention (1)

$$\mathbf{c}_i = \sum_j a_{ij}\mathbf{h}_j$$



(a) Neural language model with attention.

(b) Key-value separation.

Value=encoded vector

Key=used for attention score calculation

Reference: Daniluk, M., Rockt, T., Welbl, J., & Riedel, S. (2017). Frustratingly Short Attention Spans in Neural Language Modeling. In ICLR 2017.

Key-value attention (2)

key

value

$$\begin{bmatrix} \boldsymbol{k}_t \\ \boldsymbol{v}_t \end{bmatrix} = \boldsymbol{h}_t$$

previous outputs (memory) **with window L**

current output

$$\boldsymbol{M}_t = \tanh(\boldsymbol{W}^Y [\boldsymbol{k}_{t-L} \cdots \boldsymbol{k}_{t-1}] + (\boldsymbol{W}^h \boldsymbol{k}_t)\boldsymbol{1}^T)$$

Attention score

$$\boldsymbol{\alpha}_t = \mathrm{softmax}(\boldsymbol{w}^T \boldsymbol{M}_t)$$

Context vector (v_t not included)

$$\boldsymbol{r}_t = [\boldsymbol{v}_{t-L} \cdots \boldsymbol{v}_{t-1}]\boldsymbol{\alpha}^T$$

The final representation

$$\boldsymbol{h}_t^* = \tanh(\boldsymbol{W}^r \boldsymbol{r}_t + \boldsymbol{W}^x \boldsymbol{v}_t)$$

Context vector (vt not included)

Original vector at index t

Demo: Neural Machine Translation with attention (Additive Attention)

- Translate: one date format to another
  - 27 January 2018 2018-01-27
  - 27 JAN 2018 2018-01-27

+

Question Answering and Deep Learning

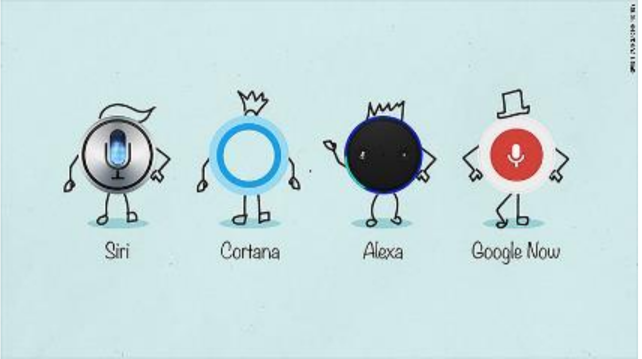
Introduction

Traditional QA

Memory Network

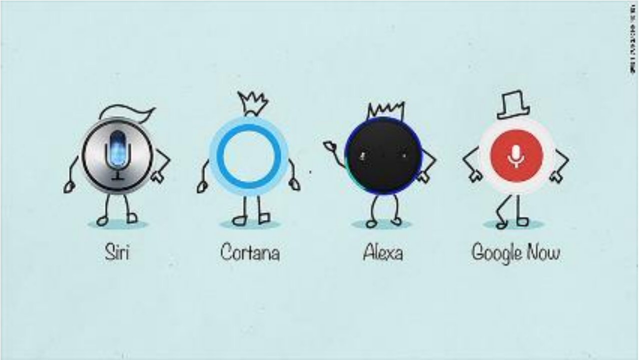
- End-to-End Memory Network
- Key-Valued Memory Network

Introduction



What's Question Answering (QA)?

- QA is a field that combines (1) Information Retrieval, (2) Information Extraction and (3) Natural Language Processing.
  - *We will focus on the NLP part*
- Most notable QA software is IBM's Watson
- Nowadays, QA also play a significant role in Personal Assistant (Siri, Cortana, etc.)



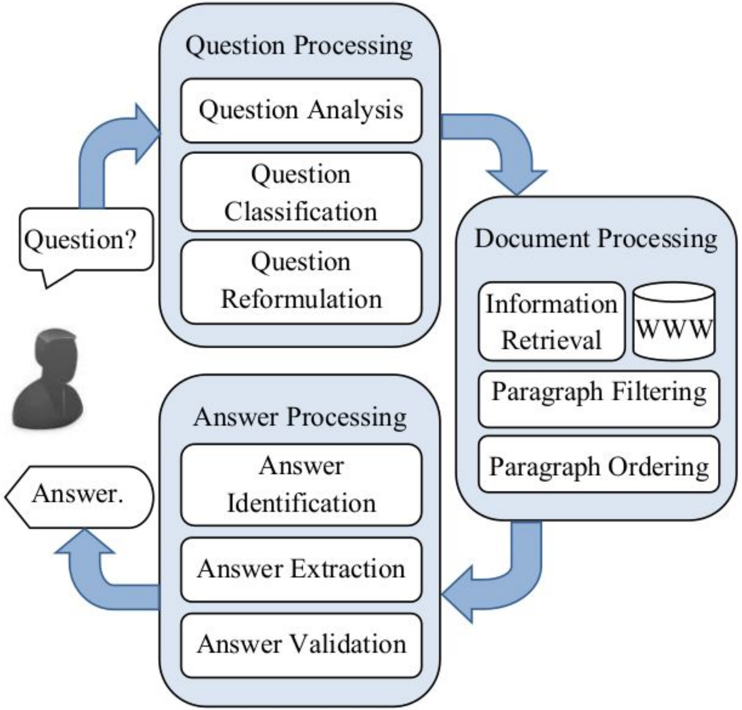
[Figure by Sandy Jakobs (left), IBM (right)

Type Of QA

- By application domains
  - Restricted Domain
  - Open Domain
- By source of data
  - Structured data (Knowledge-based) - e.g. Freebase, Google Knowledge Graph
  - Unstructured data (Document)- Web, Wiki
- By answer
  - Factoid (single word - when, what, where)
  - non-Factoid (e.g., list, how, why)
- The forms of answer
  - Extracted text
  - Generated answer

