Text Generation & Question Answering

2/2565: FRA501 Introduction to Natural Language Processing with Deep learning

Week 07

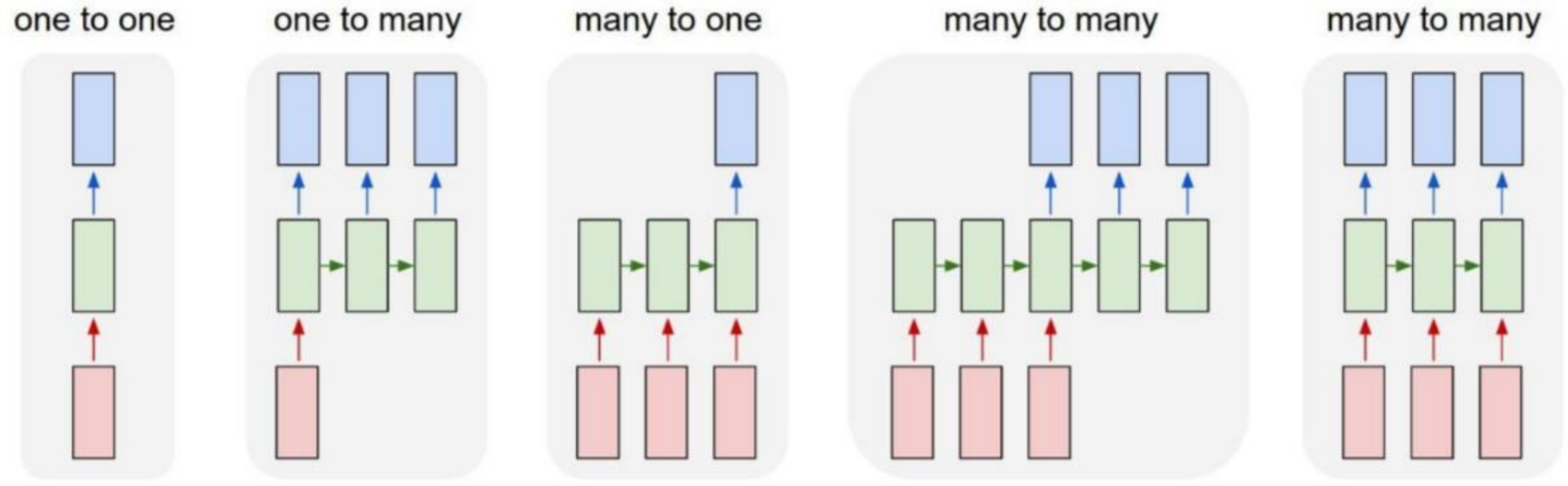
Paisit Khanarsa, Ph.D.

Institute of Field Robotics (FIBO), King Mongkut's University of Technology Thonburi

Outlines

- Text Generation
- Attention Mechanism
- Question Answering (QA) and Deep learning
    - Introduction
    - Traditional QA
    - Memory Network
        - End-to-End Memory Network
        - Key-Valued Memory Network

Different types of RNN architectures



Fixed-sized input to fixed-sized output (e.g. image classification)

Sequence output (e.g. image captioning takes an image and outputs a sentence of words).

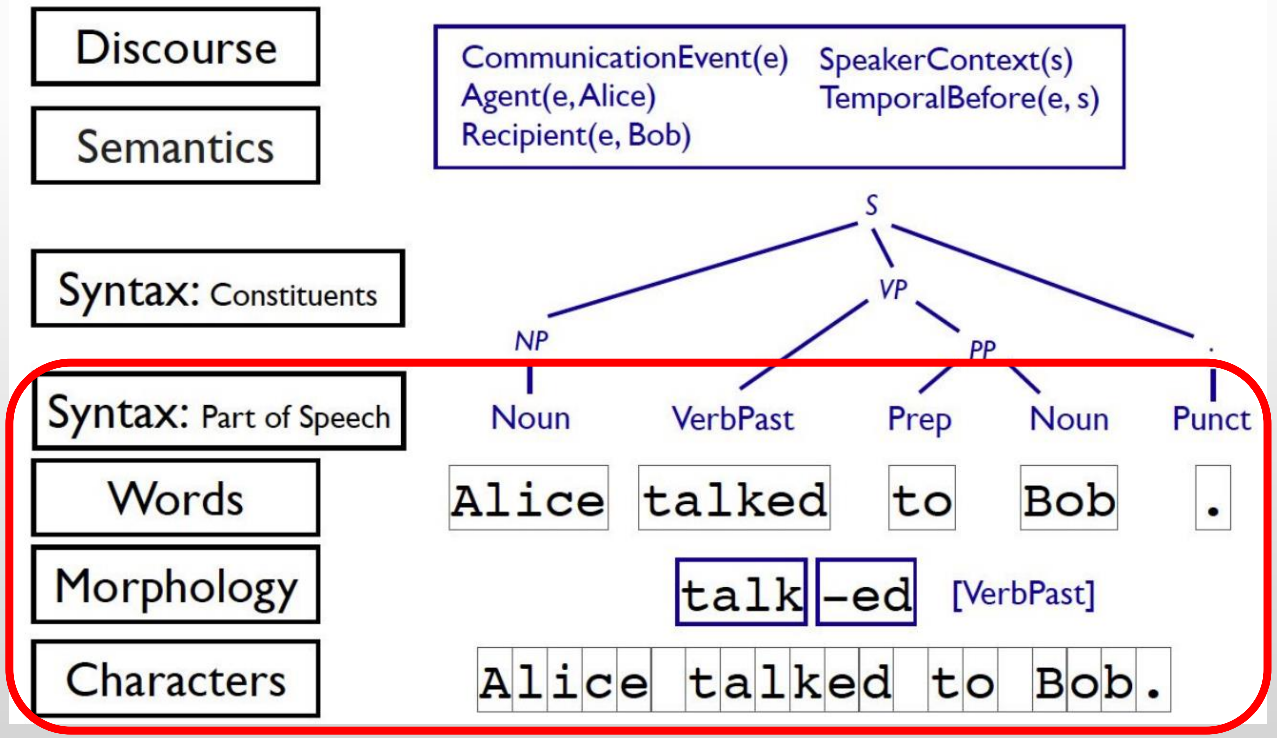
Sequence input (e.g. sentiment analysis where a given sentence is classified as expressing positive or negative sentiment).

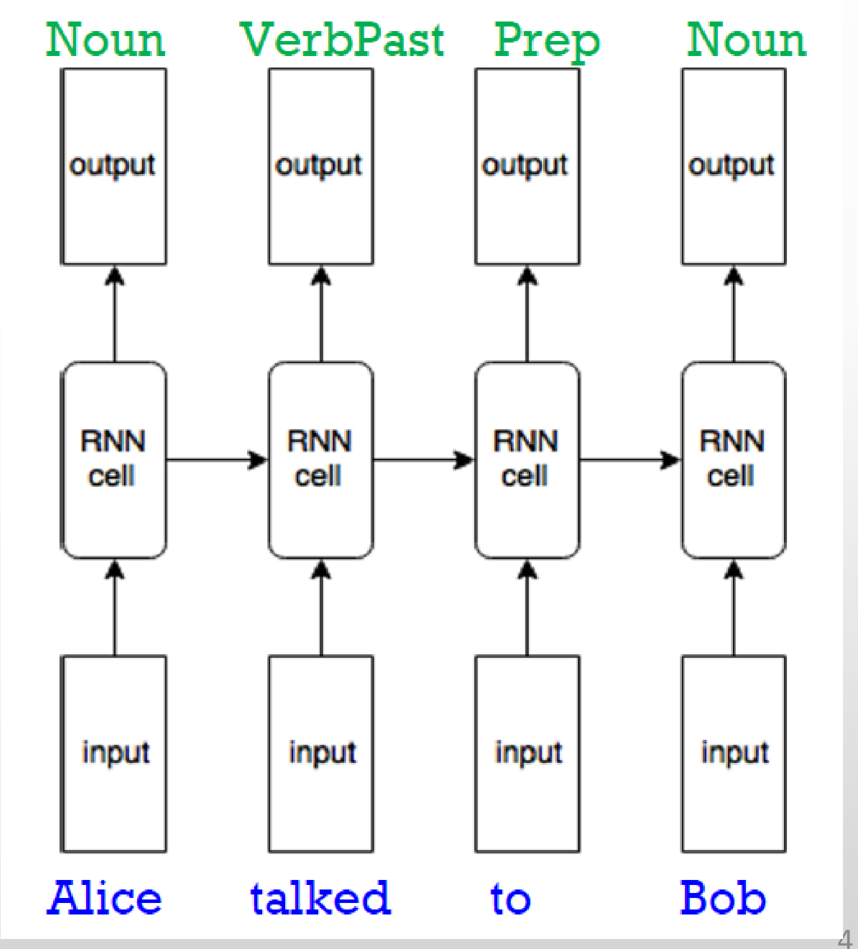
Sequence input and sequence output (e.g. Machine Translation: an RNN reads a sentence in English and then outputs a sentence in French)

Synced sequence input and output (e.g. video classification where we wish to label each frame of the video)

RNN architectures in NPL

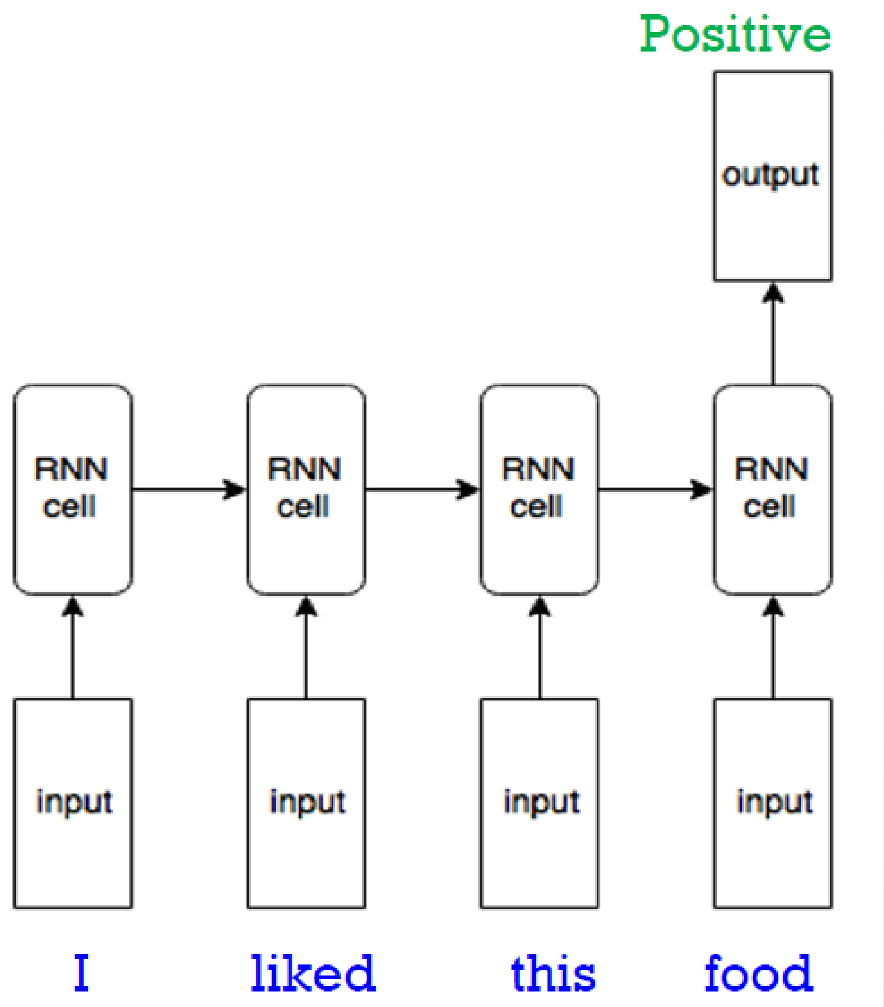
- **Many to many**
- Sequence Input, Sequence Output
- E.g. Tokenization, POS tagging





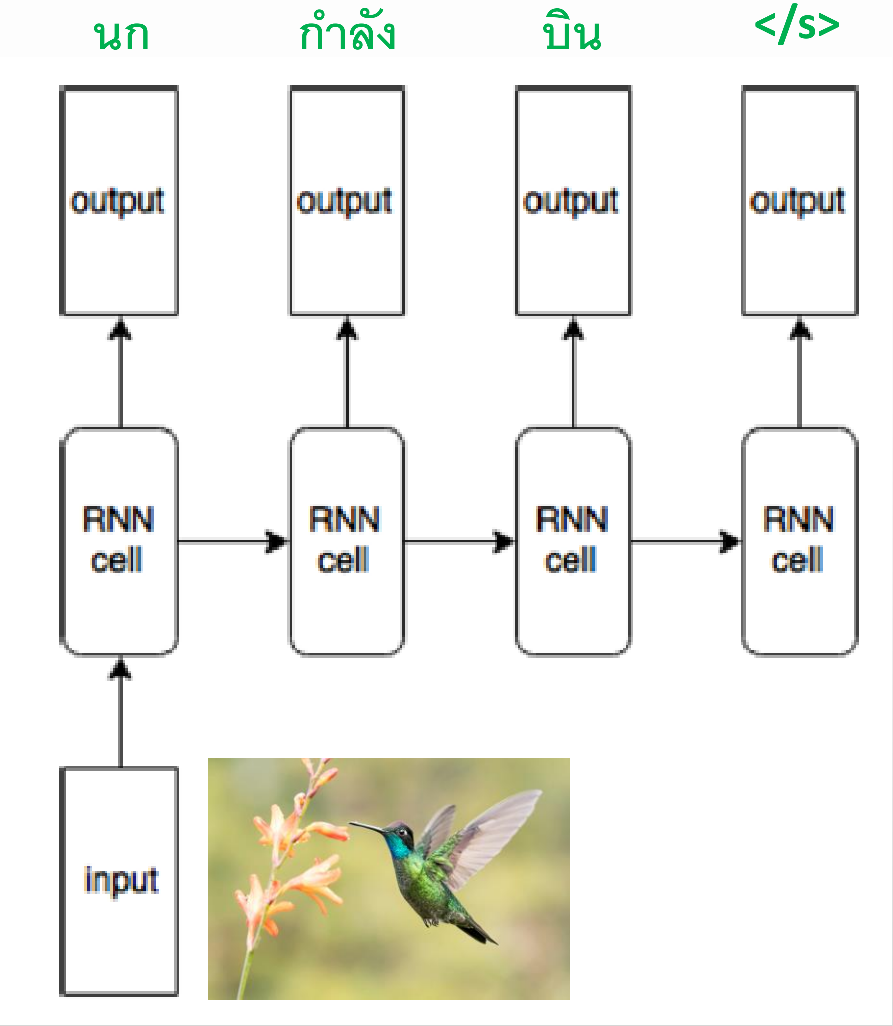
RNN architectures in NPL

- **Many to one**
- Sequence input
- E.g. Sentiment Analysis, Text classification



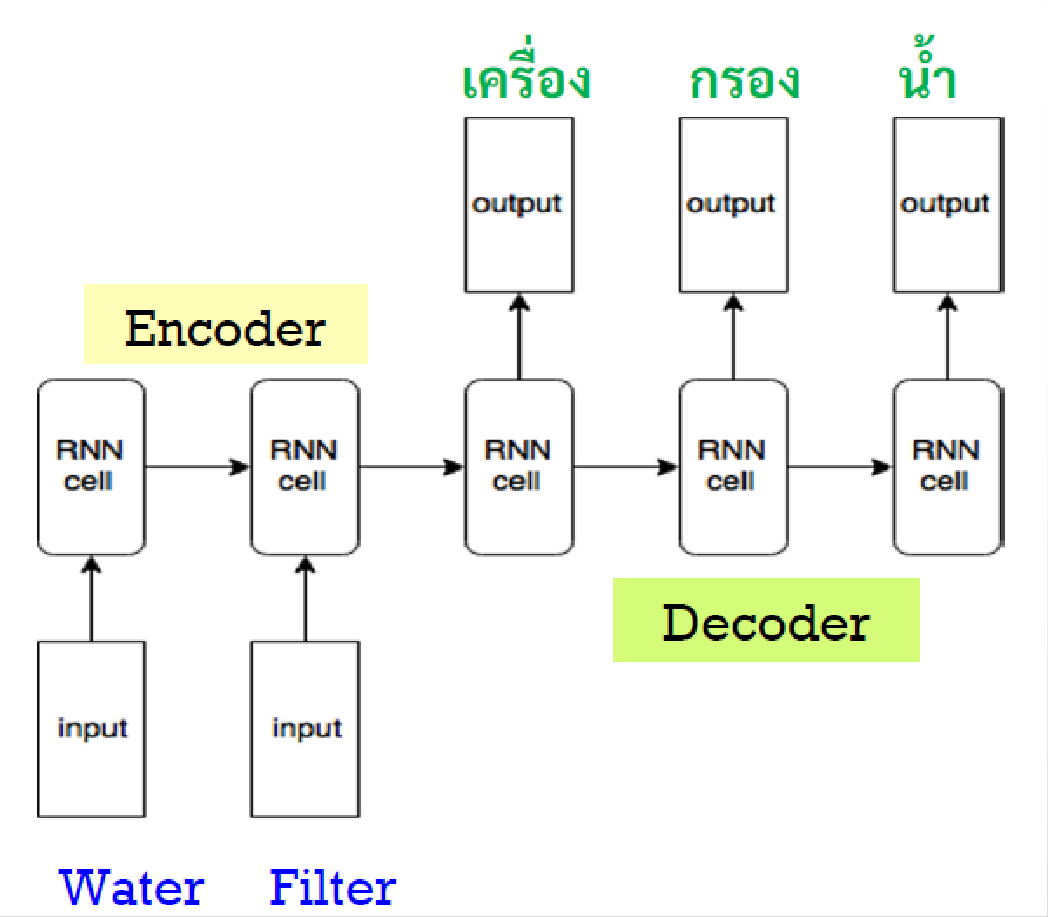
RNN architectures in NPL

- **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



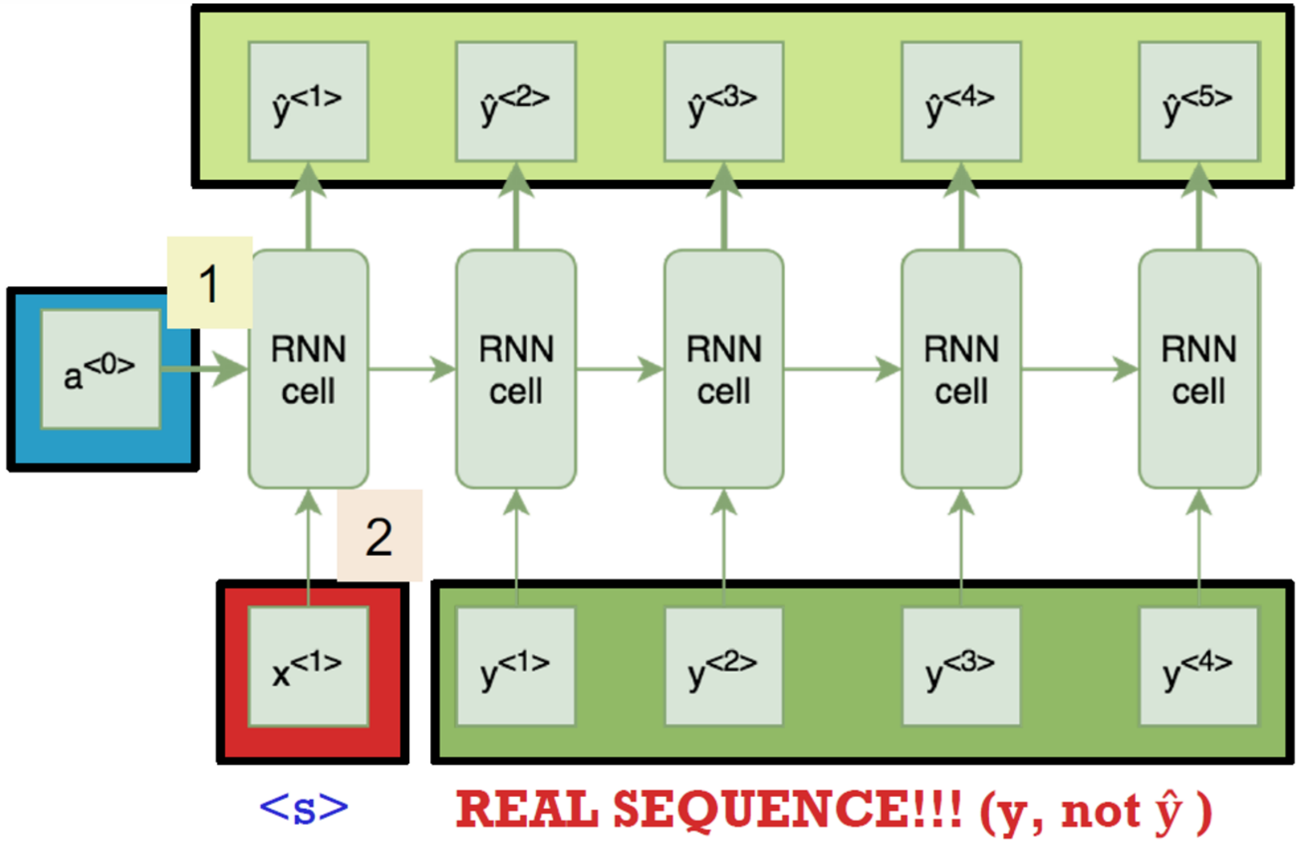
RNN architectures in NPL

- **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)