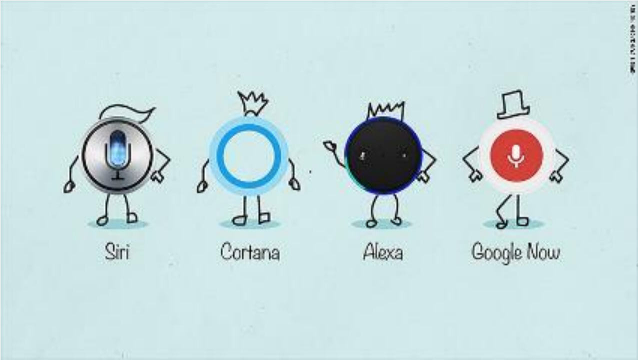
Process Of QA

- Question Processing
  - What type of question?
  - Question preprocessing
- Document Processing
  - Rank candidate document
  - Rank candidate paragraph
- **Answer Processing**
  - Extract candidate answer from paragraph
  - Construct an answer



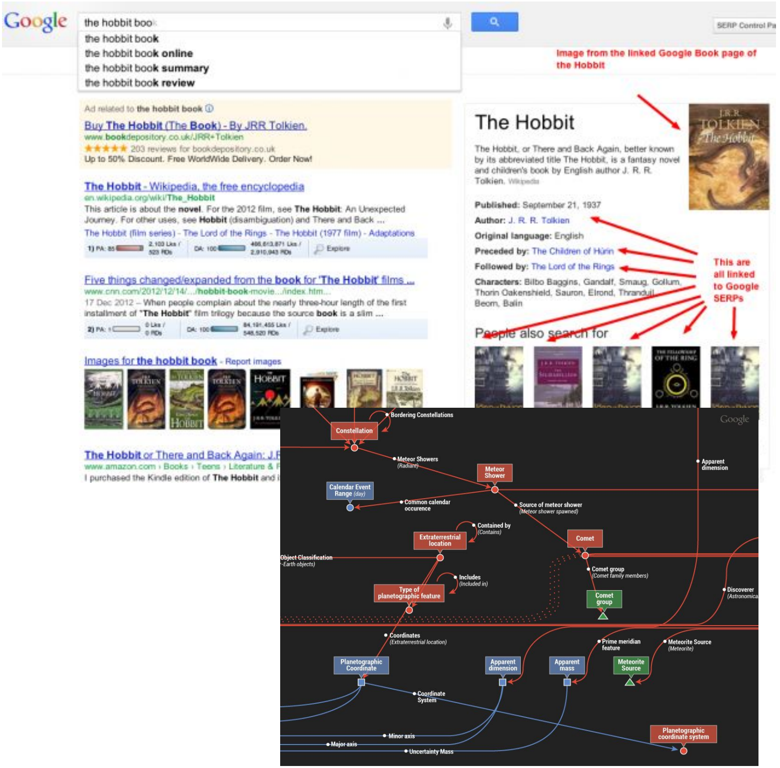
[Figure from "The Question Answering Systems: A Survey"
]

Example of QA system

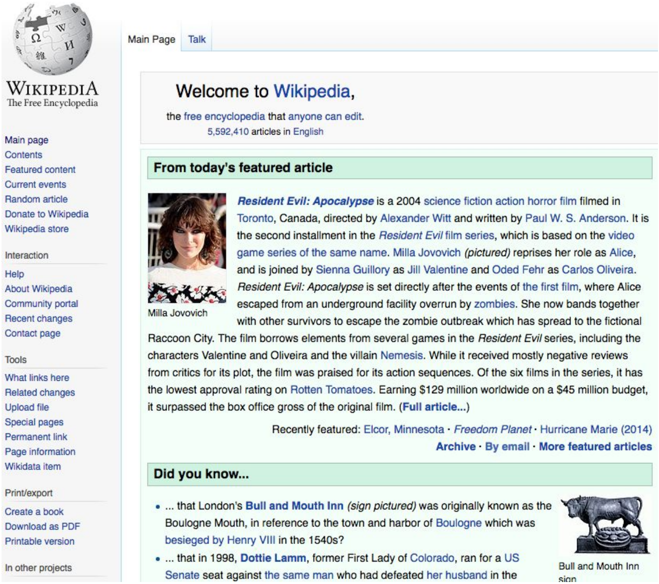


Types of QA systems

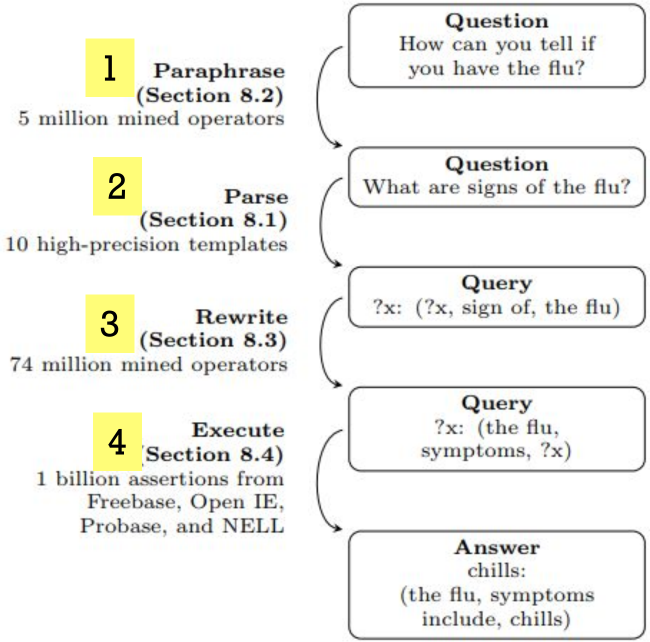
Structured Knowledge Base



Unstructured Knowledge Base



- Open Question Answering Over Curated and Extracted Knowledge Bases (A.Fader SIGKDD 2014)



[Figure from "Open Question Answering Over Curated and Extracted Knowledge Bases"]

Example of Traditional Methods (cont.)

- Open Question Answering Over Curated and Extracted Knowledge Bases (A.Fader SIGKDD 2014)
  - 1) Paraphrase operator
    - are responsible for rewording the input question into the domain of a parsing operator
    - Source template (open domain) → Target template (predefined format)

Source Template	Target Template
How does _ affect your body?	What body system does _ affect?
What is the latin name for _?	What is _'s scientific name?
Why do we use _?	What did _ replace?
What to use instead of _?	What is a substitute for _?
Was _ ever married?	Who has _ been married to?

Table 3: Example paraphrase operators that extracted from a corpus of unlabeled questions.

[Figure from "Open Question Answering Over Curated and Extracted Knowledge Bases"]

+ Example of Traditional Methods (cont.)

- Open Question Answering Over Curated and Extracted Knowledge Bases (A.Fader SIGKDD 2014)
  - 2) Parsing operator
    - responsible for interfacing between natural language questions and the KB query language
    - Target template (predefined format) → Query

Question Pattern	Query Pattern	Example Question	Example Query
Who/What RV_{rel} NP_{arg}	(?x, rel, arg)	Who invented papyrus?	(?x, invented, papyrus)
Who/What Aux NP_{arg} RV_{rel}	(arg, rel, ?x)	What did Newton discover?	(Newton, discover, ?x)
Where/When Aux NP_{arg} RV_{rel}	(arg, rel in, ?x)	Where was Edison born?	(Edison, born in, ?x)
Where/When is NP_{arg}	(arg, is in, ?x)	Where is Detroit?	(Detroit, is in, ?x)
Who/What is NP_{arg}	(arg, is-a, ?x)	What is potassium?	(potassium, is-a, ?x)
What/Which NP_{rel2} Aux NP_{arg} RV_{rel1}	(arg, rel1 rel2, ?x)	What sport does Sosa play?	(Sosa, play sport, ?x)
What/Which NP_{rel} is NP_{arg}	(arg, rel, ?x)	What ethnicity is Dracula?	(Dracula, ethnicity, ?x)
What/Who is NP_{arg}'s NP_{rel}	(arg, rel, ?x)	What is Russia's capital?	(Russia, capital, ?x)
What/Which NP_{type} Aux NP_{arg} RV_{rel}	(?x, is-a, type) (arg, rel, ?x)	What fish do sharks eat?	(?x, is-a, fish) (sharks, eat, ?x)
What/Which NP_{type} RV_{rel} NP_{arg}	(?x, is-a, type) (?x, rel, arg)	What states make oil?	(?x, is-a, states) (?x, make, oil)

[Figure from "Open Question Answering Over Curated and Extracted Knowledge Bases"]

Example of Traditional Methods (cont.)

- Open Question Answering Over Curated and Extracted Knowledge Bases (A.Fader SIGKDD 2014)
  - 3) Query-rewrite operators
    - responsible for interfacing between the vocabulary used in the input question and the internal vocabulary used by the KBs
    - Source Query → Target Query (only vocab in knowledge base)

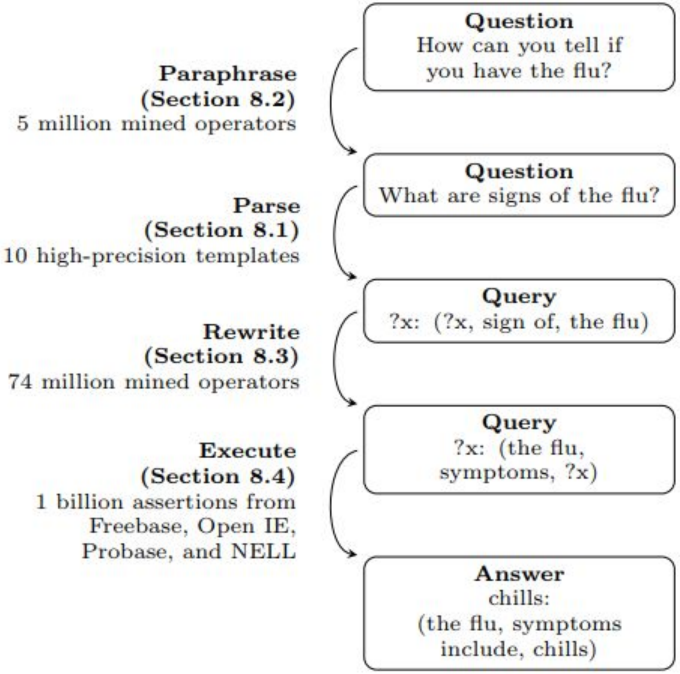
Source Query	Target Query
(?x, children, ?y)	(?y, was born to, ?x)
(?x, birthdate, ?y)	(?x, date of birth, ?y)
(?x, is headquartered in, ?y)	(?x, is based in, ?y)
(?x, invented, ?y)	(?y, was invented by, ?x)
(?x, is the language of, ?y)	(?y, languages spoken, ?x)

Table 4: Example query-rewrite operators mined from the knowledge bases described in Section 4.1.

[Figure from "Open Question Answering Over Curated and Extracted Knowledge Bases"]

Example of Traditional Methods (cont.)