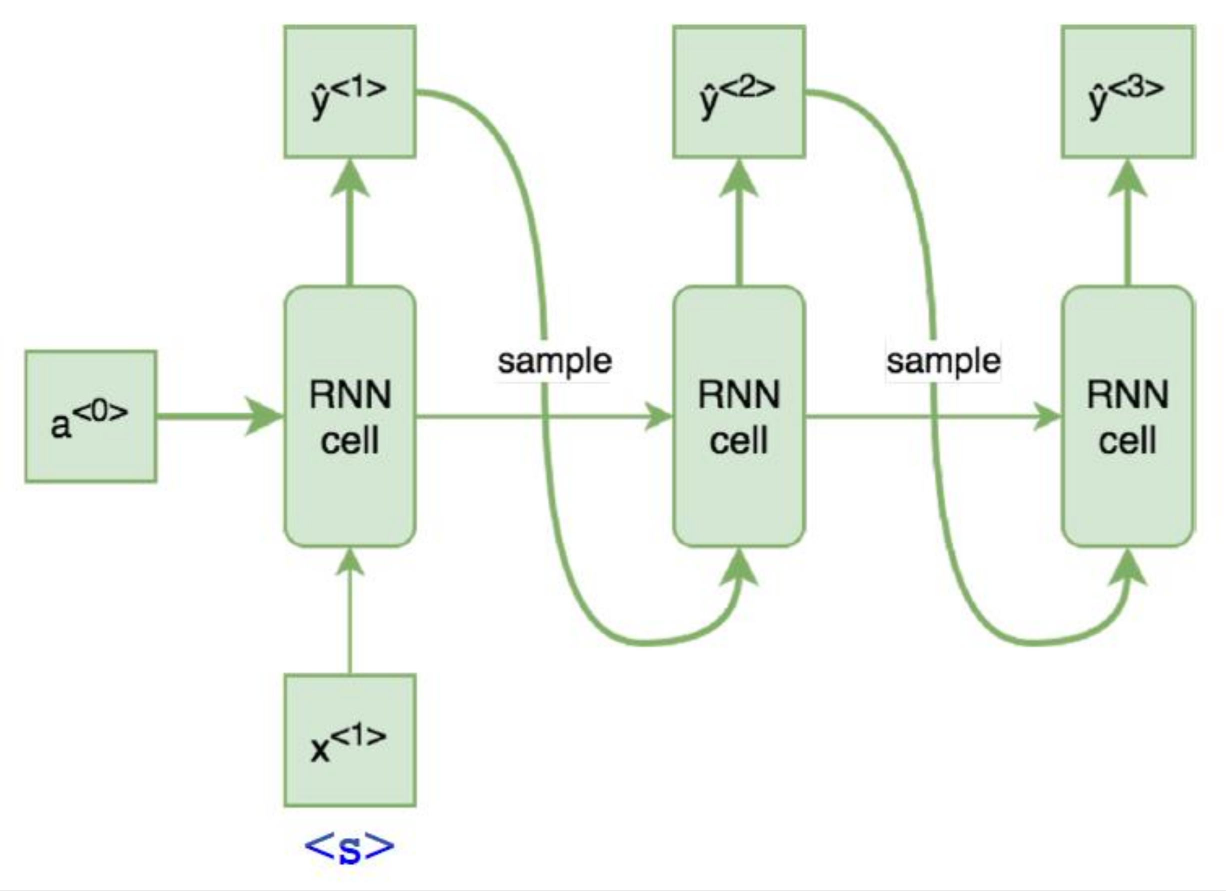
- One-to-Many RNN (autoregressive)
- The only real input is $x^{<1>}$
- $a^{<0>}$ is the initial hidden state.
- $\hat{y}$ is the predicted output.
- y is an actual output.
- During the training phase, instead of using the predicted output to feed into the next time-step, we use the actual output.



$$a^{<t>} = \overset{1}{\boxed{W a^{<t-1>}}} + \overset{2}{\boxed{W x^{<t>}}} + b$$

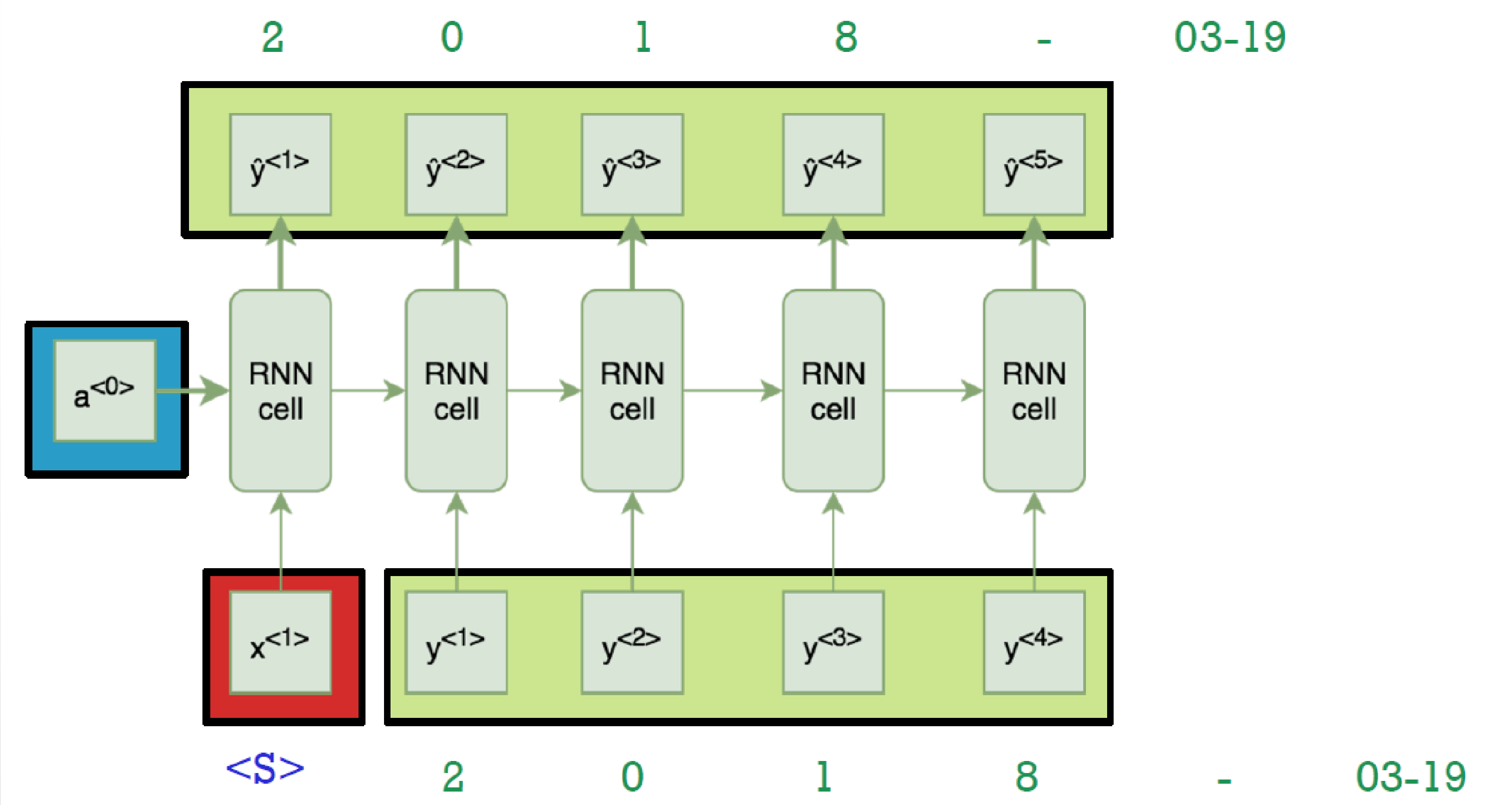
Text generation model (inference; testing)

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



Text generation: Demo

- Generating a piece of text using RNN; Random Date Generation "2018-03-19"

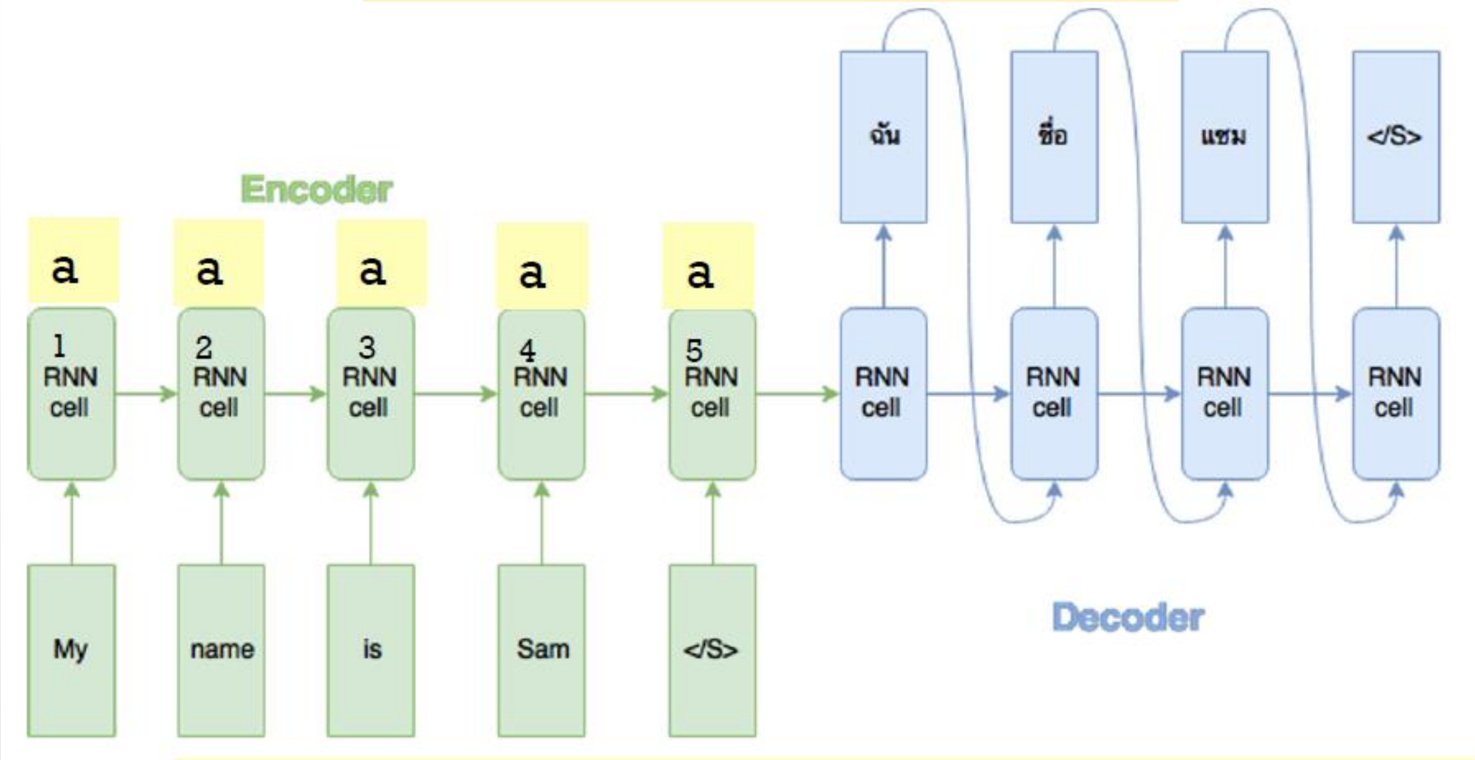


Attention Mechanism (Many-to-Many)

- 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 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?

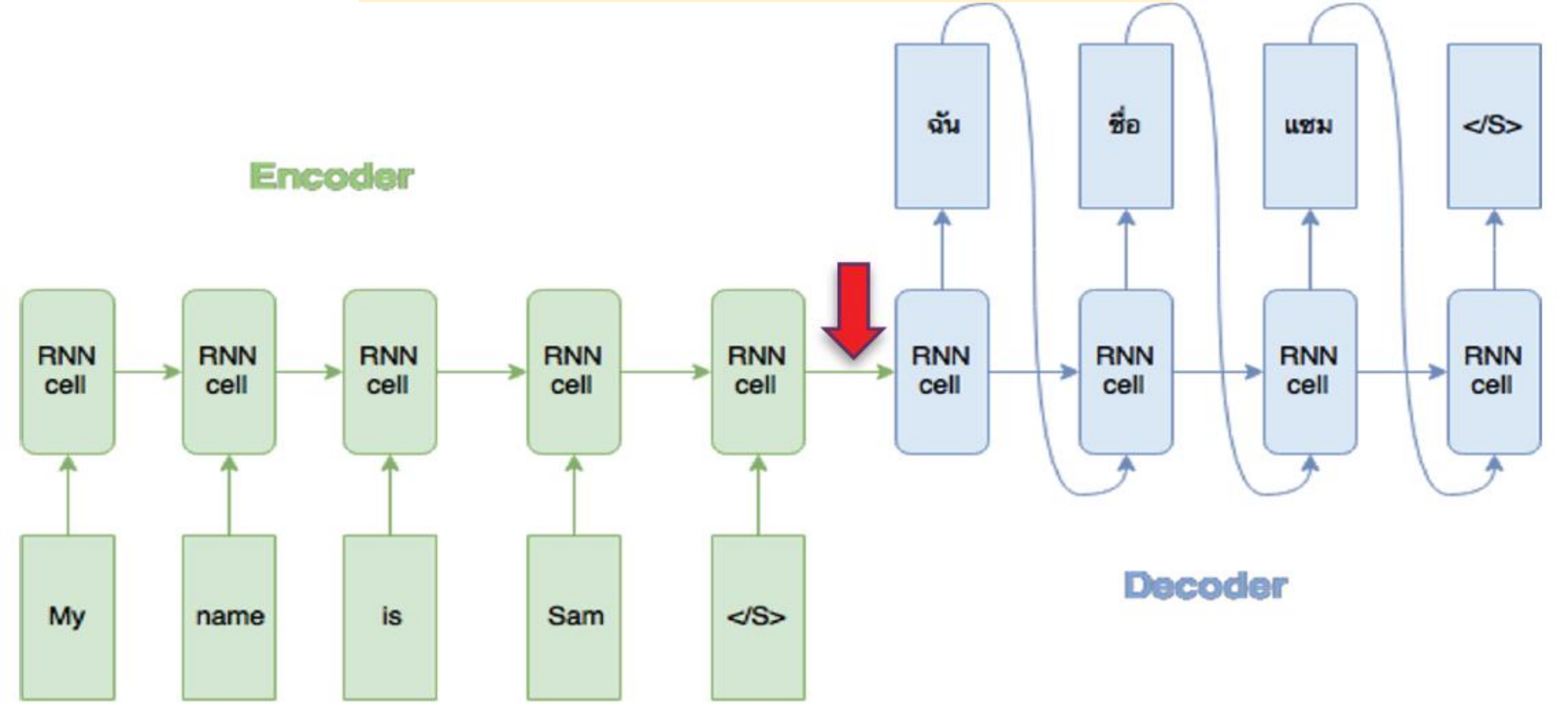
Machine Translation: English to Thai



Attention Mechanism (Many-to-Many)

- Why attention?
  - “You can’t cram the meaning of a whole %&!$# sentence into a single $&!#* vector!” – Raymond Mooney (2014)

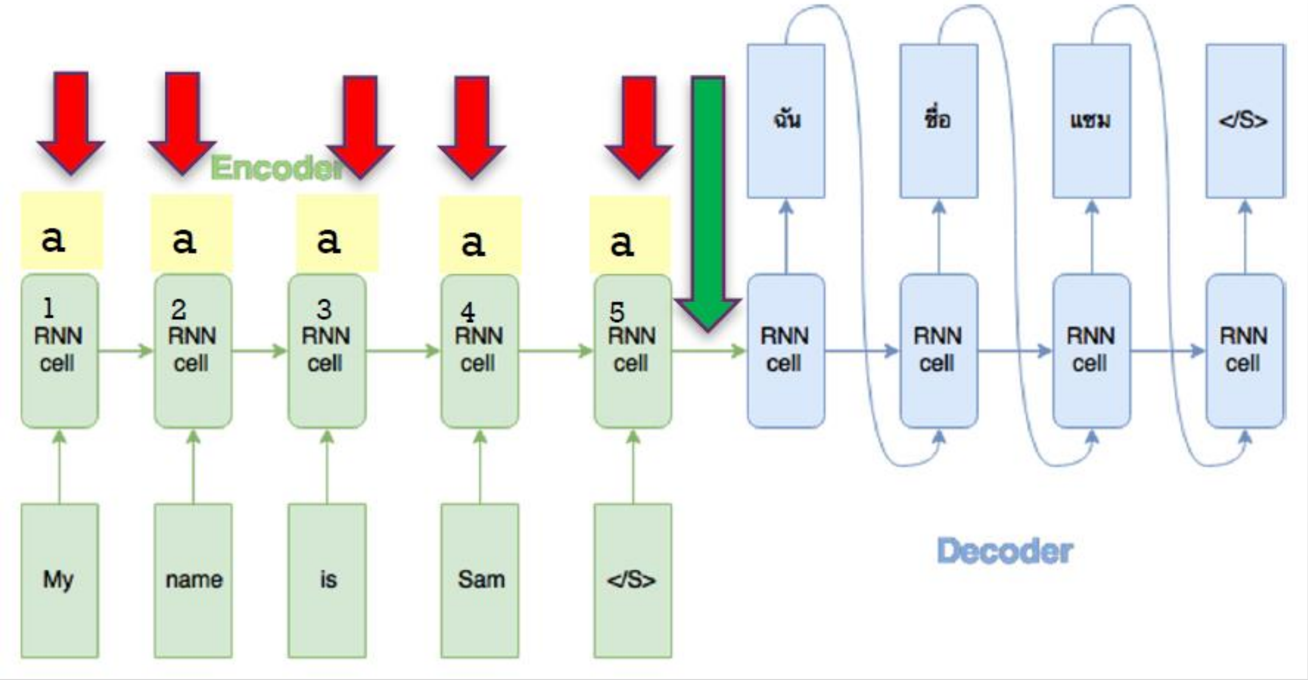
Machine Translation: English to Thai



Attention Mechanism (Many-to-Many)