- Open Question Answering Over Curated and Extracted Knowledge Bases (A.Fader SIGKDD 2014)
  - 4) Execution operator
    - responsible for fetching and combining evidence from the Knowledge based, given a query

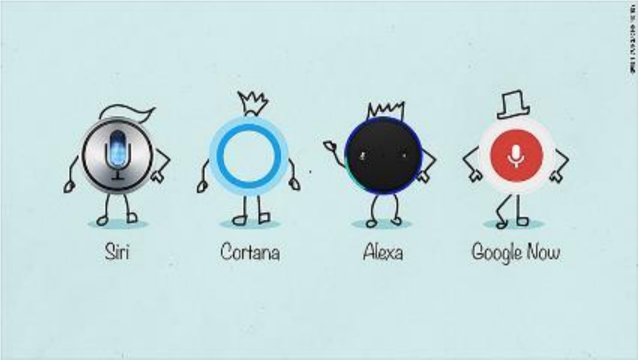


[Figure from "Open Question Answering Over Curated and Extracted Knowledge Bases"]

Limitation

- Require a lot of time and linguistic knowledge to create a template
- Require many templates for each question type (manual process)
- Can only answer simple **factoid** question

Memory Network



Deep Learning and QA: Memory Network

- **Memory Network [Jason Weston, et'al, 2015]**
  - Deep Learning with a memory component.
  - Incorporates **reasoning** over **memory**
- Why memory network and QA?
  - LONG-term memory is required to read a story to answer questions about it
    - Long-term = HDD (database)
    - Short-term = RAM (question, chat)

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	1) I'm looking a fun comedy to watch tonight, any ideas?
	c_1^r	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.

Example: Memory Network?

- Factoid QA with Two Supporting Facts **("where is actor + object")**

fact

John is in the playground.
Bob is in the office.
John picked up the football.
Bob went to the kitchen.

question

Where is the football? A:playground

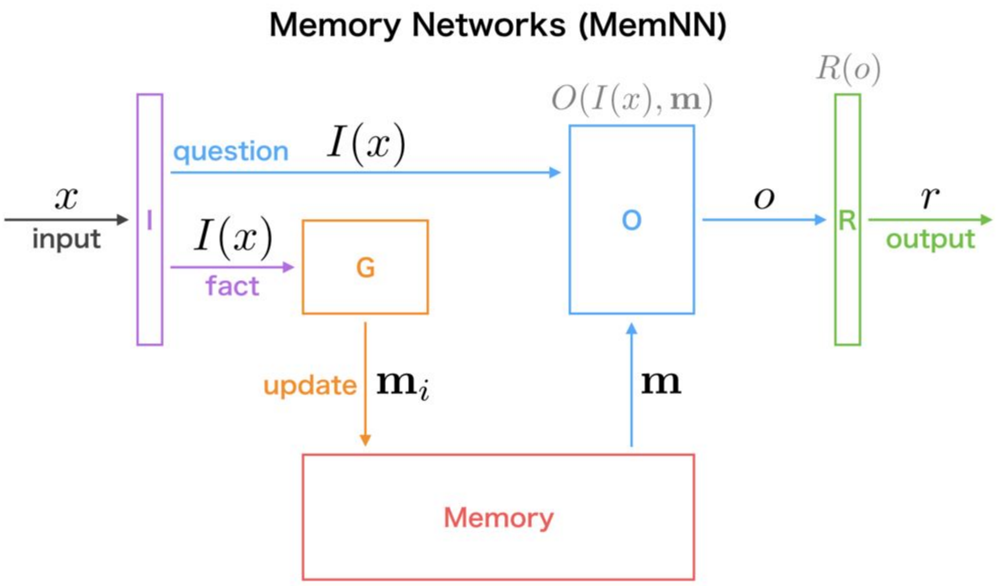
Example: Memory Network? (cont.)

- Factoid QA with Two Supporting Facts **("where is actor+object")**

John is in the playground. ← SUPPORTING FACT
Bob is in the office.
John picked up the football. ← SUPPORTING FACT
Bob went to the kitchen.
Where is the football? A:playground

+
What is Memory Network?

- MemNNs have **four component** networks (which may or may not have shared parameters):
  - I: (input feature map) convert incoming data to the internal feature representation.
    - bag of words, RNN style reading at word or character level, etc.
  - G: (generalization) update memories given new input.
  - O: produce new output (in feature representation space) given the memories.
    - multi-class classifier or uses an RNN to output sentences
  - R: (response) convert output O into a response seen by the outside world.
    - For example, factoid (softmax), text generation



Memory Network: Core Inference

- The core of inference lies in the O and R modules.
  - In case that we only have one supporting sentence, the model can find it by using input x

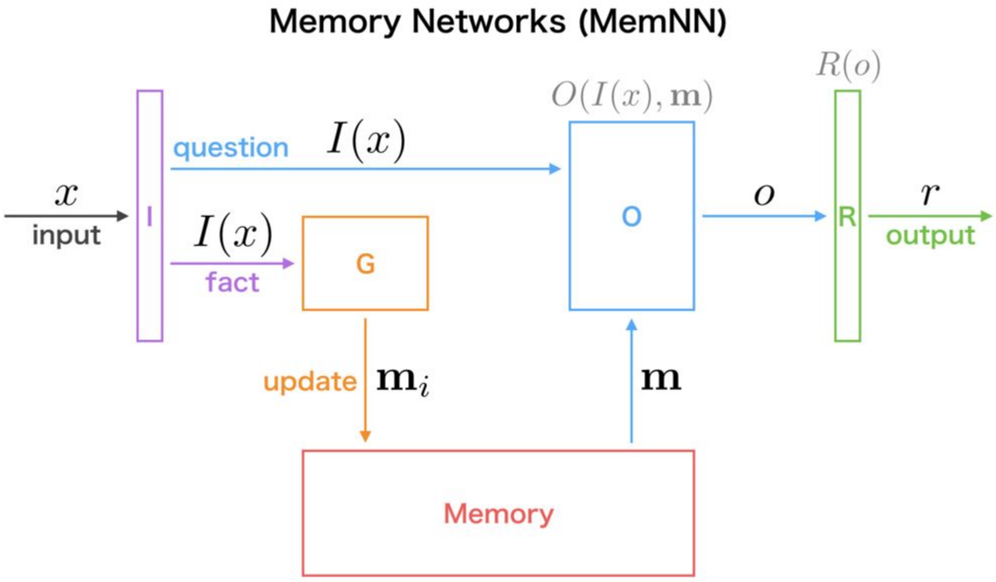
$$o_1 = O_1(x, \mathbf{m}) = \arg\max_{i=1,\ldots,N} s_O(x, \mathbf{m}_i)$$