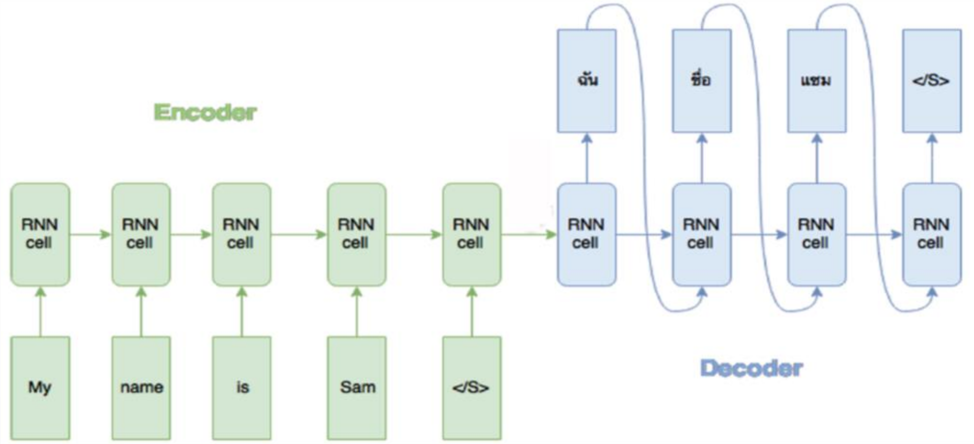
- Why attention?
  - Main idea: We can use multiple vectors based on the length of the sentence instead of one.
  - Attention mechanism = Instead of encoding all the information into a fixed-length vector, the decoder gets to decide parts of the input source to pay attention.

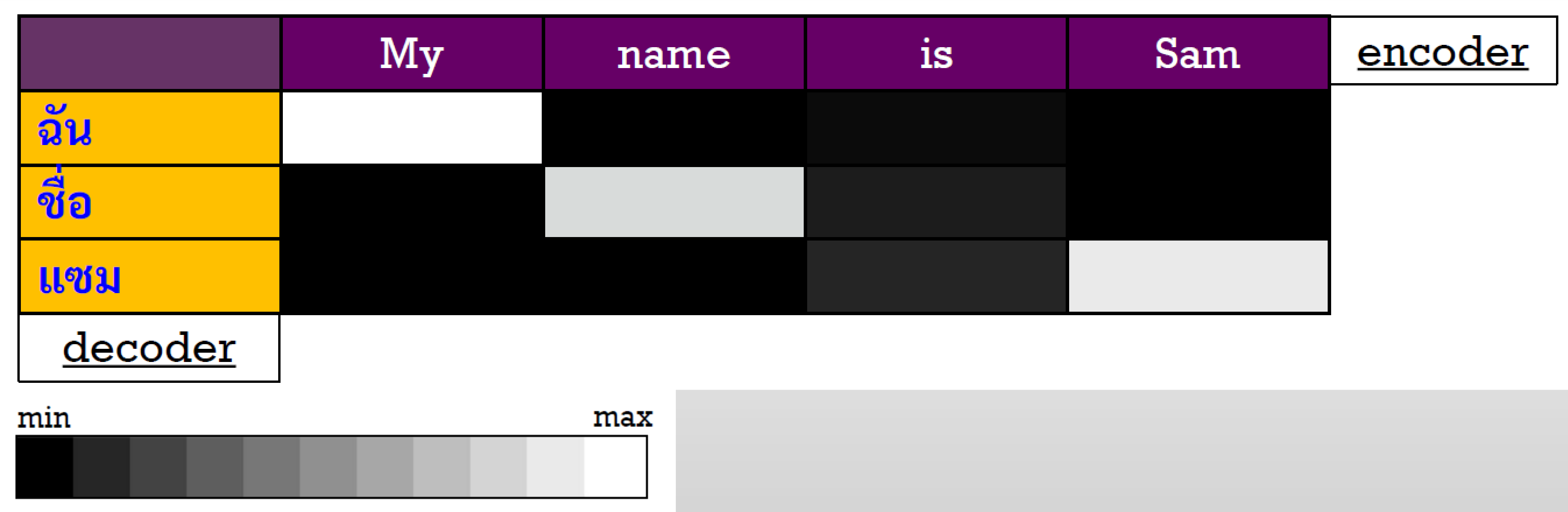
Machine Translation: English to Thai



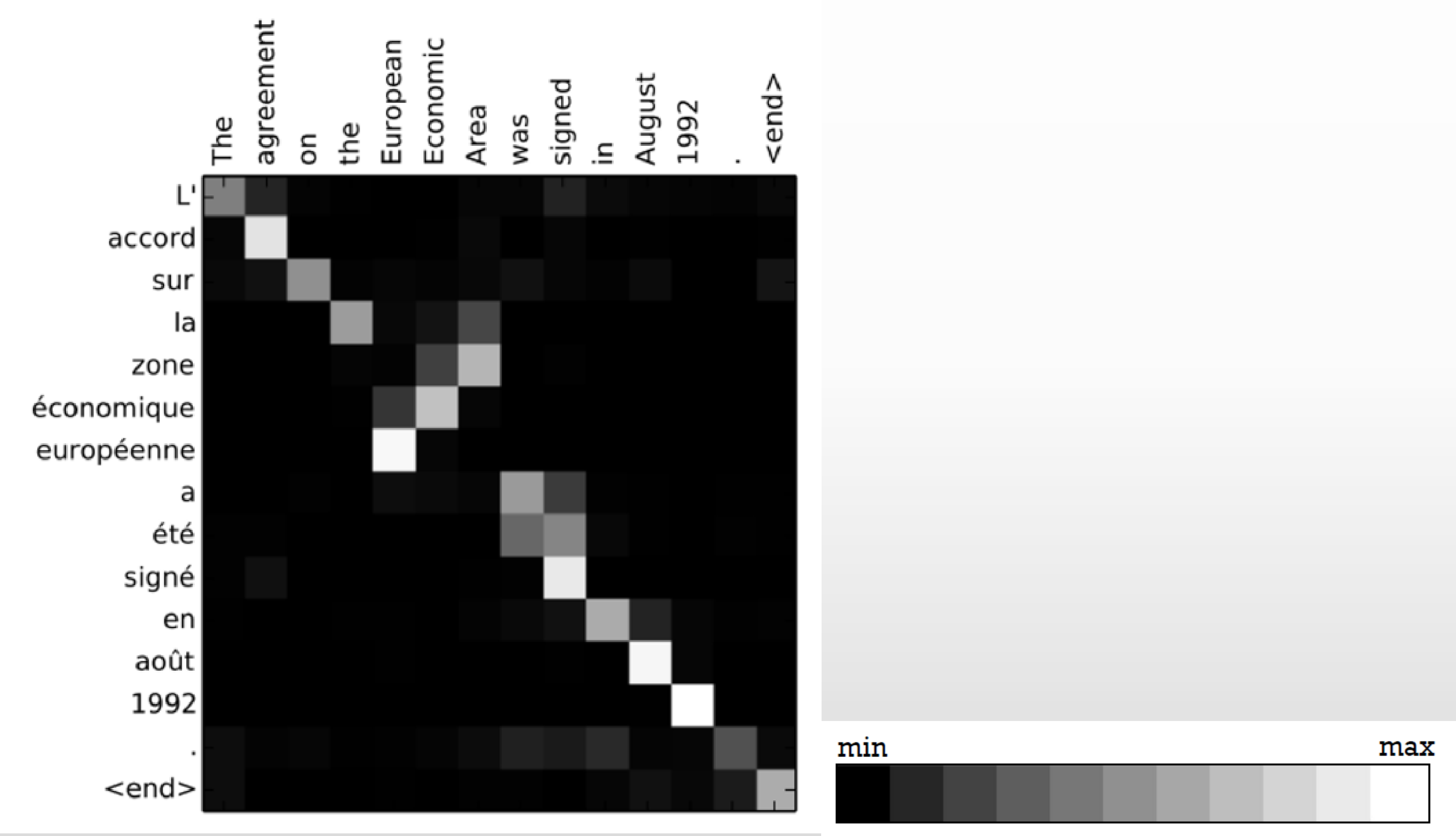
Graphical Example: English-to-Thai machine translation

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





Graphical Example: English-to-French machine translation

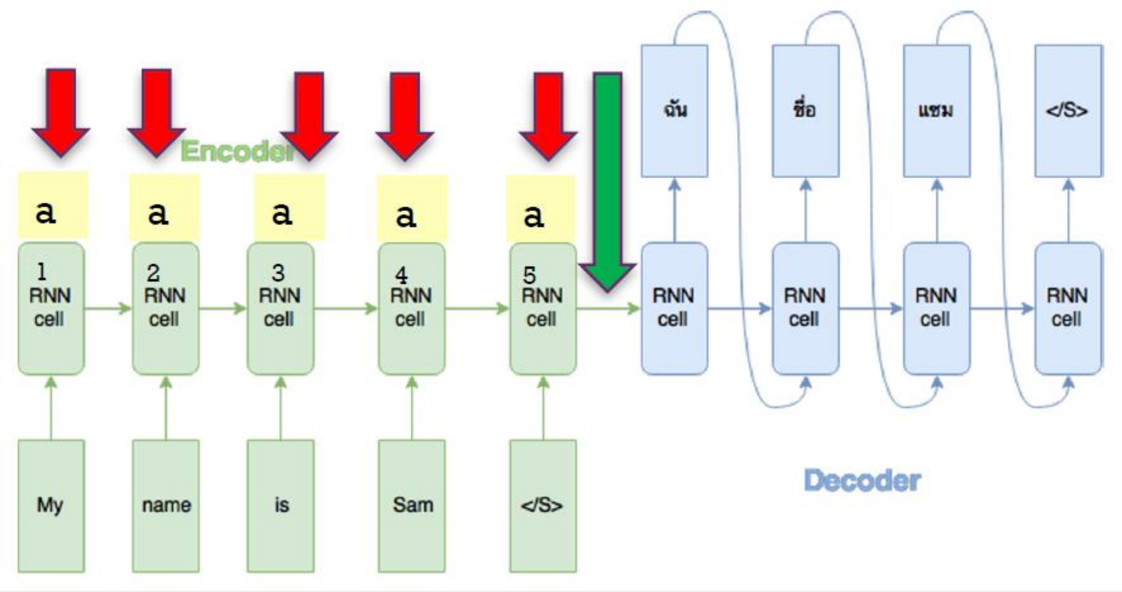


Reference: Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." ICLR(2015).

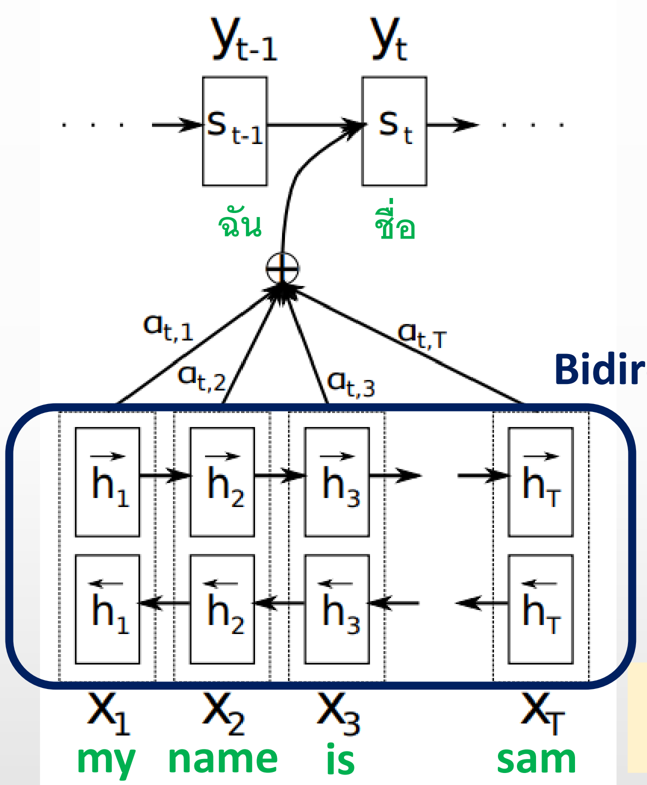
Attention Mechanism: Recap Basic Idea

- 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)
  - $c_i = \sum_j a_{ij} h_j$
  - j: each encoder’s input
  - i: each decoder’s input
- A vector formed by this linear combination is called “context vector”
- Use context vectors as inputs for the decoding step

Attention Mechanism: Recap Basic Idea



Decoder


y_{t-1}
y_t
s_{t-1}
s_t
ฉัน
ชื่อ
$\alpha_{t,1}$
$\alpha_{t,2}$
$\alpha_{t,3}$
$\alpha_{t,T}$
$\overrightarrow{h}_1$ $\overrightarrow{h}_2$ $\overrightarrow{h}_3$ $\overrightarrow{h}_T$
$\overleftarrow{h}_1$ $\overleftarrow{h}_2$ $\overleftarrow{h}_3$ $\overleftarrow{h}_T$
x_1 x_2 x_3 x_T
my
name
is
sam

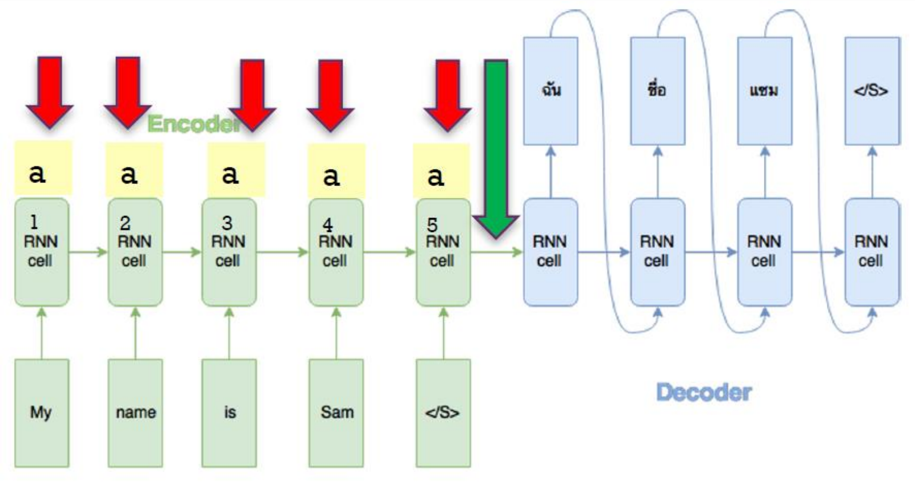
Bidirectional LSTM

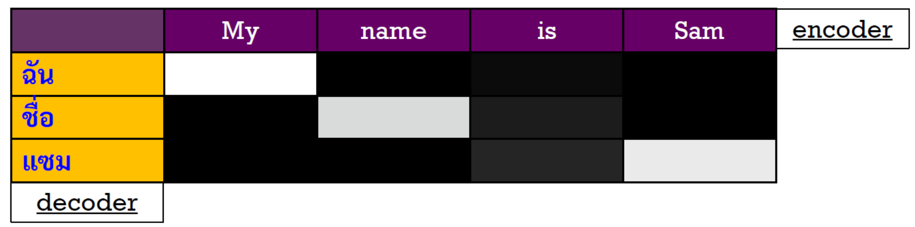
Encoder

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

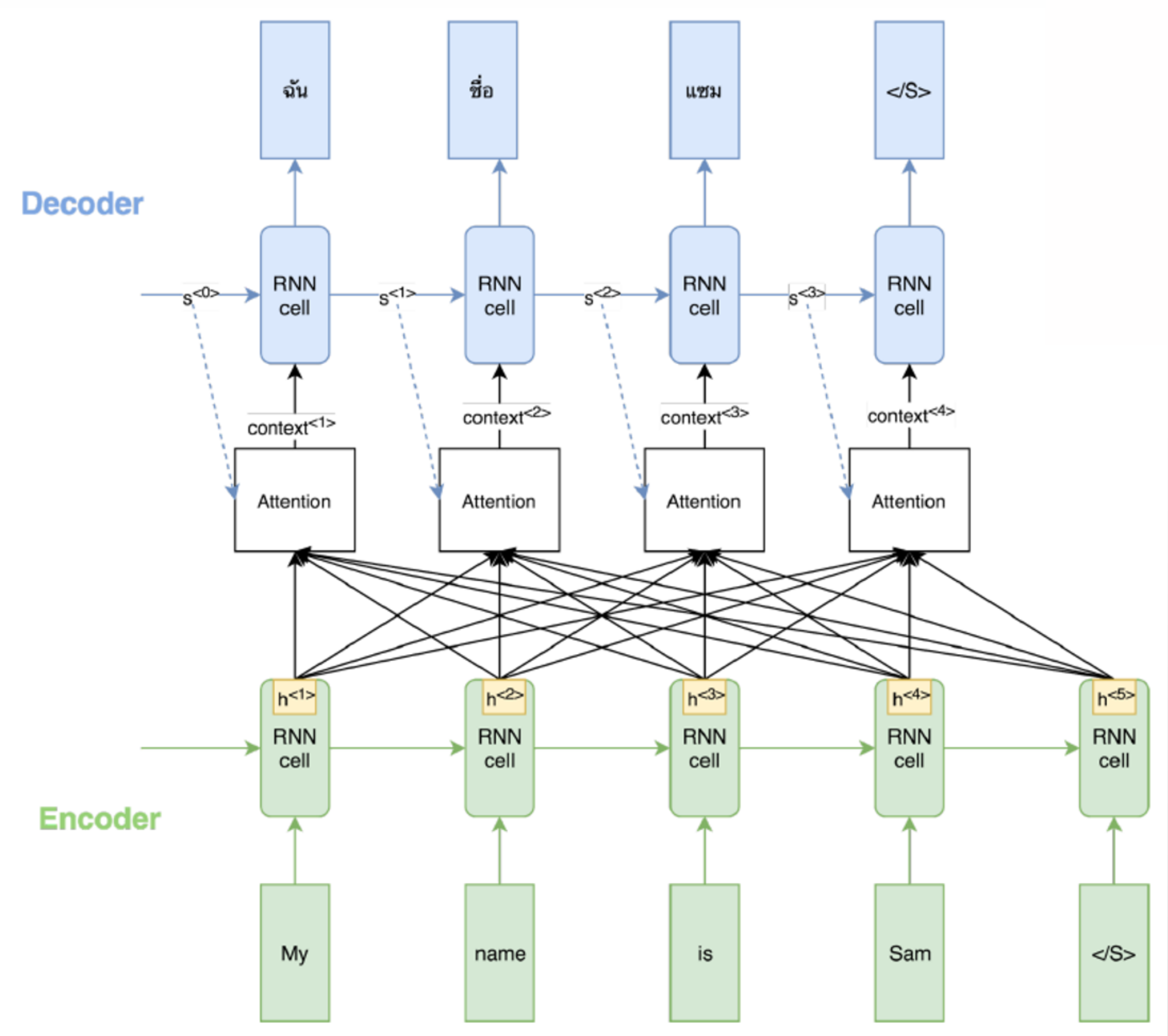
Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

RNN and attention mechanism

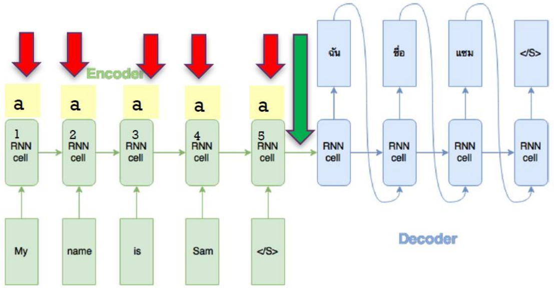




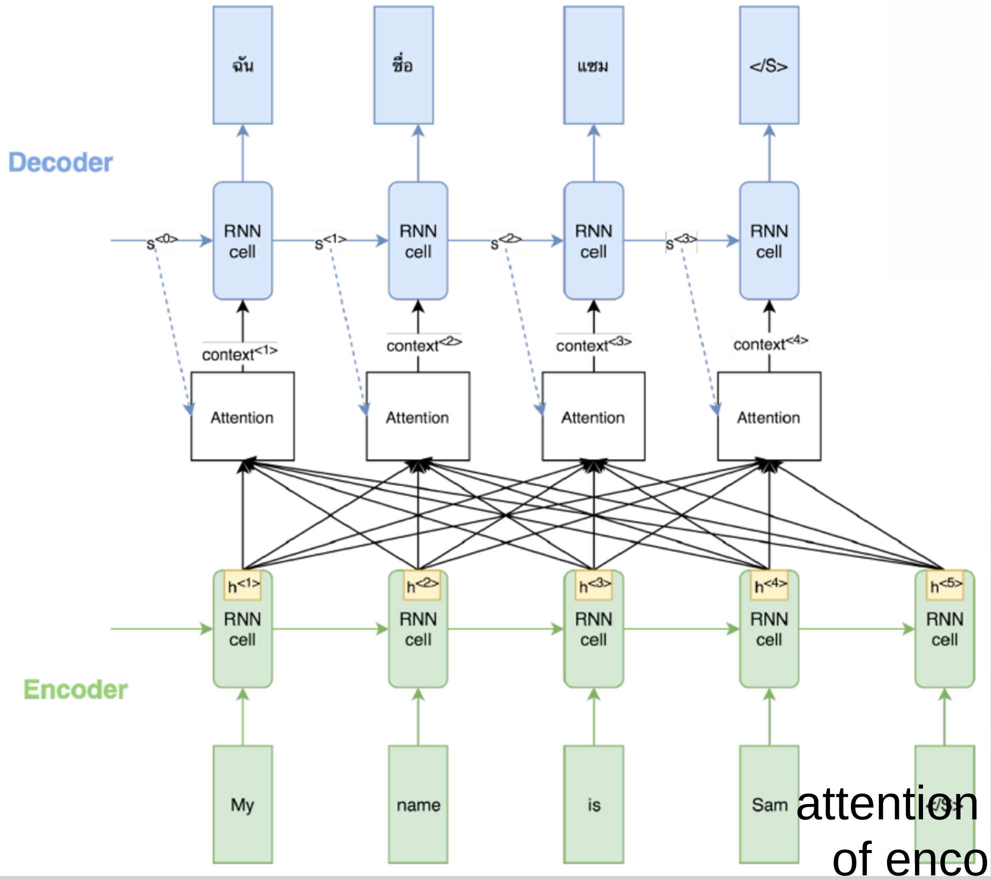
	My	name	is	Sam	encoder
ฉัน	⬜	⬛	⬛	⬛	
ชื่อ	⬛	⬜	⬛	⬛	
แซม	⬛	⬛	⬛	⬜	
decoder					



Attention Mechanism: calculate c_i



We want to calculate a context vector c based on hidden states $s_0 \ldots s_{m-1}$ that can be used with the current state h_j for prediction. The context vector c_i at position "i" is calculated as an average of the previous states weighted with the attention scores a_i.



i:decoder index
j: encoder index

$$c_i = \sum_j a_{ij} h_j$$

context vector

attention score

encoder state at index j

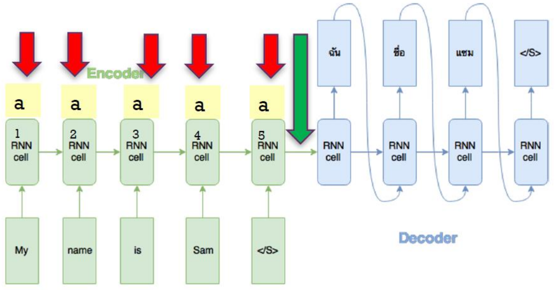
$$a_{ij} = softmax(f_{att}(s_{i-1}, h_j))$$

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

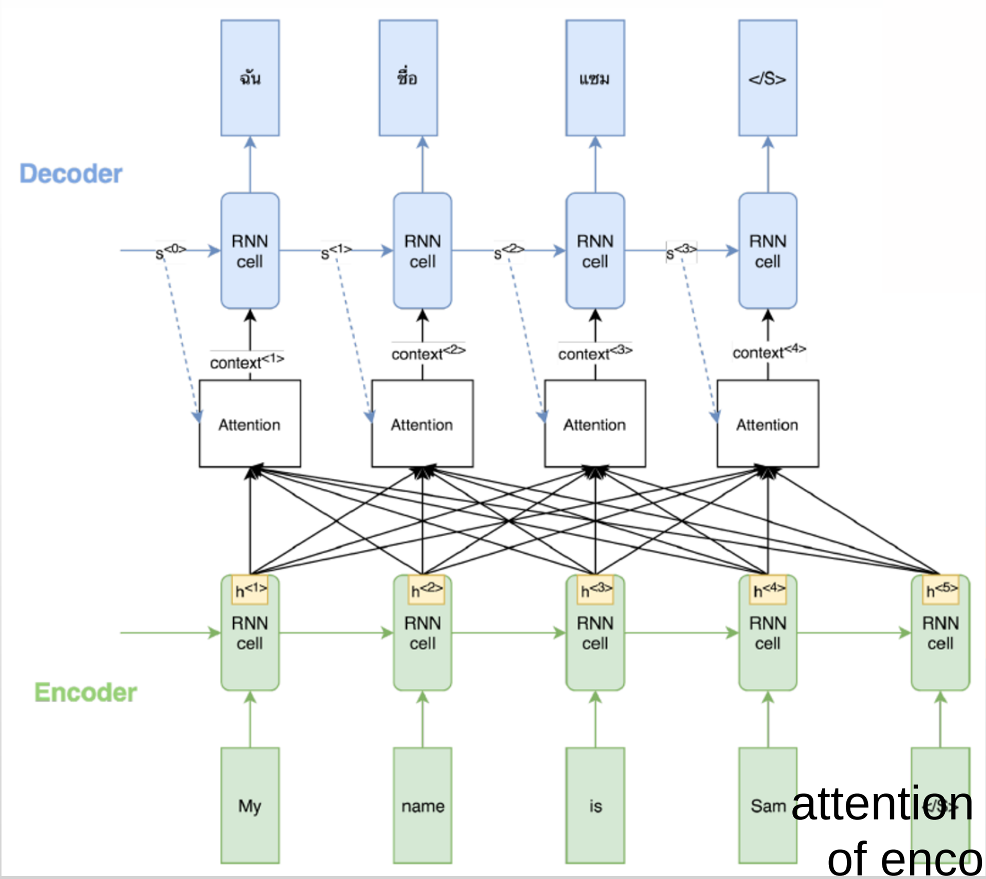
previous hidden state/ **decoder** state

encoder state at index j

Attention Mechanism: calculate f_{att}



The attention function $f_{att}(s_{i-1}, h_j)$ calculates an unnormalized alignment score between the current hidden state s_{i-1} and the previous hidden state h_j. There are many variants of the attention function f_{att}.



i:decoder index

j: encoder index

$$c_i = \sum_j a_{ij} h_j$$

context vector

attention score

encoder state at index j

$$a_{ij} = softmax(f_{att}(s_{i-1}, h_j))$$

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

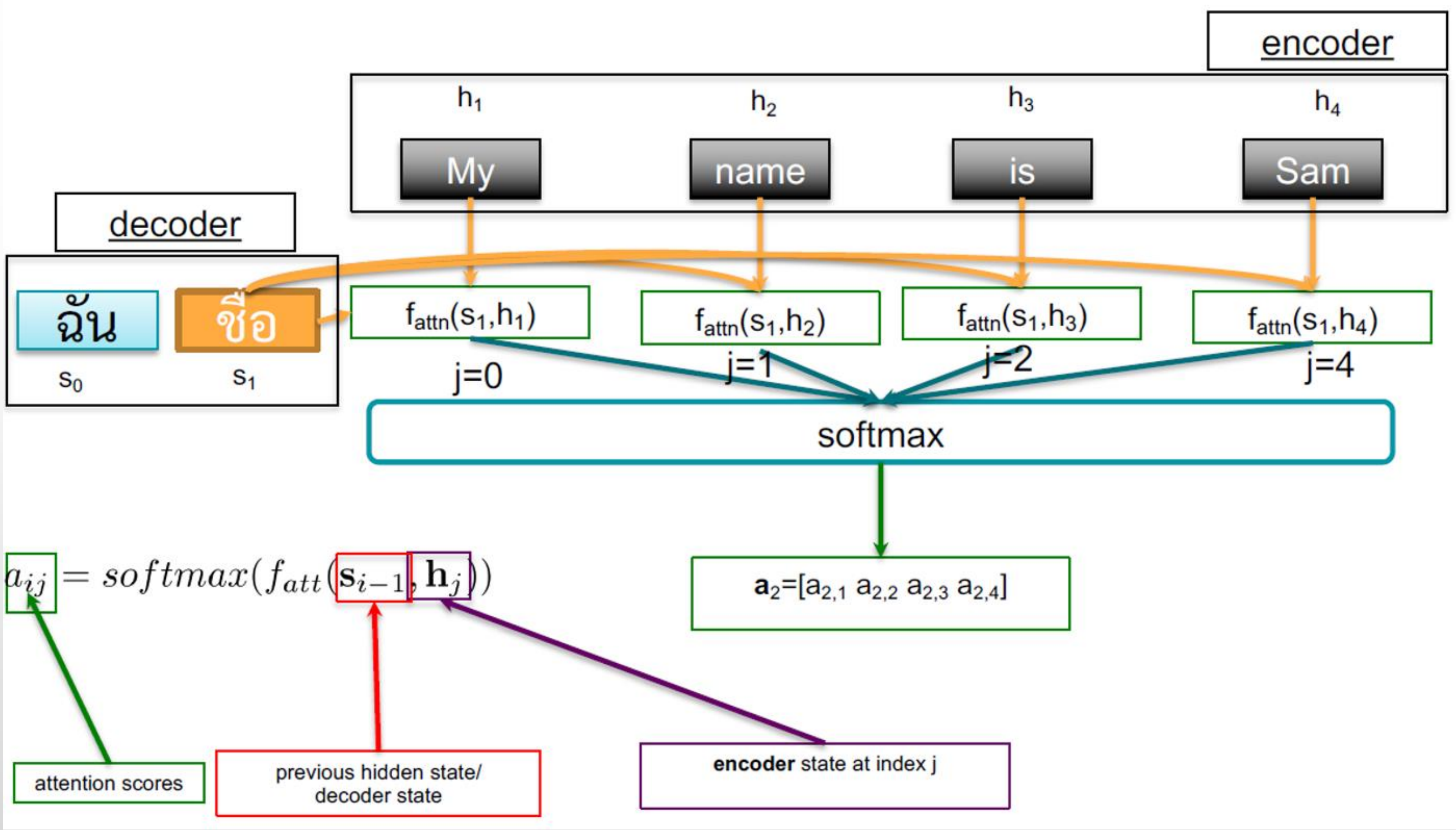
previous hidden state/ **decoder** state

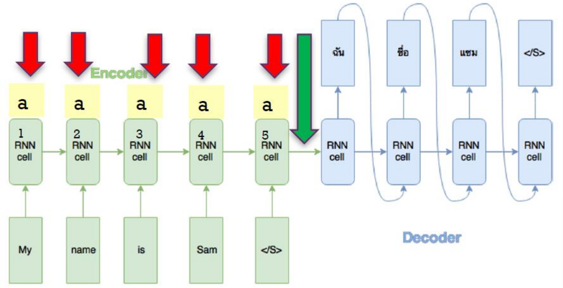
encoder state at index j

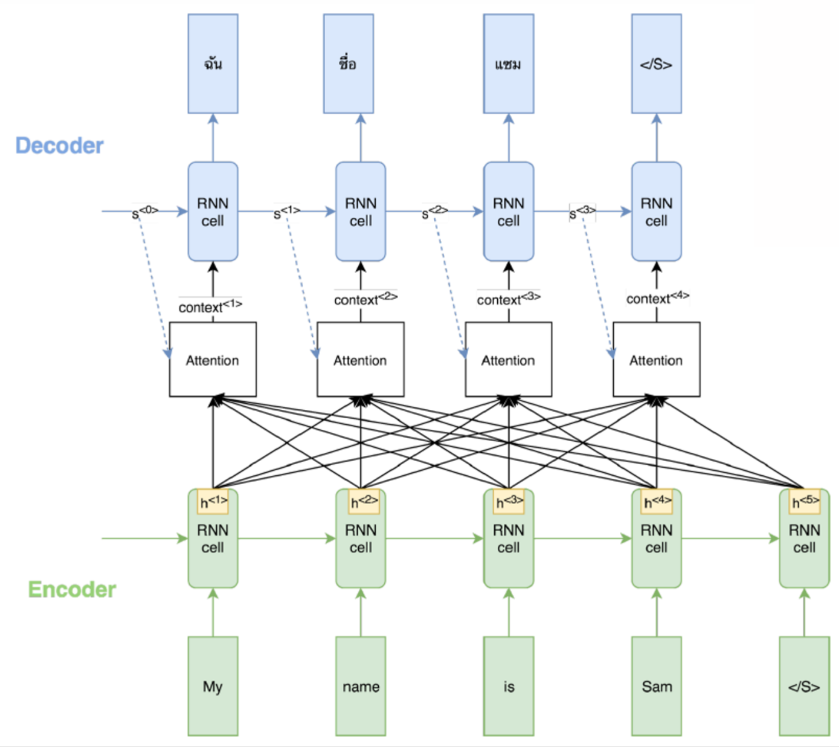
i:decoder index
j: encoder index

Attention Calculation: Attention Scores

- Example: Now we want to predict “ชื่อ”



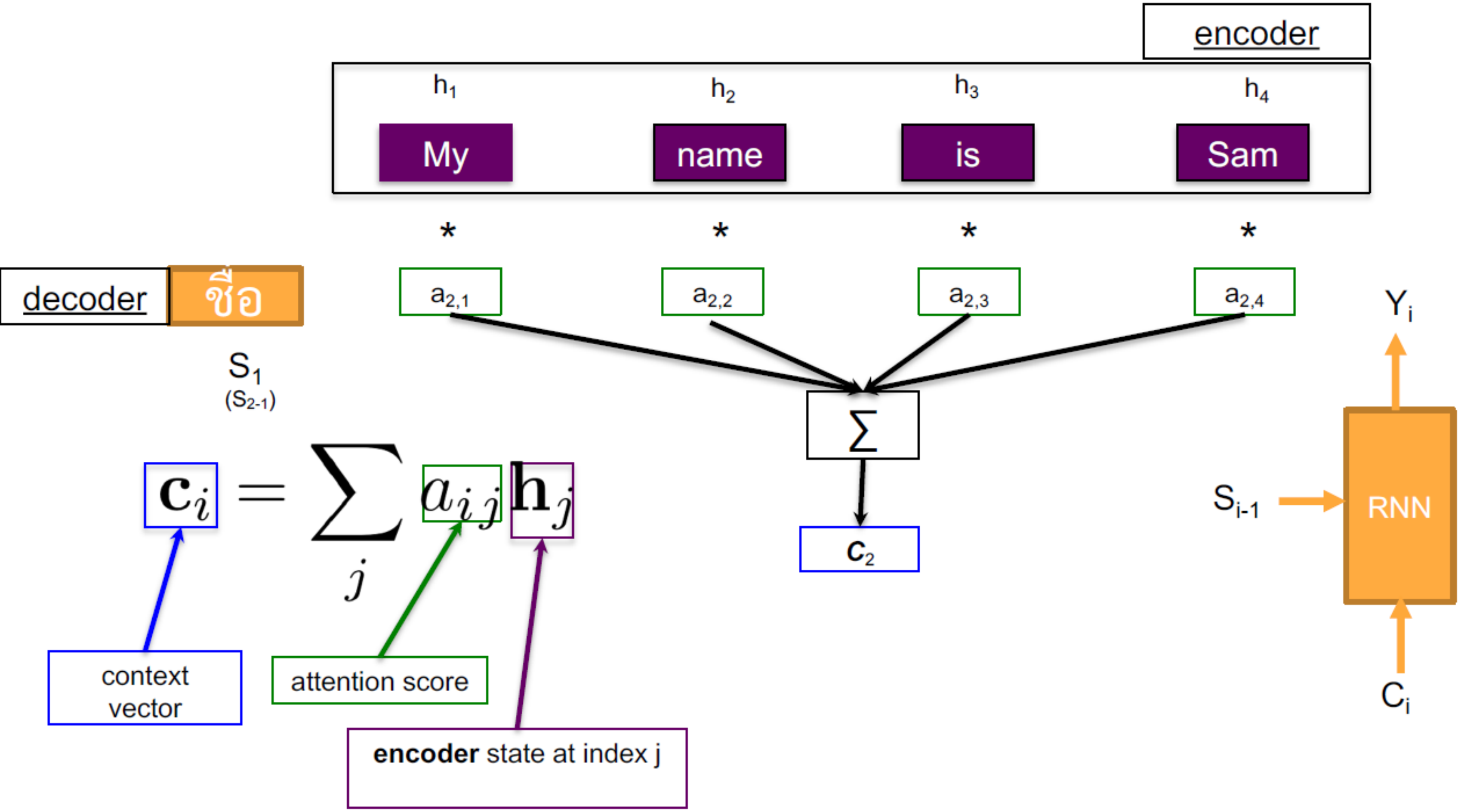




Attention Calculation: Context Vector

i:decoder index
j: encoder index

- Example: Now we want to predict “ชื่อ”