  - If we have 2 supporting sentences, we can use input x and the first supporting sentence to find the second supporting sentence

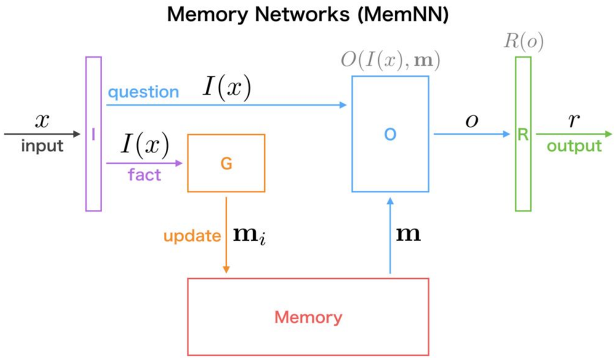
$$o_2 = O_2(x, \mathbf{m}) = \arg\max_{i=1,\ldots,N} s_O([x, \mathbf{m}_{o_1}], \mathbf{m}_i)$$

  - In general we can make any N inference step using x and all previous supporting sentences
  - After that we can produce, the response (r) by selecting the single word answer with input and all supporting sentences

$$r = \operatorname{argmax}_{w \in W} s_R([x, \mathbf{m}_{o_1}, \mathbf{m}_{o_2}], w)$$



Memory Network: Train and Loss



- Scoring function s is just a Matrix multiplication operation
  - Where x=inputs, y=target

$$s(x, y) = \Phi_x(x)^\top U^\top U \Phi_y(y).$$

- Training
  - Max margin ranking loss and stochastic gradient descent

$$\sum_{\bar{f} \neq \mathbf{m}_{o_1}} \max(0, \gamma - s_O(x, \mathbf{m}_{o_1}) + s_O(x, \bar{f})) + \tag{6}$$

$$\sum_{\bar{f}' \neq \mathbf{m}_{o_2}} \max(0, \gamma - s_O([x, \mathbf{m}_{o_1}], \mathbf{m}_{o_2}]) + s_O([x, \mathbf{m}_{o_1}], \bar{f}'])) + \tag{7}$$

$$\sum_{\bar{r} \neq r} \max(0, \gamma - s_R([x, \mathbf{m}_{o_1}, \mathbf{m}_{o_2}], r) + s_R([x, \mathbf{m}_{o_1}, \mathbf{m}_{o_2}], \bar{r}])) \tag{8}$$

where $\bar{f}$, $\bar{f}'$ and $\bar{r}$ are all other choices than the correct labels, and γ is the margin. At every step of SGD we sample $\bar{f}$, $\bar{f}'$, $\bar{r}$ rather than compute the whole sum for each training example, following e.g., Weston et al. (2011).

End-To-End Memory Network: Overview (1)

- Limitation of Memory Networks
  - Use hard attention
  - Requires explicit supervision of attention during training (must identify **all facts** for each questions)
  - Only feasible for simple tasks
- End-to-end (MemN2N) model (Sukhbaatar '15):
  - Reads from memory with soft attention (weight)
  - End-to-end training with backpropagation
  - Only need supervision on the final output

End-To-End Memory Network
Attention during **three memory hops**

- Example of model mechanism

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				

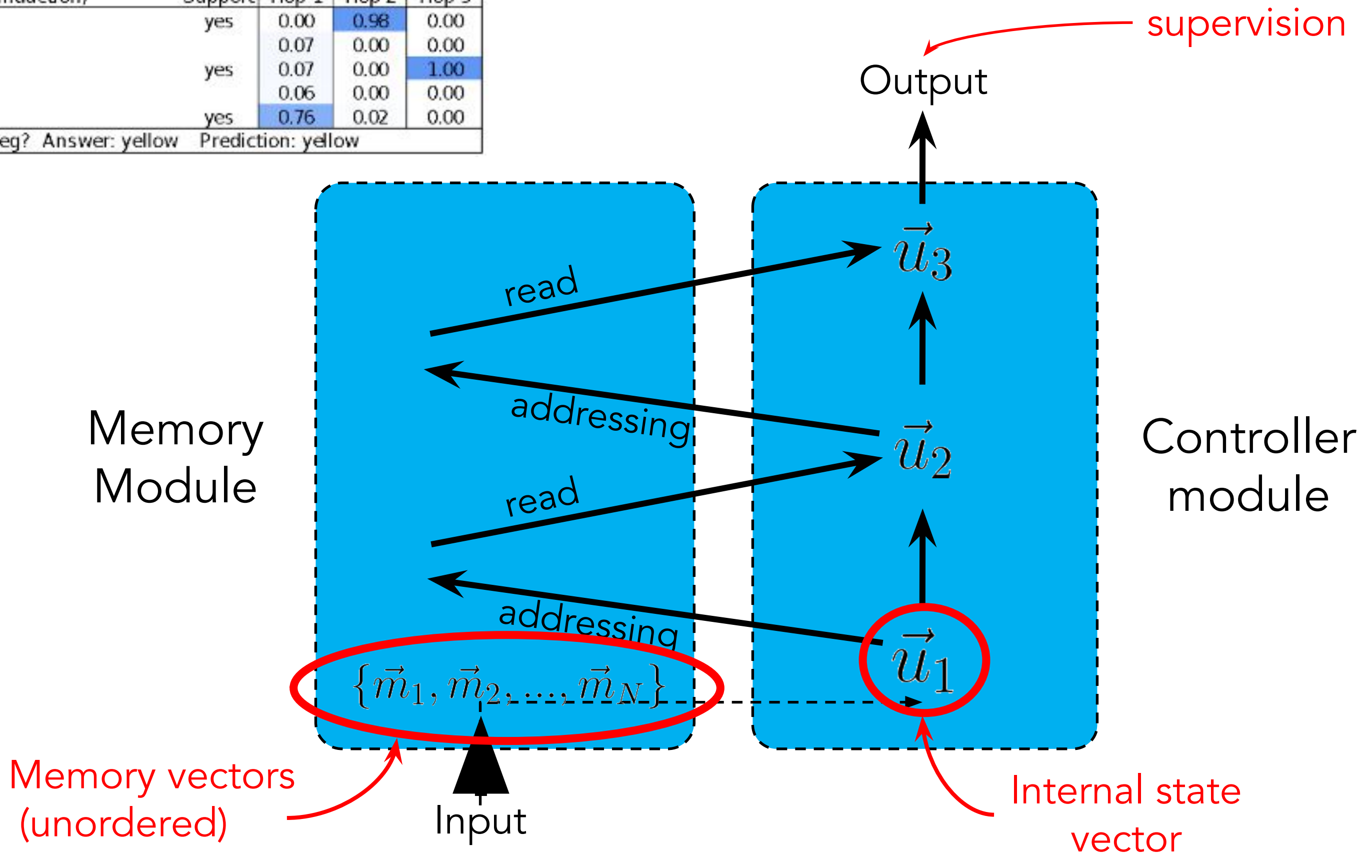
Story (16: basic induction)	Support	Hop 1	Hop 2	Hop 3
Brian is a frog.	yes	0.00	0.98	0.00
Lily is gray.		0.07	0.00	0.00
Brian is yellow.	yes	0.07	0.00	1.00
Julius is green.		0.06	0.00	0.00
Greg is a frog.	yes	0.76	0.02	0.00
What color is Greg? Answer: yellow Prediction: yellow				