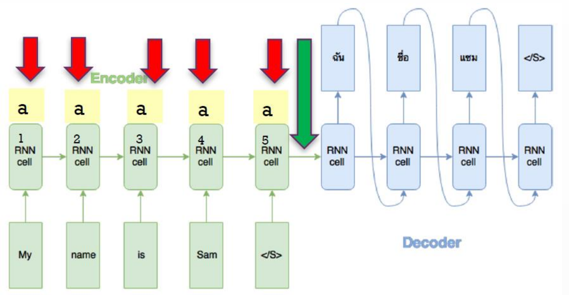


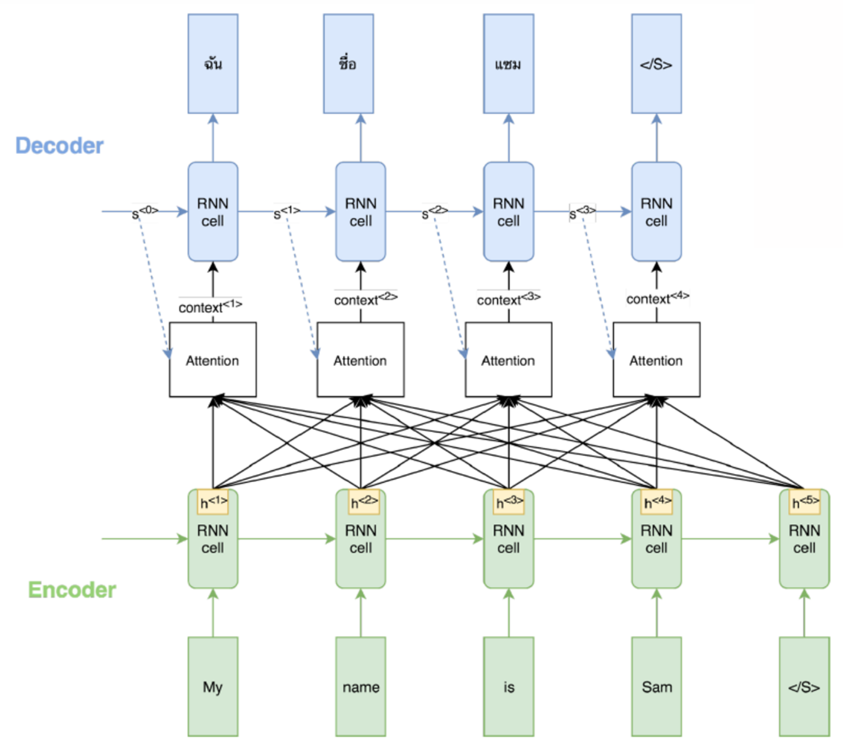
$$\mathbf{c}_i = \sum_j a_{ij}\mathbf{h}_j$$

context vector

attention score

encoder state at index j





Type of Attention Mechanisms

	My	name	is	Sam	encoder
ฉัน	□	■	■	■	
ชื่อ	■	□	▨	■	
แซม	■	■	▨	□	
decoder					

$$a_{ij} = softmax(f_{att}(s_{i-1}, h_j))$$

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

$$f_{att}\left(s_{i-1}, h_j\right) = tanh(\mathrm{W}_a[s_{i-1}; h_j])$$

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

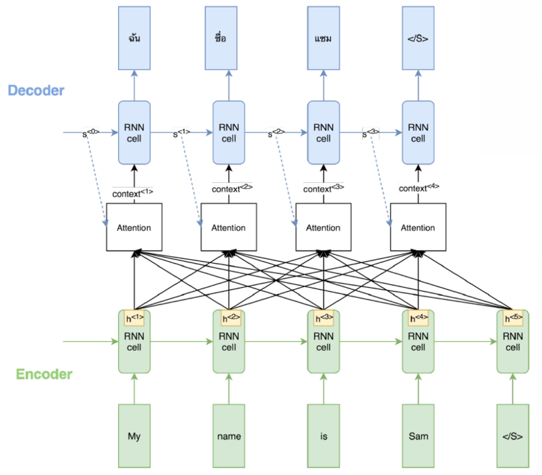
$$f_{att}\left(s_{i-1}, h_j\right) = \mathrm{S}_{i-1}^T W_a h_j$$

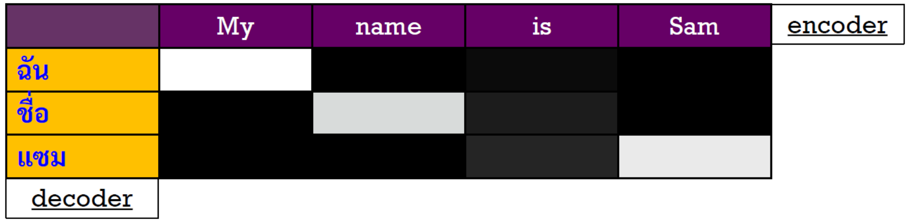
- **Self-attention:** Without any additional information, however, we can still extract relevant aspects from the sentence by allowing it to attend to itself using self-attention (Lin et al., 2017)

$$a = softmax(W_{s_2} \tanh(W_{s_1} H^T))$$

- **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.

Additive Attention





$$a_{ij} = softmax(f_{att}(s_{i-1}, h_j))$$

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

concatenation

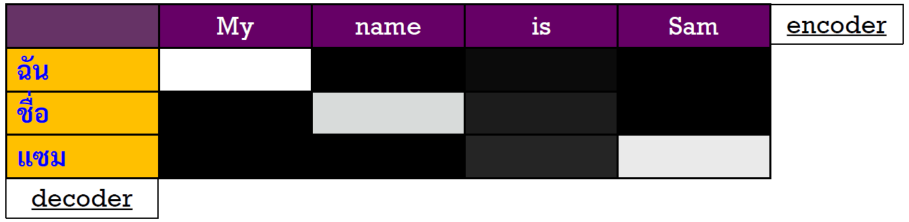
$$f_{att}(s_{i-1}, h_j) = tanh(\mathrm{W}_a[s_{i-1}; h_j])$$

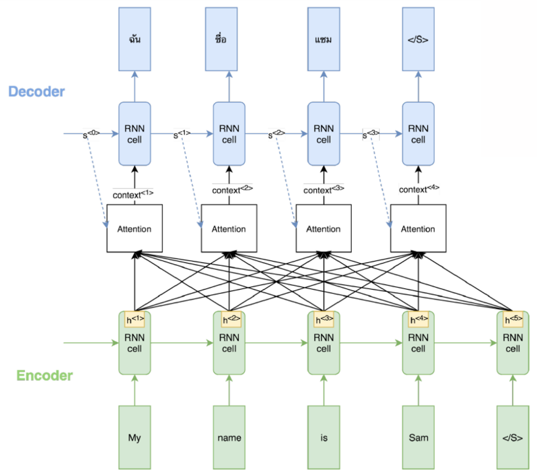
One-hidden layer
(Dense)

- Where W_a are learned attention parameters. Analogously, we can also use matrices W_1 and W_2 to learn separate transformations for s_{i-1} and h_j respectively, which are then summed (hence the name additive):

$$f_{att}(s_{i-1}, h_j) = tanh(\mathrm{W}_1 s_{i-1} + \mathrm{W}_2 h_j)$$

Multiplicative Attention





$$a_{ij} = softmax(f_{att}(s_{i-1}, h_j))$$

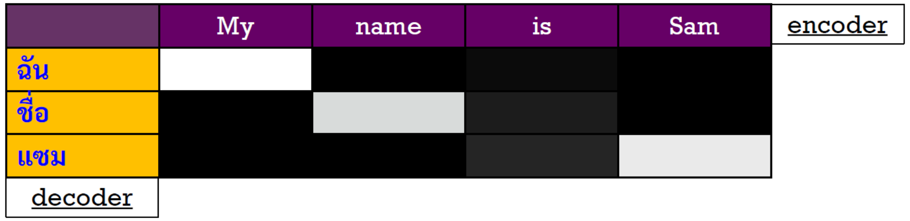
- Multiplicative attention (Luong et al., 2015) simplifies the attention operation by calculating the following function:

$$f_{att}\left(s_{i-1}, h_j\right) = S_{i-1}^{T} W_a h_j$$

- Faster, more efficient than additive attention BUT additive attention performs better for larger dimensions
- One way to mitigate this is to scale f_{att} by $\frac{1}{\sqrt{d_s}}$

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

Self Attention



$$a_{ij} = softmax(f_{att}(s_{i-1}, h_j))$$

- 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)

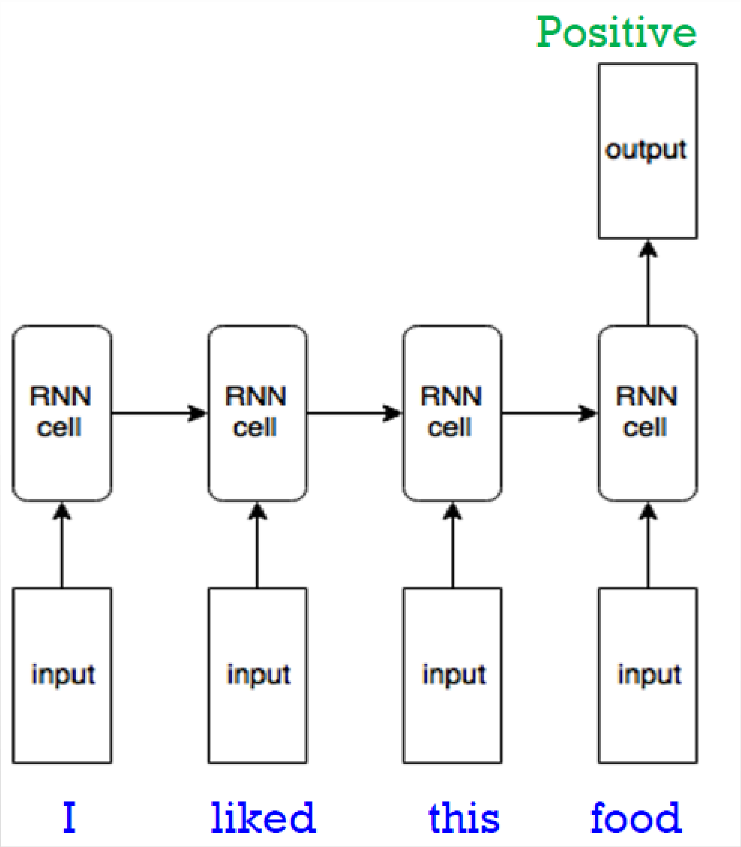
$$H = (\mathrm{h}_1, h_2, \dots h_n)$$

Fully connected layer

$$a = softmax(W_{s_2} \tanh(W_{s_1} H^T))$$

One-hidden layer (Dense)

- w_{s_1} is a weight matrix, w_{s_2} 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.



Self Attention

	My	name	is	Sam	encoder
ฉัน	□	■	■	■	
ชื่อ	■	▨	■	■	
แซม	■	■	■	▨	
decoder					

$$a_{ij} = softmax(f_{att}(s_{i-1}, h_j))$$

- 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)

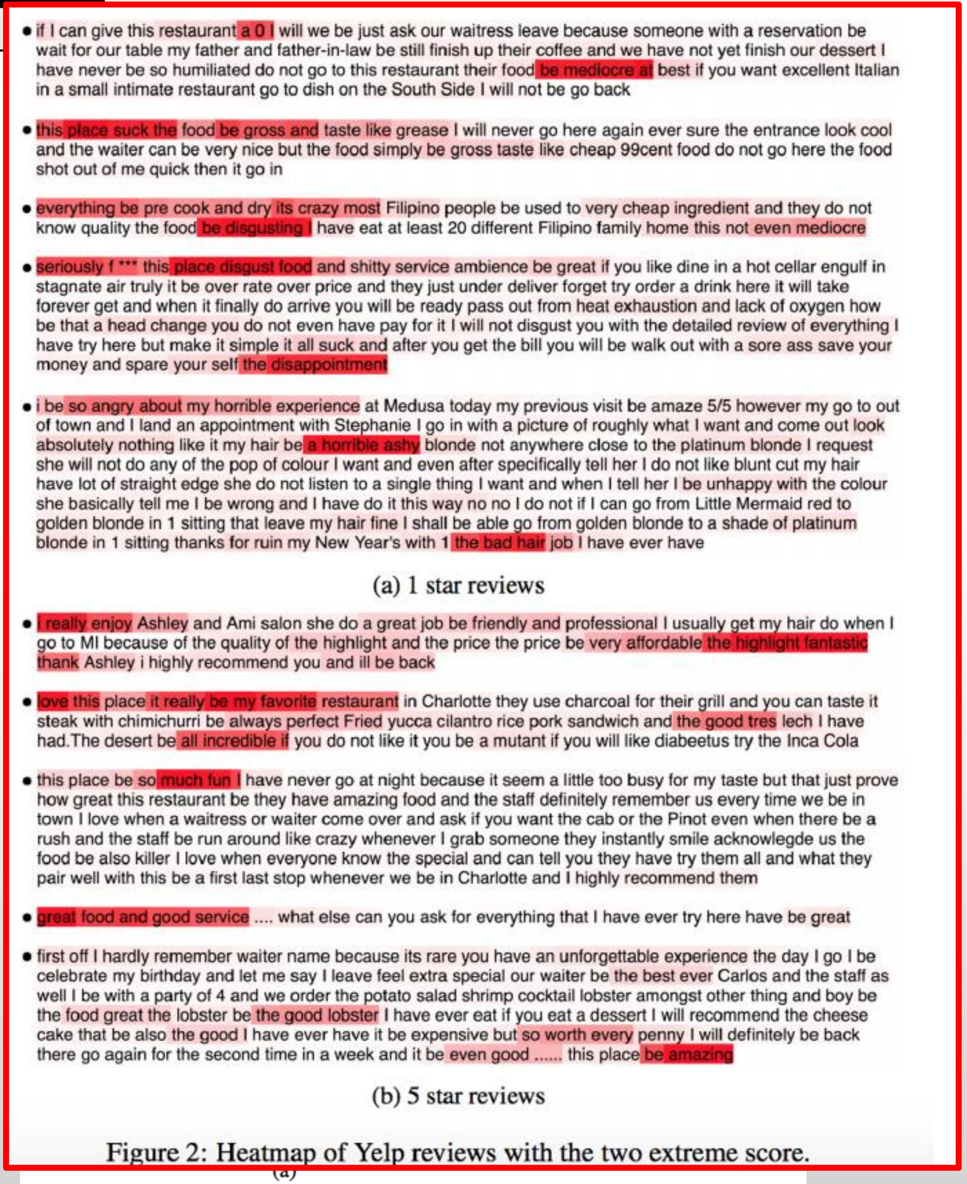
$$H = (\mathrm{h}_1, h_2, \dots h_n)$$

Fully connected layer

$$a = softmax(W_{s_2} \tanh(W_{s_1} H^T))$$

One-hidden layer (Dense)

- w_{s_1} is a weight matrix, w_{s_2} 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.



- 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

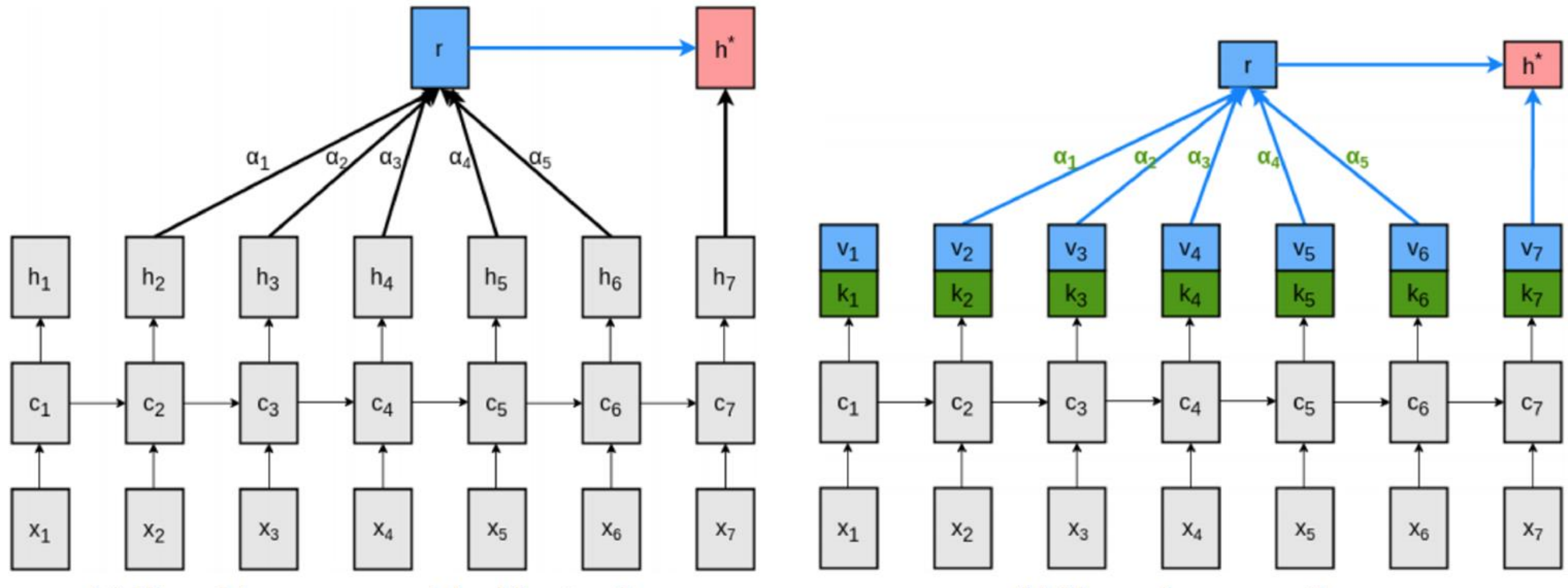
Figure 2: Heatmap of Yelp reviews with the two extreme score.

(a)

Key-value attention

$$a_{ij} = softmax(f_{att}(s_{i-1}, h_j))$$

$$c_i = \sum_j a_{ij} h_j$$



(a) Neural language model with attention.

(b) Key-value separation.

Value=encoded vector

Key=used for attention score calculation

Daniluk, M., Rocktäschel, T., Welbl, J., & Riedel, S. (2017). Frustratingly short attention spans in neural language modeling. arXiv preprint arXiv:1702.04521.

Key-value attention

$$c_i = \sum_j a_{ij} h_j$$

k: output dimension
L: output length

previous outputs (memory) with window L

current output

key

value

$$\begin{bmatrix} \boldsymbol{k}_t \\ \boldsymbol{v}_t \end{bmatrix} = \boldsymbol{h}_t \qquad \in \mathbb{R}^{2k}$$

$$\boldsymbol{M}_t = \tanh(\boldsymbol{W}^Y \boxed{[\boldsymbol{k}_{t-L} \ \cdots \ \boldsymbol{k}_{t-1}]} + (\boldsymbol{W}^h \boldsymbol{k}_t)\mathbf{1}^T) \qquad \in \mathbb{R}^{k \times L}$$

Attention score

$$\boldsymbol{\alpha}_t = \mathrm{softmax}(\boldsymbol{w}^T \boldsymbol{M}_t) \qquad \in \mathbb{R}^{1 \times L}$$

Context vector (v_t not included)

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

The final representation

$$\boldsymbol{h}_t^* = \tanh(\boldsymbol{W}^r \boldsymbol{r}_t + \boldsymbol{W}^x \boldsymbol{v}_t) \qquad \in \mathbb{R}^{k}$$

Daniluk, M., Rocktäschel, T., Welbl, J., & Riedel, S. (2017). Frustratingly short attention spans in neural language modeling. arXiv preprint arXiv:1702.04521.

Introduction to Question Answering

- 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 (ChatGPT, Siri, Cortana, etc.)

Type Of Question Answering

- 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 Question Answering

- 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

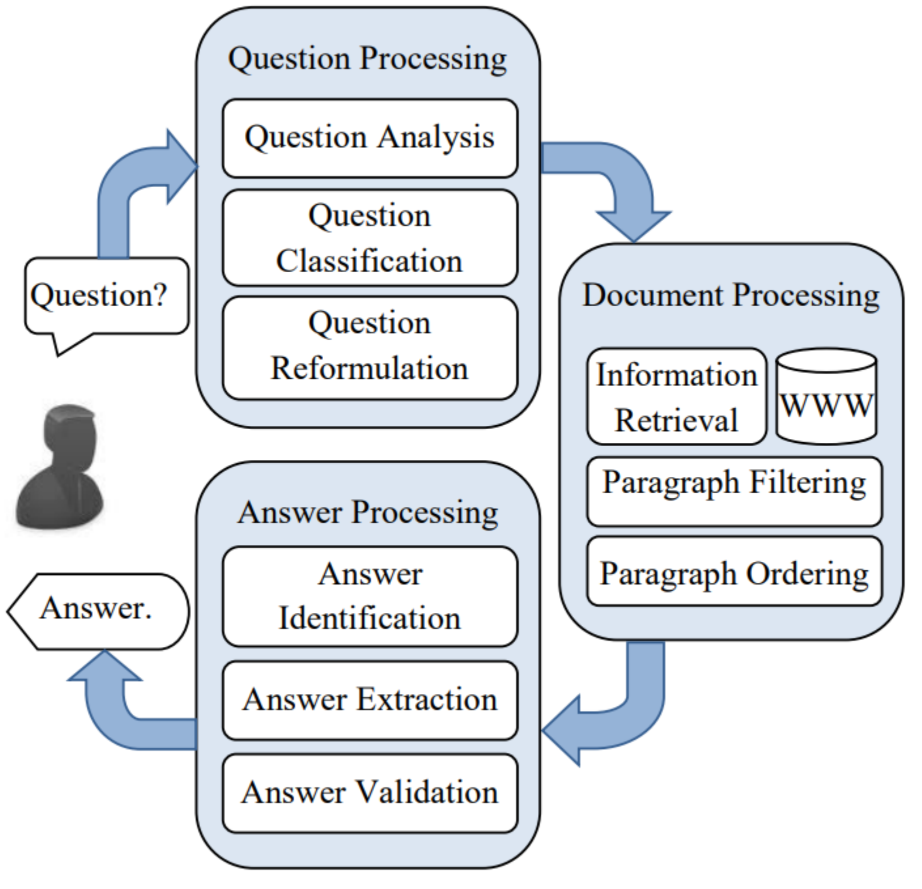
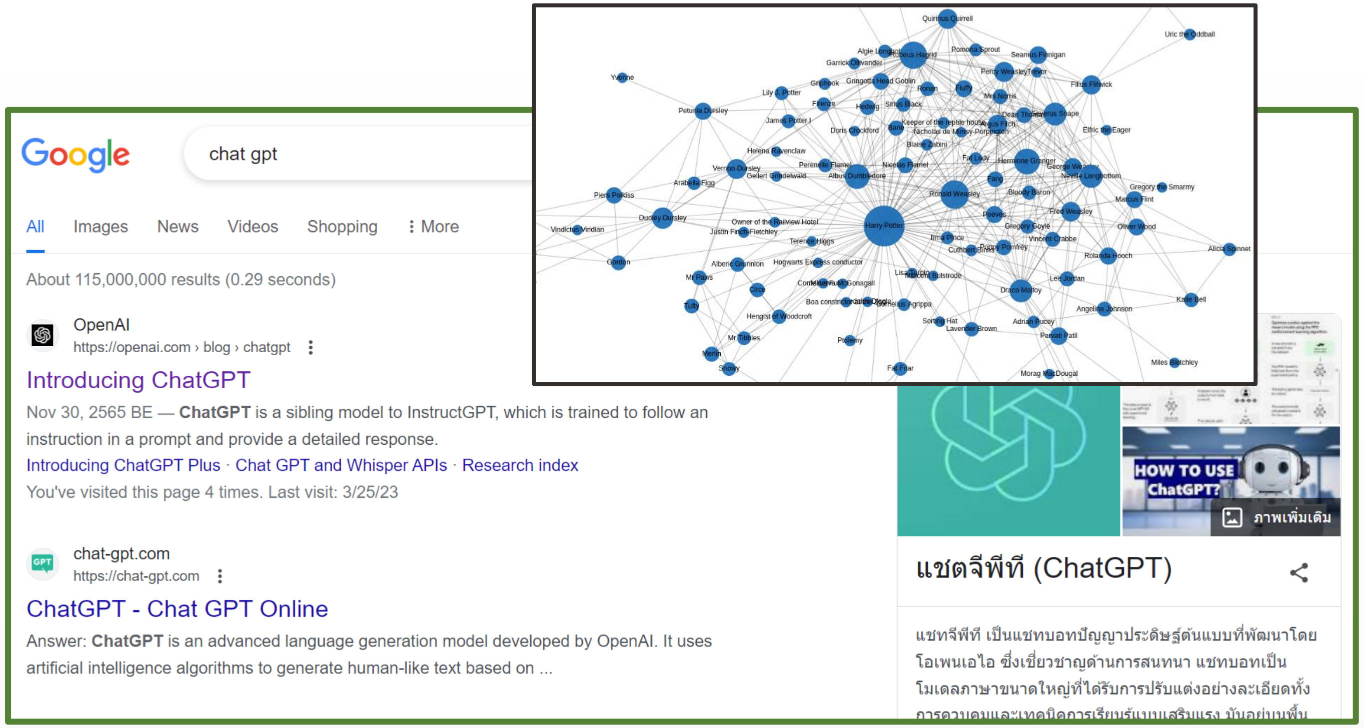


Figure 1: Question Answering System Architecture

Allam, A. M. N., & Haggag, M. H. (2012). The question answering systems: A survey. *International Journal of Research and Reviews in Information Sciences (IJRRIS)*, *2*(3).

Types of QA systems

Structured Knowledge Base



Google

chat gpt

All Images News Videos Shopping ⋮ More

About 115,000,000 results (0.29 seconds)

OpenAI
https://openai.com › blog › chatgpt ⋮

Introducing ChatGPT

Nov 30, 2565 BE — **ChatGPT** is a sibling model to InstructGPT, which is trained to follow an instruction in a prompt and provide a detailed response.

Introducing ChatGPT Plus · Chat GPT and Whisper APIs · Research index

You've visited this page 4 times. Last visit: 3/25/23

chat-gpt.com
https://chat-gpt.com ⋮

ChatGPT - Chat GPT Online

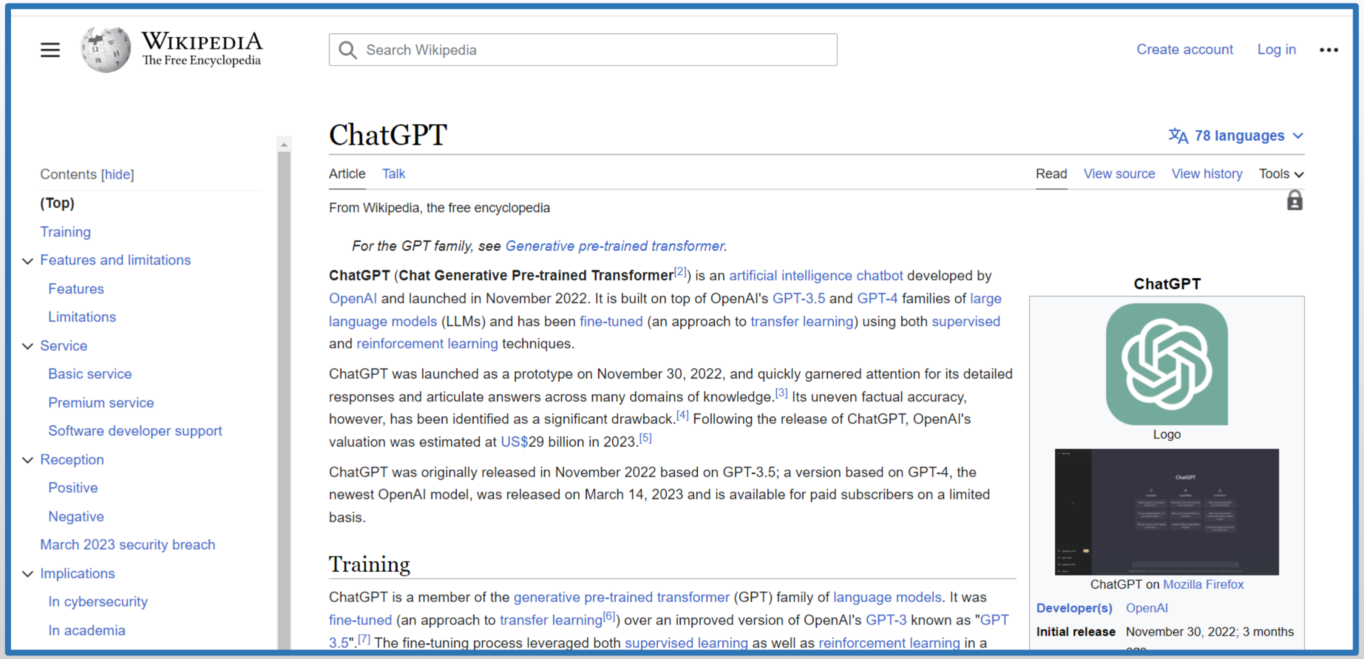
Answer: **ChatGPT** is an advanced language generation model developed by OpenAI. It uses artificial intelligence algorithms to generate human-like text based on ...

ภาพเพิ่มเติม

แชตจีพีที (ChatGPT)

แชทจีพีที เป็นแชทบอทปัญญาประดิษฐ์ต้นแบบที่พัฒนาโดยโอเพนเอไอ ซึ่งเชี่ยวชาญด้านการสนทนา แชทบอทเป็นโมเดลภาษาขนาดใหญ่ที่ได้รับการปรับแต่งอย่างละเอียดทั้งการควบคุมและเทคนิคการเรียนรู้แบบเสริมแรง มันอยู่บนพื้น

Unstructured Knowledge Base



Wikipedia
The Free Encyclopedia

Search Wikipedia

Create account Log in •••

Contents [hide]
(Top)
Training
Features and limitations
Features
Limitations
Service
Basic service
Premium service
Software developer support
Reception
Positive
Negative
March 2023 security breach
Implications
In cybersecurity
In academia

ChatGPT

78 languages

Article Talk Read View source View history Tools

From Wikipedia, the free encyclopedia

For the GPT family, see Generative pre-trained transformer.

ChatGPT (**Chat Generative Pre-trained Transformer**[2]) is an artificial intelligence chatbot developed by OpenAI and launched in November 2022. It is built on top of OpenAI's GPT-3.5 and GPT-4 families of large language models (LLMs) and has been fine-tuned (an approach to transfer learning) using both supervised and reinforcement learning techniques.

ChatGPT was launched as a prototype on November 30, 2022, and quickly garnered attention for its detailed responses and articulate answers across many domains of knowledge.[3] Its uneven factual accuracy, however, has been identified as a significant drawback.[4] Following the release of ChatGPT, OpenAI's valuation was estimated at US$29 billion in 2023.[5]

ChatGPT was originally released in November 2022 based on GPT-3.5; a version based on GPT-4, the newest OpenAI model, was released on March 14, 2023 and is available for paid subscribers on a limited basis.

Training

ChatGPT is a member of the generative pre-trained transformer (GPT) family of language models. It was fine-tuned (an approach to transfer learning[6]) over an improved version of OpenAI's GPT-3 known as "GPT 3.5".[7] The fine-tuning process leveraged both supervised learning as well as reinforcement learning in a

ChatGPT

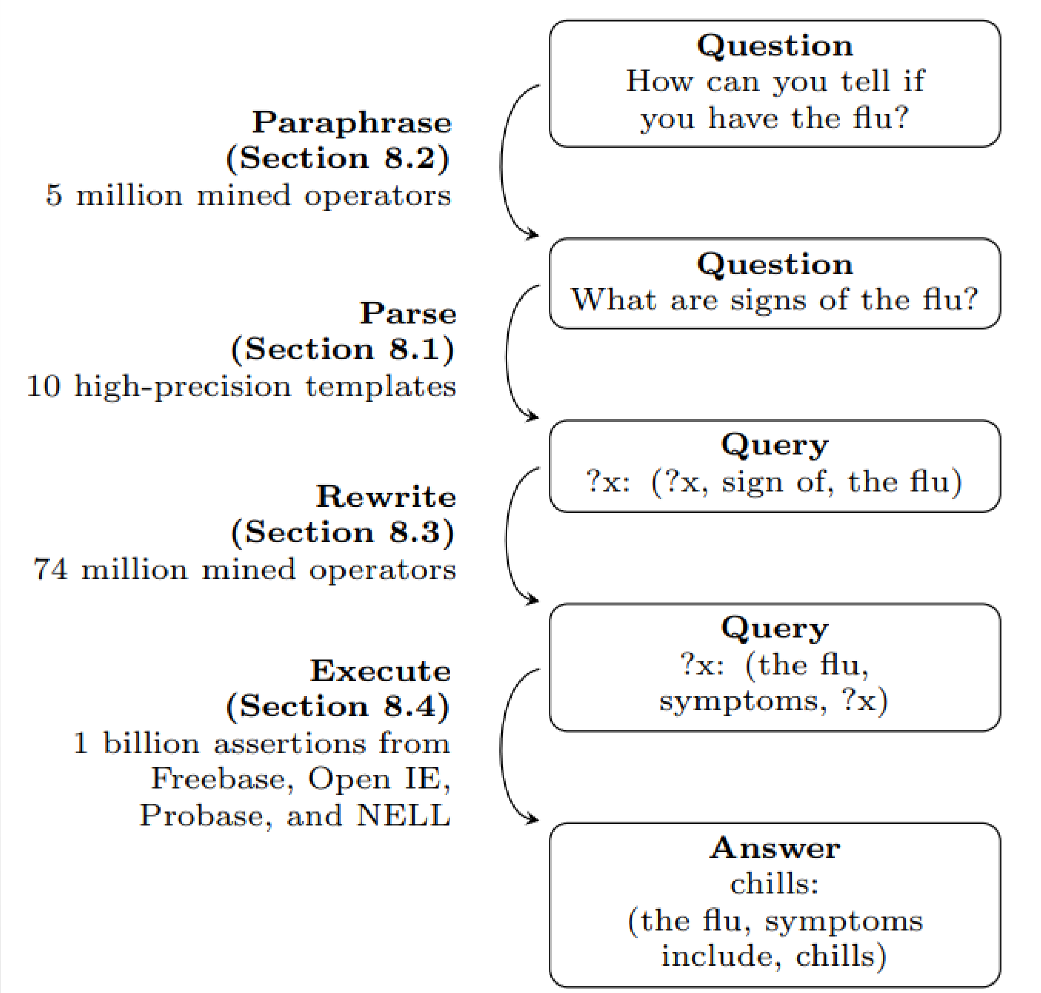
Logo

ChatGPT on Mozilla Firefox

Developer(s)	OpenAI
Initial release	November 30, 2022; 3 months

Example of Traditional Methods

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



Question
How can you tell if you have the flu?

Paraphrase
(Section 8.2)
5 million mined operators

Question
What are signs of the flu?

Parse
(Section 8.1)
10 high-precision templates

Query
?x: (?x, sign of, the flu)

Rewrite
(Section 8.3)
74 million mined operators

Query
?x: (the flu, symptoms, ?x)

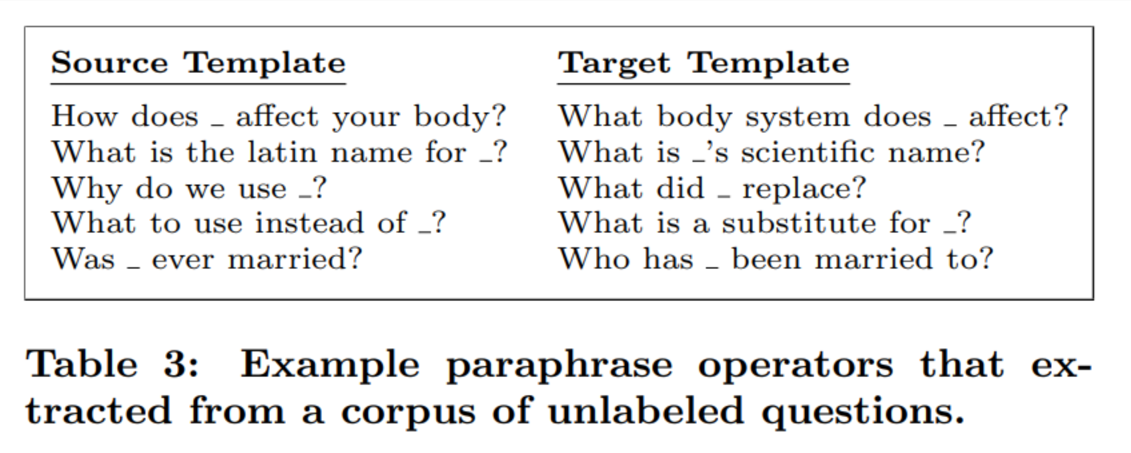
Execute
(Section 8.4)
1 billion assertions from Freebase, Open IE, Probase, and NELL

Answer
chills:
(the flu, symptoms include, chills)

Fader, A., Zettlemoyer, L., & Etzioni, O. (2014, August). Open question answering over curated and extracted knowledge bases. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 1156-1165).