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				

[Figure by Sukhbaatar Sainbayar]

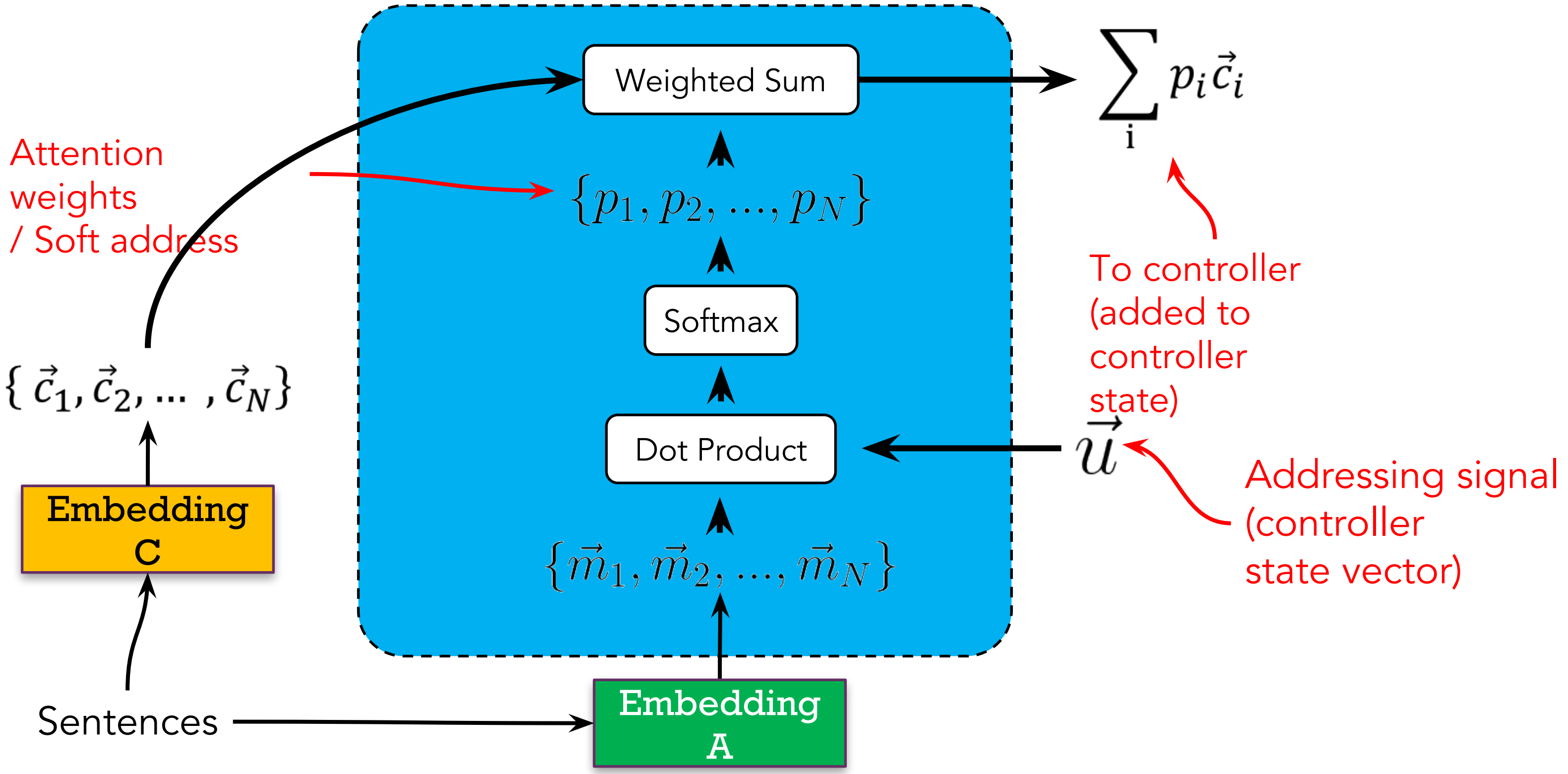
+ End-To-End Memory Network: Overview (2); 2 hops

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			

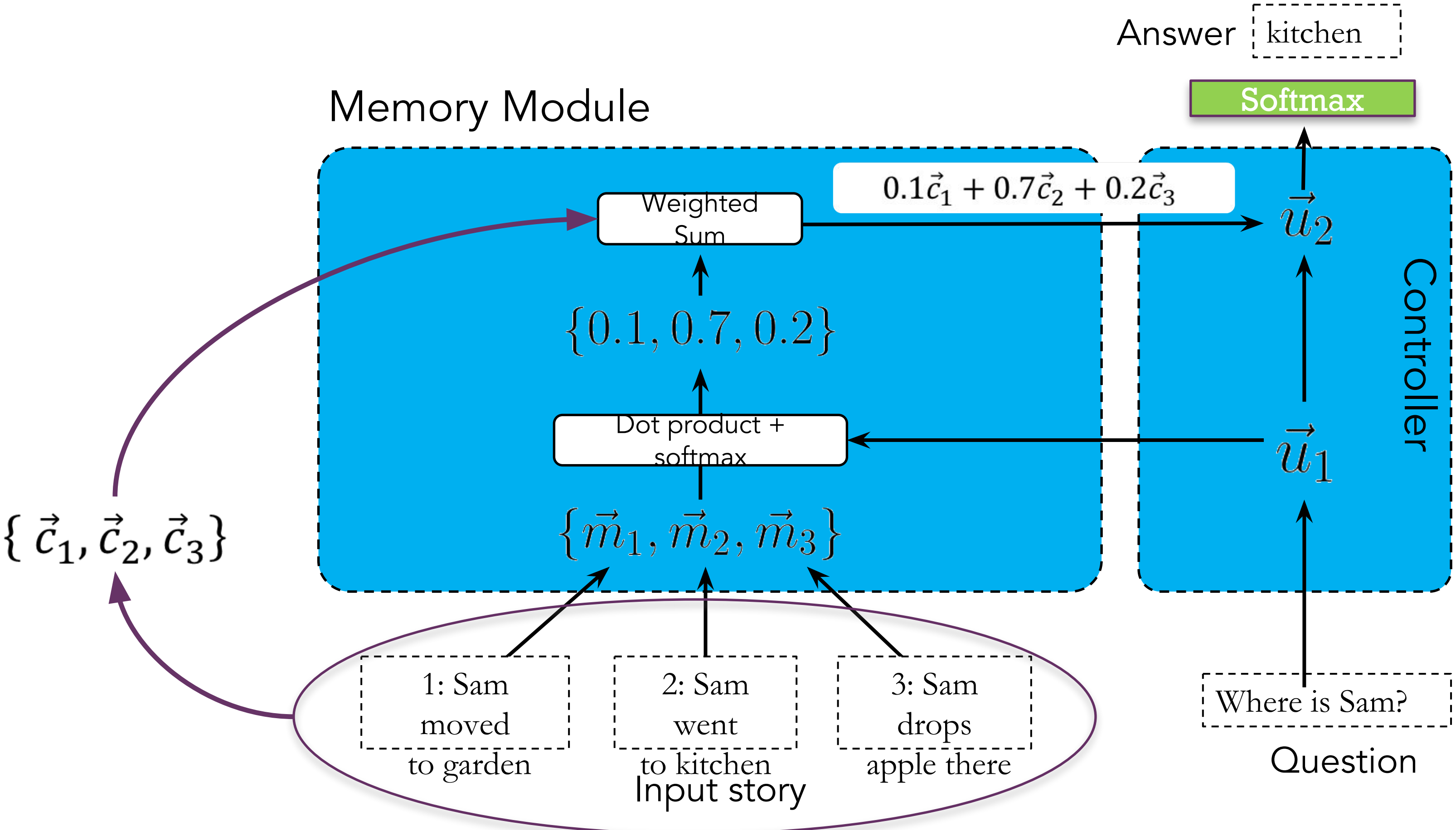


[Figure by Jason Weston]

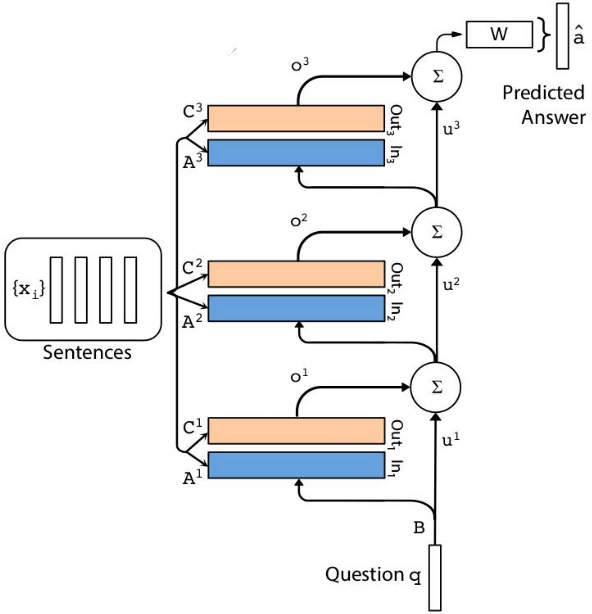
End-To-End Memory Network: Memory Module