Example of Traditional Methods

- 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)



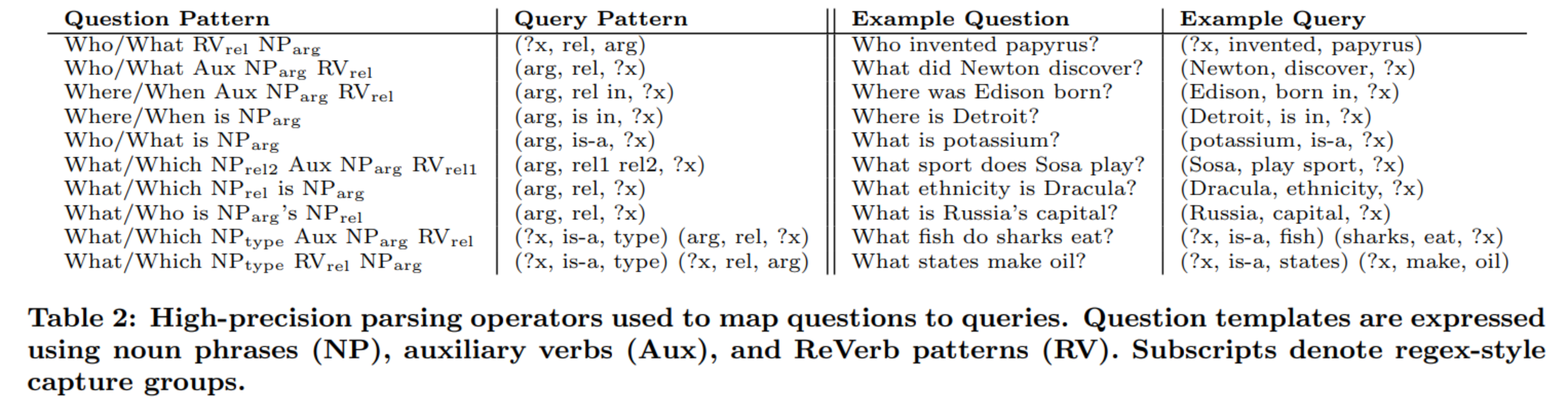
Source Template	Target Template
How does _ affect your body?	What body system does _ affect?
What is the latin name for _?	What is _'s scientific name?
Why do we use _?	What did _ replace?
What to use instead of _?	What is a substitute for _?
Was _ ever married?	Who has _ been married to?

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

Fader, A., Zettlemoyer, L., & Etzioni, O. (2014, August). Open question answering over curated and extracted knowledge bases. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 1156-1165).

Example of Traditional Methods

- 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



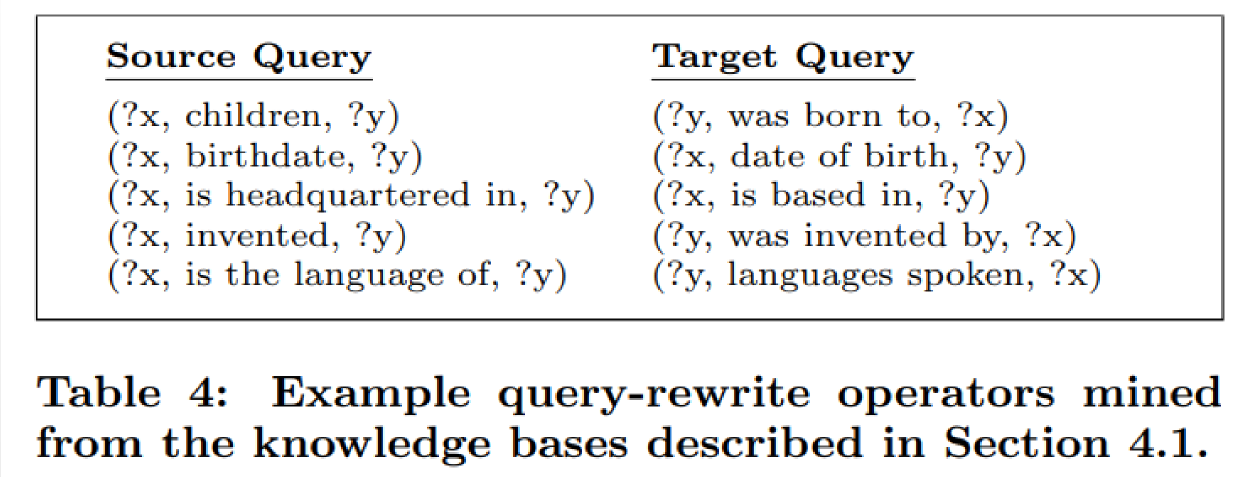
Question Pattern	Query Pattern	Example Question	Example Query
Who/What RV_{rel} NP_{arg}	(?x, rel, arg)	Who invented papyrus?	(?x, invented, papyrus)
Who/What Aux NP_{arg} RV_{rel}	(arg, rel, ?x)	What did Newton discover?	(Newton, discover, ?x)
Where/When Aux NP_{arg} RV_{rel}	(arg, rel in, ?x)	Where 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)

Table 2: High-precision parsing operators used to map questions to queries. Question templates are expressed using noun phrases (NP), auxiliary verbs (Aux), and ReVerb patterns (RV). Subscripts denote regex-style capture groups.

Fader, A., Zettlemoyer, L., & Etzioni, O. (2014, August). Open question answering over curated and extracted knowledge bases. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 1156-1165).

Example of Traditional Methods

- 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)



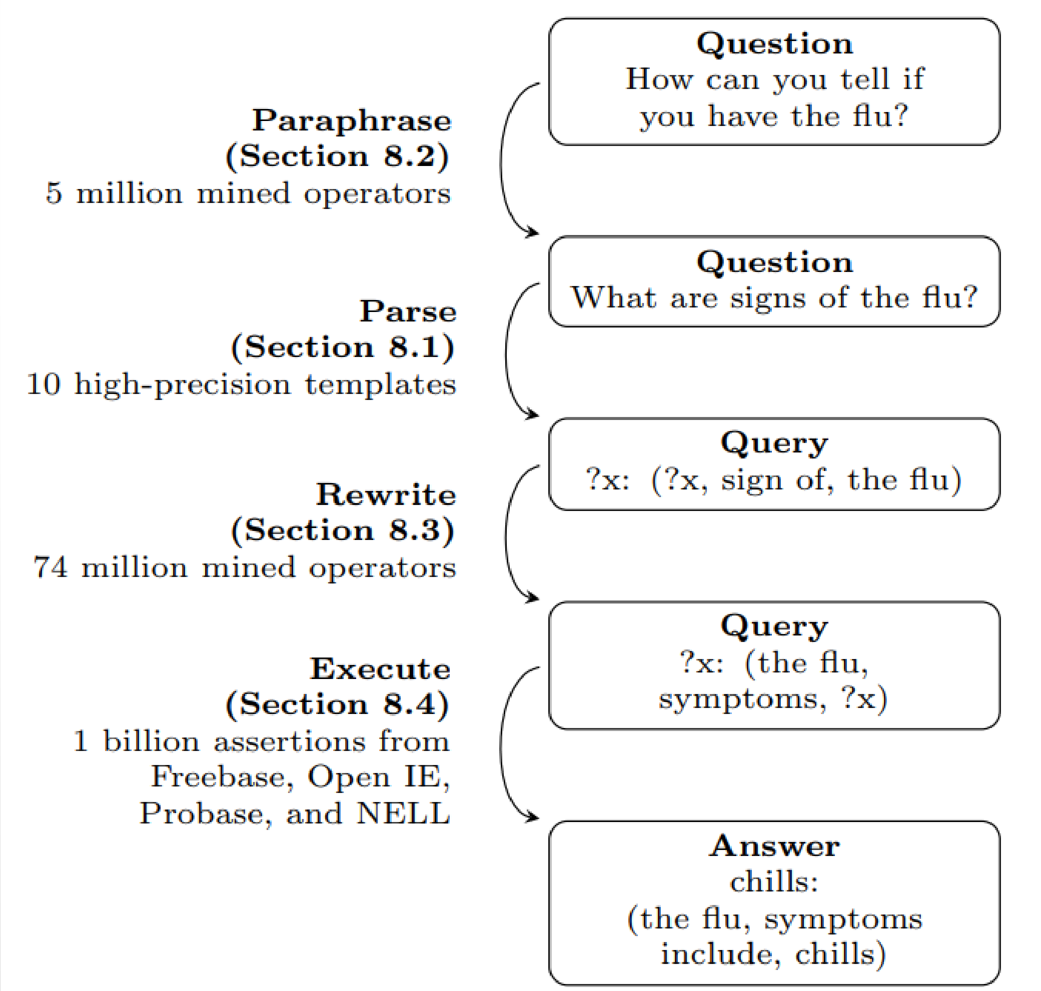
Source Query	Target Query
(?x, children, ?y)	(?y, was born to, ?x)
(?x, birthdate, ?y)	(?x, date of birth, ?y)
(?x, is headquartered in, ?y)	(?x, is based in, ?y)
(?x, invented, ?y)	(?y, was invented by, ?x)
(?x, is the language of, ?y)	(?y, languages spoken, ?x)

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

Fader, A., Zettlemoyer, L., & Etzioni, O. (2014, August). Open question answering over curated and extracted knowledge bases. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 1156-1165).

Example of Traditional Methods

- 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


Question
How can you tell if you have the flu?
Paraphrase
(Section 8.2)
5 million mined operators
Question
What are signs of the flu?
Parse
(Section 8.1)
10 high-precision templates
Query
?x: (?x, sign of, the flu)
Rewrite
(Section 8.3)
74 million mined operators
Query
?x: (the flu, symptoms, ?x)
Execute
(Section 8.4)
1 billion assertions from Freebase, Open IE, Probase, and NELL
Answer
chills:
(the flu, symptoms include, chills)

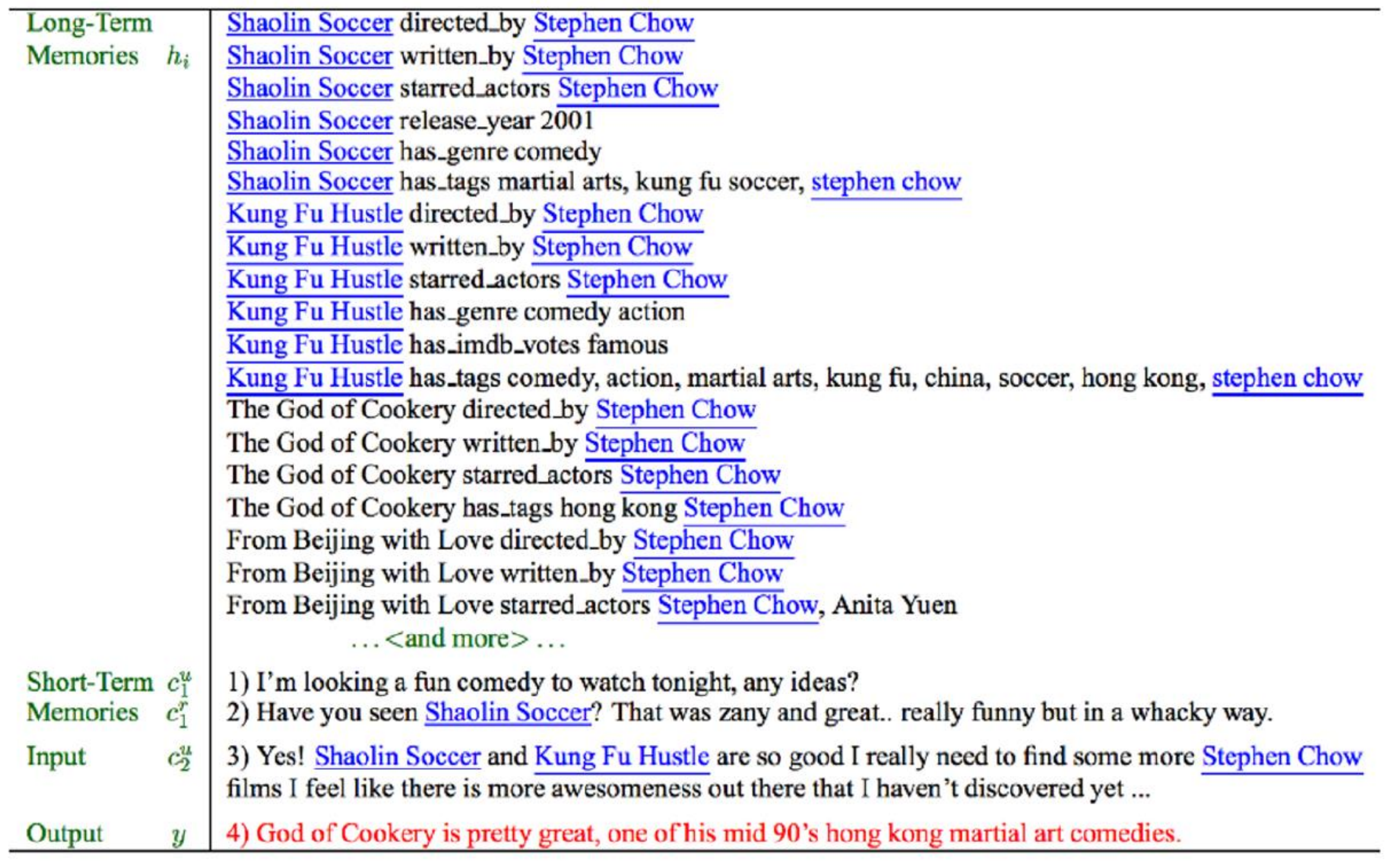
Fader, A., Zettlemoyer, L., & Etzioni, O. (2014, August). Open question answering over curated and extracted knowledge bases. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 1156-1165).

Limitation of Traditional Methods

- 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

Deep Learning and QA: Memory Neural Network

- Memory Neural 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.

Deep Learning and QA: Memory Neural Network

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

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

Fact

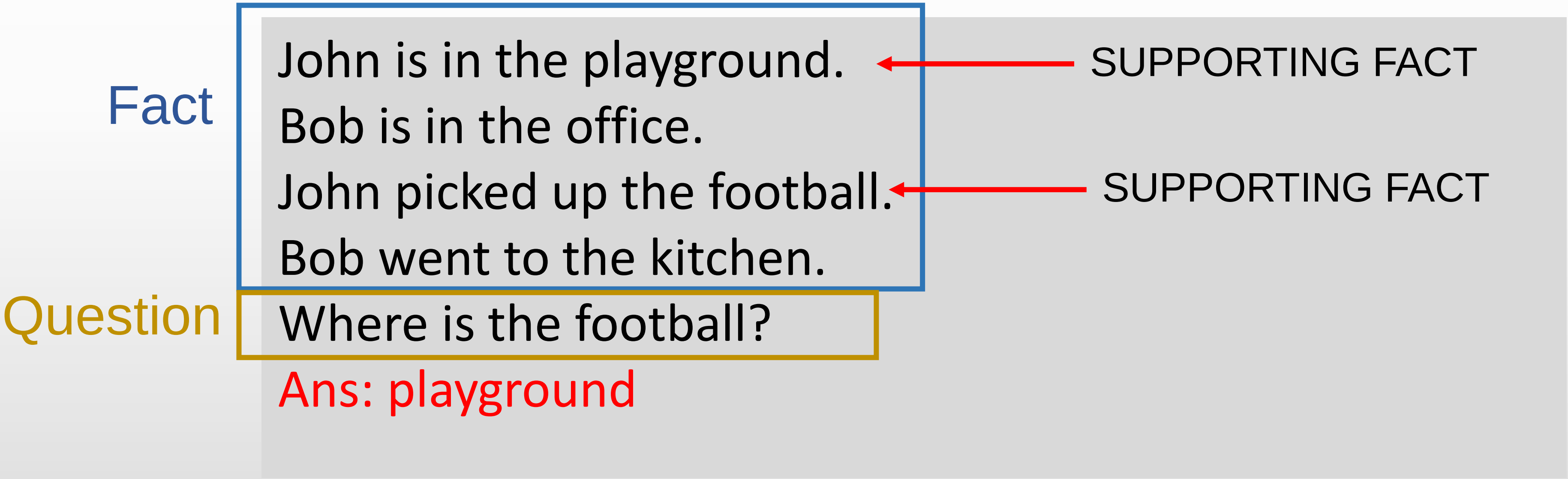
Where is the football?

Question

Ans: playground

Deep Learning and QA: Memory Neural Network

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



Fact

John is in the playground. ← SUPPORTING FACT

Bob is in the office.

John picked up the football. ← SUPPORTING FACT

Bob went to the kitchen.

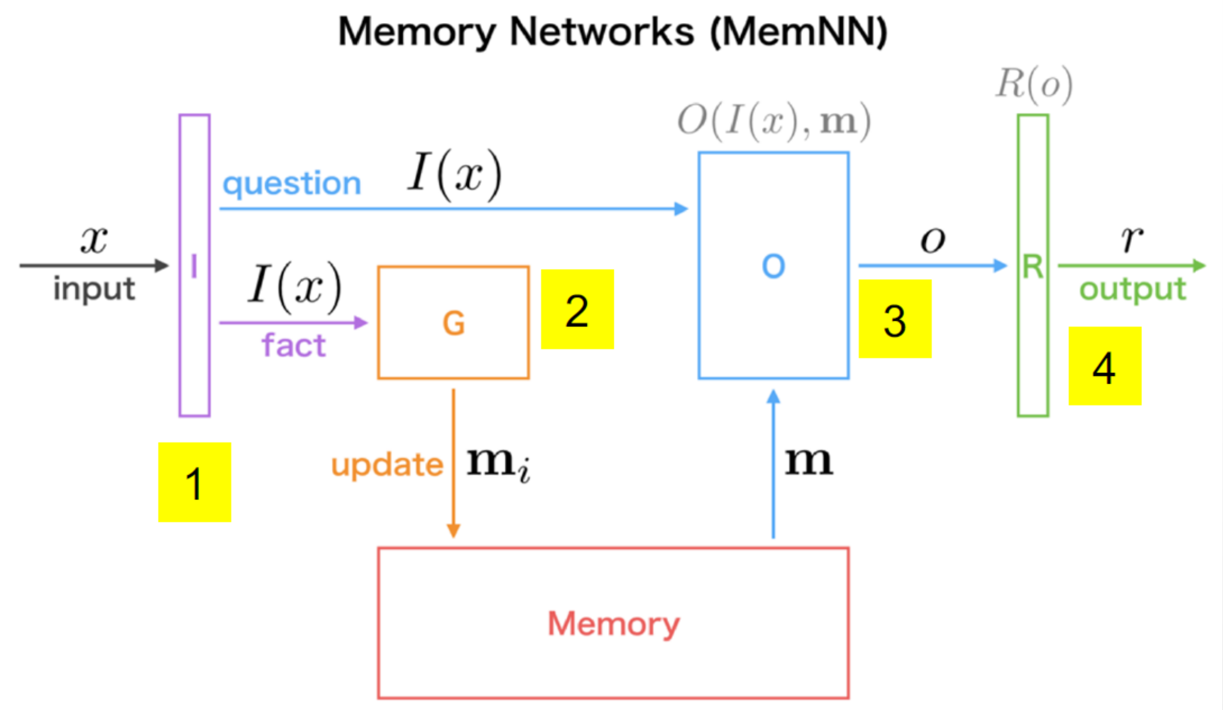
Question

Where is the football?

Ans: playground

Memory Network: What is it?