End-To-End Memory Network: Example



End-To-End Memory Network Multiple Hops Reasoning

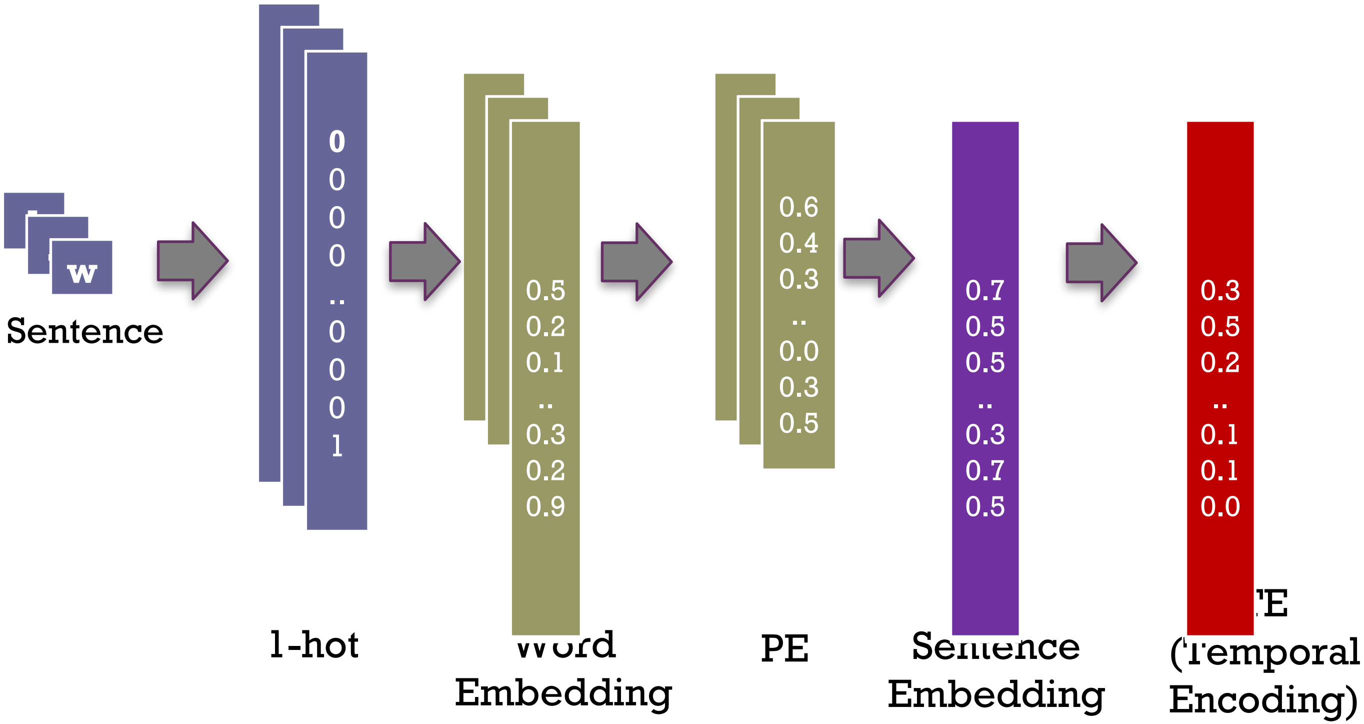


[Figure by Sukhbaatar Sainbayar]

End-To-End Memory Network
Multiple Hops Reasoning

- There are two ways to do the multiple hops reasoning
- 1) Adjacent
  - The query representation (u) is updated every hop
    - $u_{k+1} = u_k + o_k$
  - Input embedding of the new layer is output embedding of the previous layer
    - $A_{k+1} = C_k$
- 2) Layer-wise (RNN-like)
  - The query representation (u) is updated every hop with H linear mapping
    - $u_{k+1} = Hu_k + o_k$
  - Every embedding matrix is the same
    - $A_1 = A_2 ... = A_k$
    - $C_1 = C_2 .. = C_k$

End-To-End Memory Network
Memory Vector



End-To-End Memory Network
Memory Vector

- Word Embedding
  - Embed every word in a sentence
- Positional Encoding (PE)
  - Position is modeled by a multiplicative term on each word vector with weights depending on the position in the sentence.
- Sentence Embedding
  - Summation of all embedded words in the sentence
- Temporal Encoding (TE)
  - Encoded timestamp (or index) of the sentence in the story

End-To-End Memory Network
Positional Encoding

- Positional Encoding (PE)
  - Position is modeled by a multiplicative term on each word vector with weights depending on the position in the sentence

j=positional index
d=dimensional index

$$S = \underbrace{\begin{array}{|c|c|c|c|}\hline \textbf{I} & \textbf{ate} & \textbf{fried} & \textbf{rice} \\ \hline\end{array}}_{J}$$

$f(j,d) = (1 - j/J) - (d/D)(1 - 2j/J)$

PE(I) =

	I			=
1	0.13	O	0.70	0.09
2	0.04	O	0.65	0.03
…	…	O	…	…
…	…	O	…	…
D-1	0.91	O	0.30	0.27
D	0.82	O	0.25	0.20

O is element-wise multiplication

End-To-End Memory Network Sentence Embedding

- Sentence Embedding
  - Summation of all embedded words in the sentence

S =

I	ate	fried	rice
0.09	0.14	0.00	0.01
0.03	0.01	0.01	0.00
…	…	…	…
…	…	…	…
0.27	0.05	0.21	0.03
0.20	0.02	0.16	0.31

Σ

=

0.24
0.05
…
…
0.56
0.69

End-To-End Memory Network
Temporal Encoding (TE)

- Temporal Encoding (TE)
  - Encoded timestamp (or index) of the sentence in the story

Sentence		T(index)		
0.24		**0.01**		**0.25**
0.05		0.42		0.47
…	+	…	=	…
…		…		…
0.56		0.03		0.59
0.69		-0.11		0.58

End-To-End Memory Network
Attention during three memory hops

- Example of model mechanism

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				

[Figure by Sukhbaatar Sainbayar]

Memory Network Briefing

- Memory Network [Jason Weston, et'al, 2015]
  - Machine learning with a memory component.
  - The model is trained to learn how to operate effectively with the memory component.
  - Multiple inference steps
  - Need strong supervision (Limitation)
- End-To-End Memory Network [Sainbayar Sukhbaatar, et' al, 2015]
  - Use attention to let model learn to select relevant memory
  - Weight tying let model remember previous step decision

Demo: QA (AllenNLP)

https://demo.allennlp.org/reading-comprehension