- **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

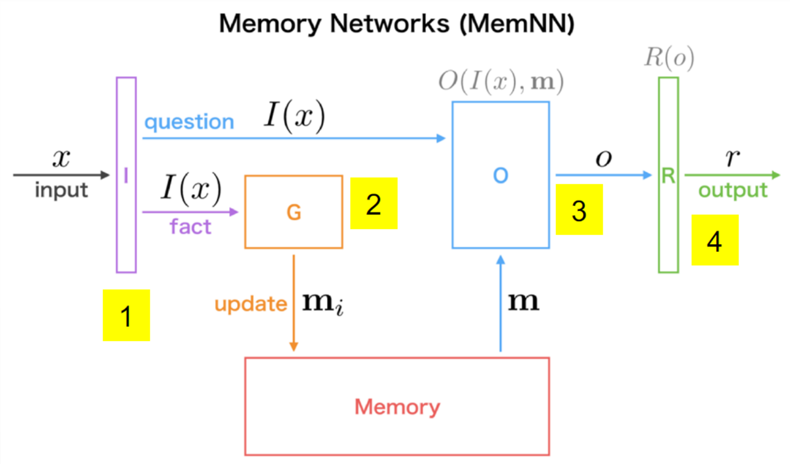


Weston, J., Chopra, S., & Bordes, A. (2014). Memory networks. *arXiv preprint arXiv:1410.3916*.

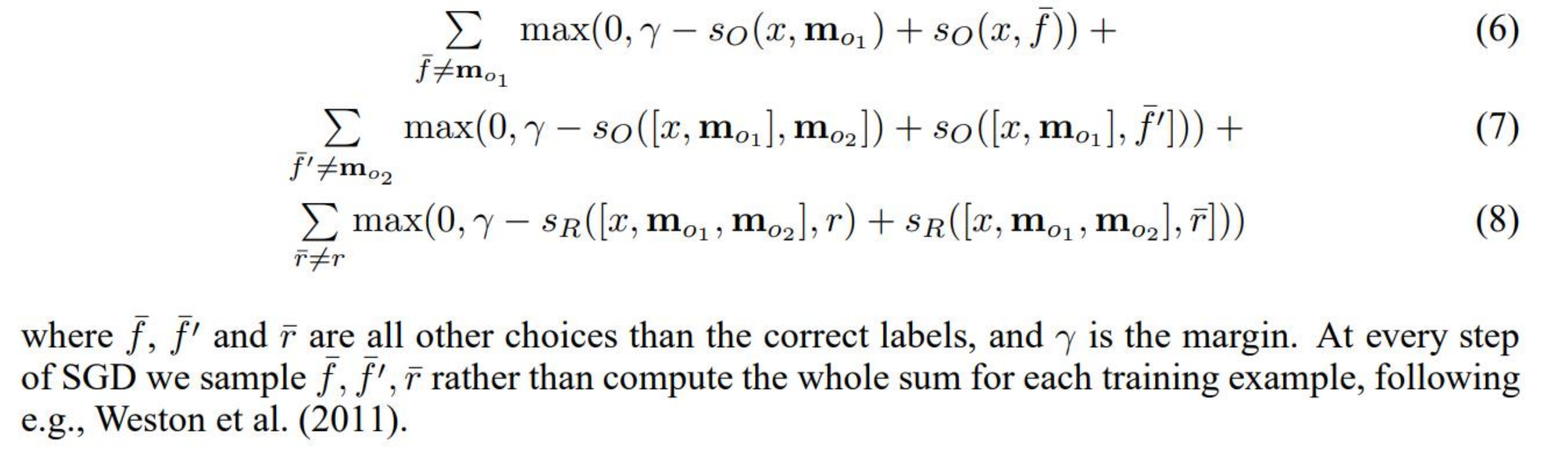
Memory Network: Train and Loss

- 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
  - Where x=inputs, y=target
  - 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).

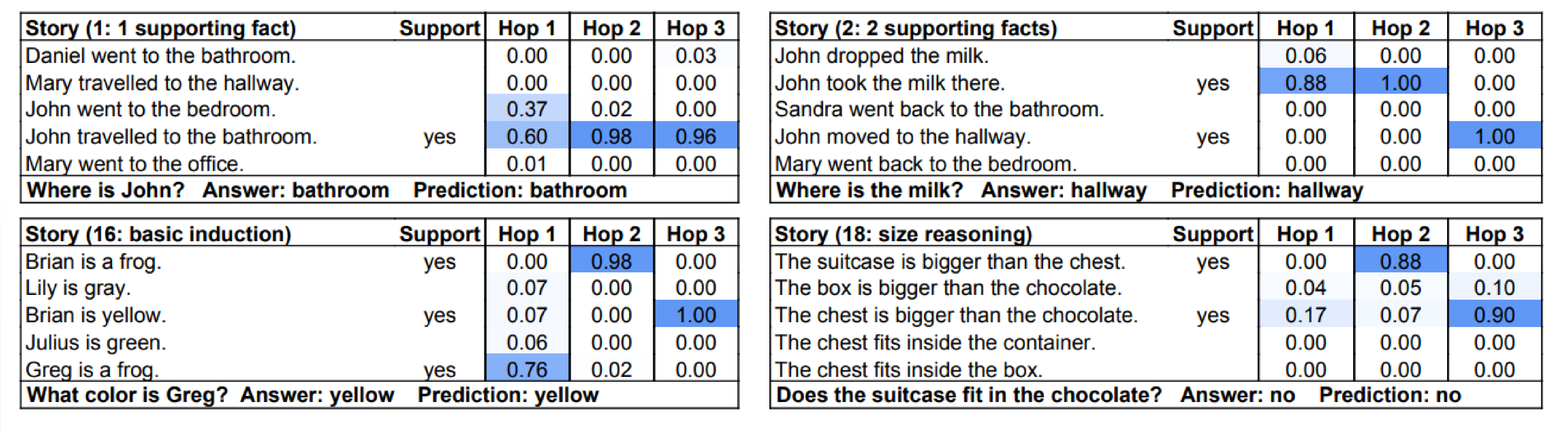
Weston, J., Chopra, S., & Bordes, A. (2014). Memory networks. *arXiv preprint arXiv:1410.3916*.

End-To-End Memory Network: Overview

- 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
- Soft Attention is when we calculate the context vector as a weighted sum of the encoder hidden states
- Hard Attention is when, instead of weighed average of all hidden states, we use attention scores to select a single hidden state.

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

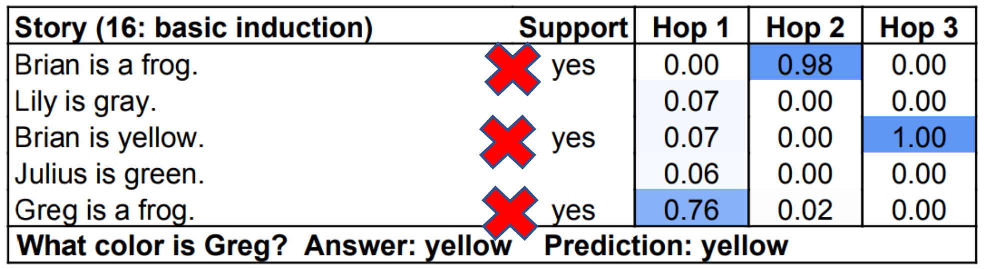
Story (18: size reasoning)	Support	Hop 1	Hop 2	Hop 3
The suitcase is bigger than the chest.	yes	0.00	0.88	0.00
The box is bigger than the chocolate.		0.04	0.05	0.10
The chest is bigger than the chocolate.	yes	0.17	0.07	0.90
The chest fits inside the container.		0.00	0.00	0.00
The chest fits inside the box.		0.00	0.00	0.00

Does the suitcase fit in the chocolate? Answer: no Prediction: no

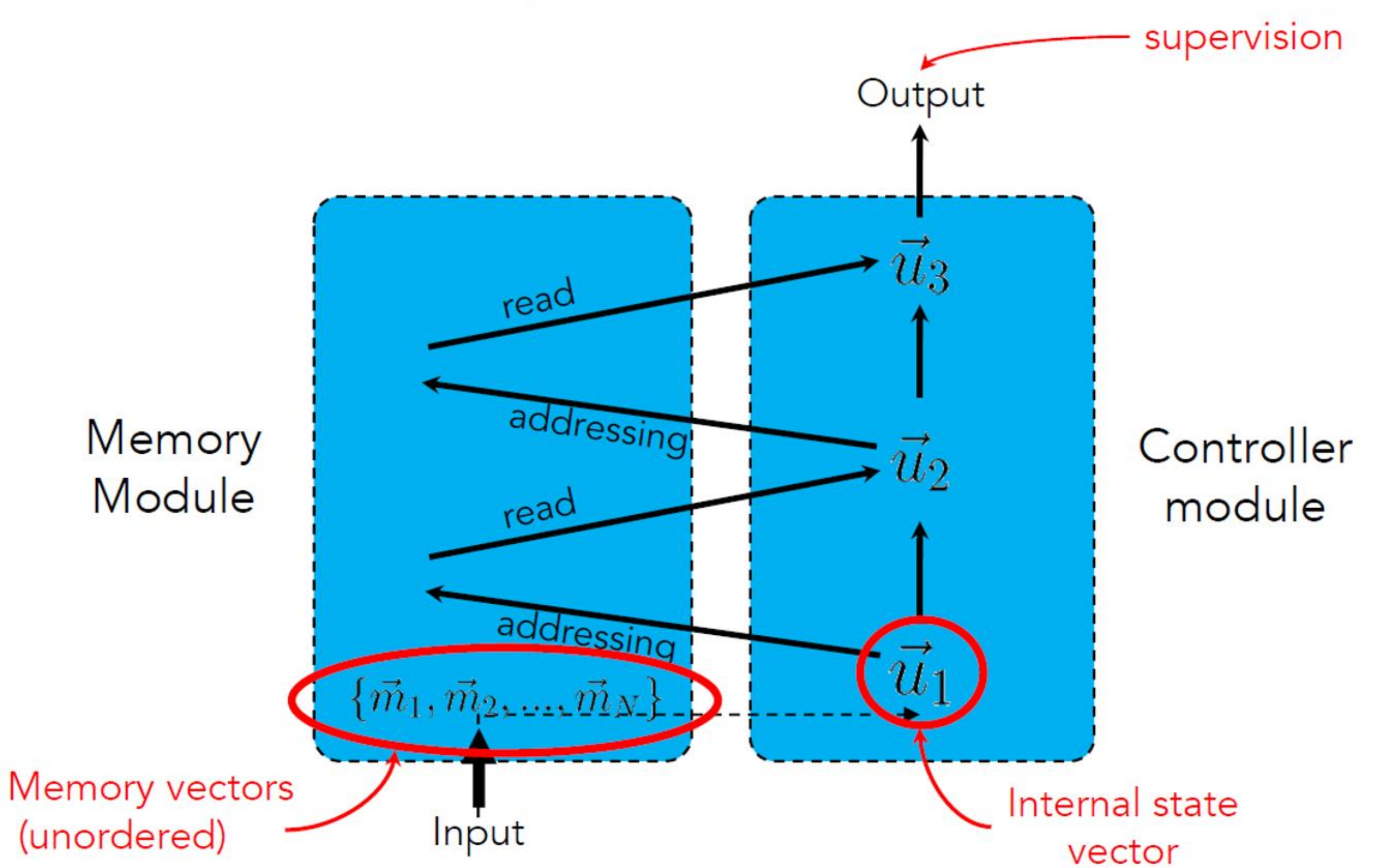
Sukhbaatar, S., Weston, J., & Fergus, R. (2015). End-to-end memory networks. *Advances in neural information processing systems*, *28*.

End-To-End Memory Network: Overview 2 hops (MemN2N)

Use soft Attention

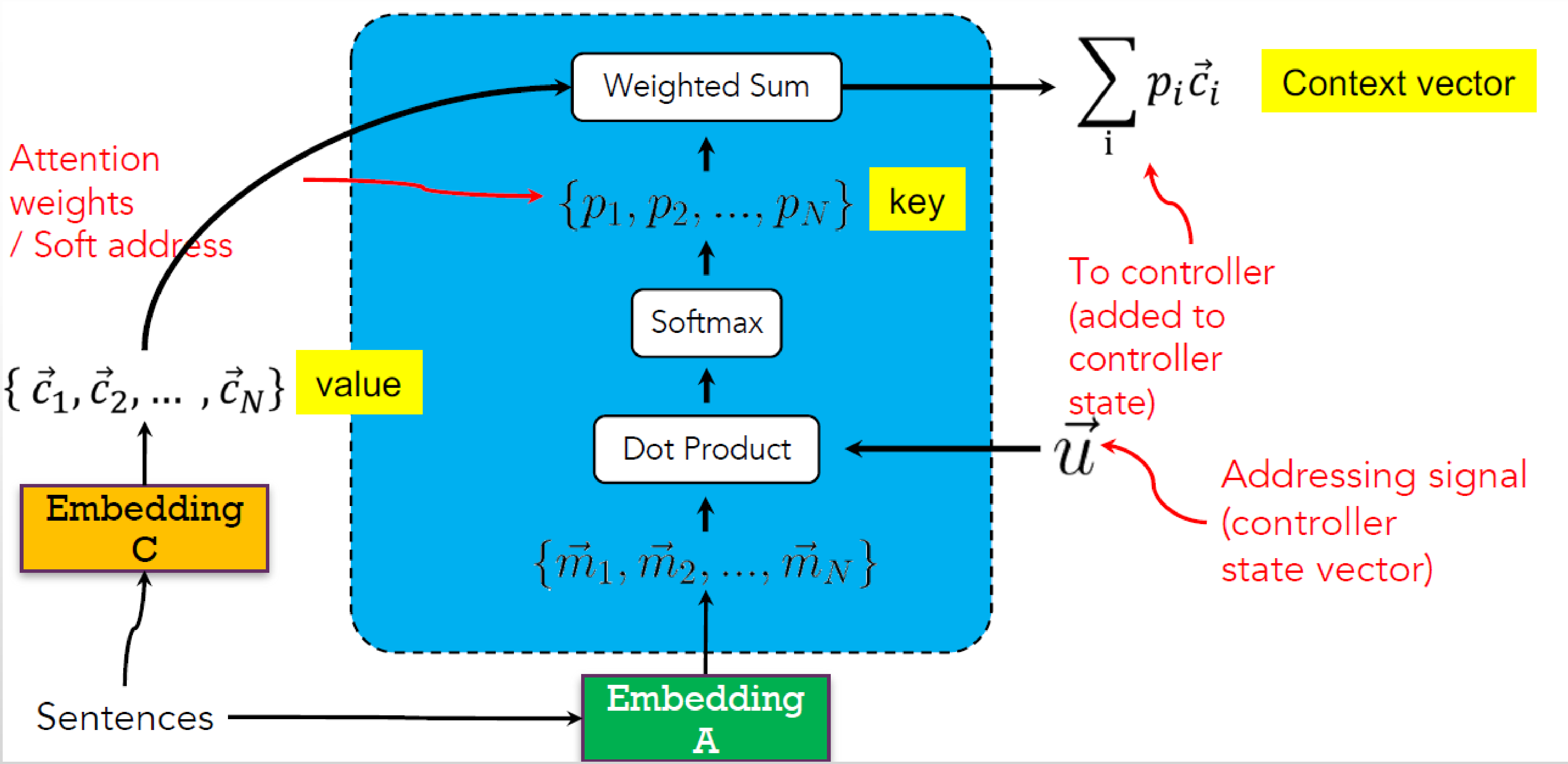


Story (16: basic induction)	Support	Hop 1	Hop 2	Hop 3
Brian is a frog.	✗ yes	0.00	0.98	0.00
Lily is gray.		0.07	0.00	0.00
Brian is yellow.	✗ yes	0.07	0.00	1.00
Julius is green.		0.06	0.00	0.00
Greg is a frog.	✗ yes	0.76	0.02	0.00
What color is Greg? Answer: yellow	**Prediction: yellow**			

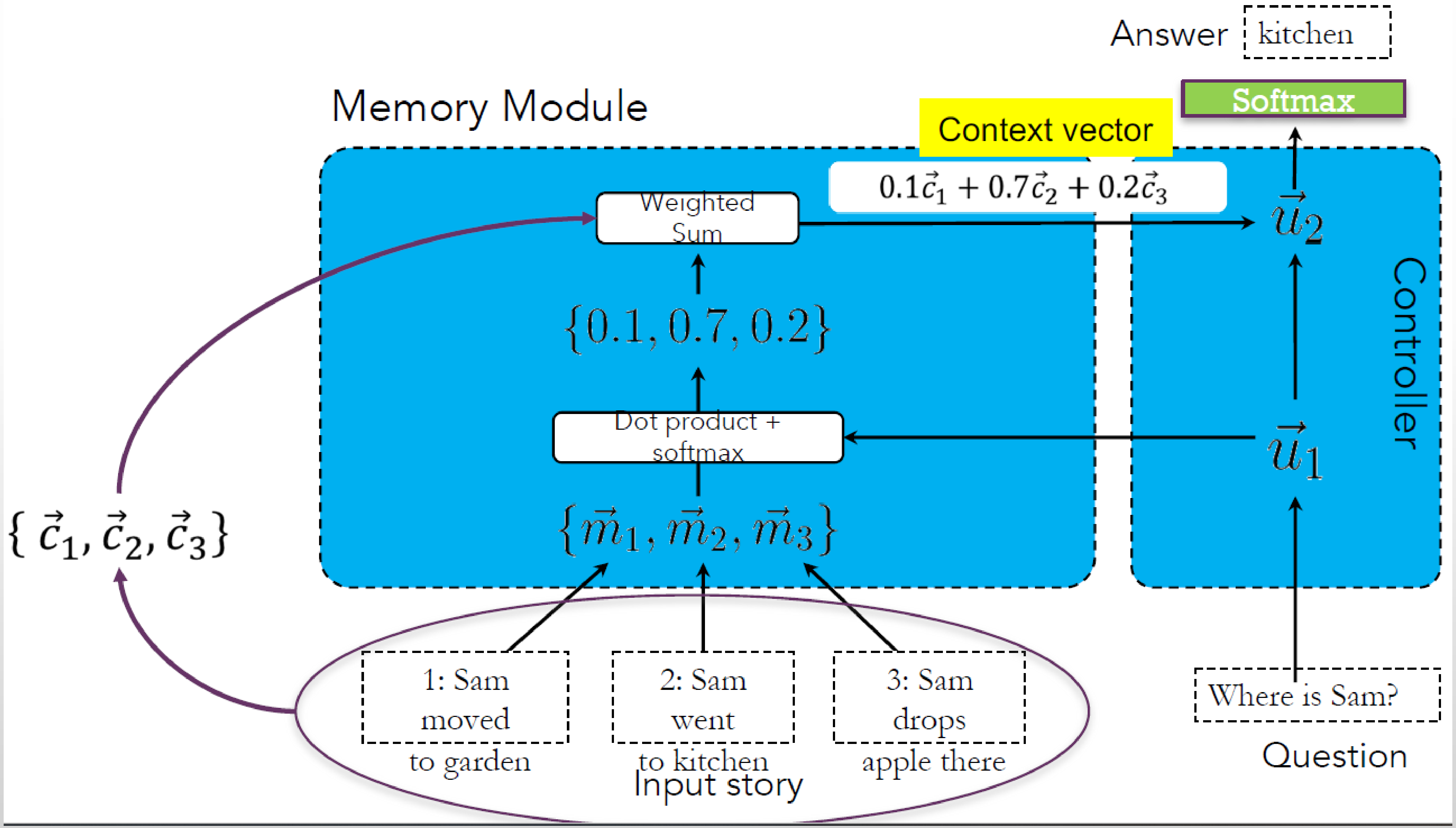


Sukhbaatar, S., Weston, J., & Fergus, R. (2015). End-to-end memory networks. *Advances in neural information processing systems*, *28*.

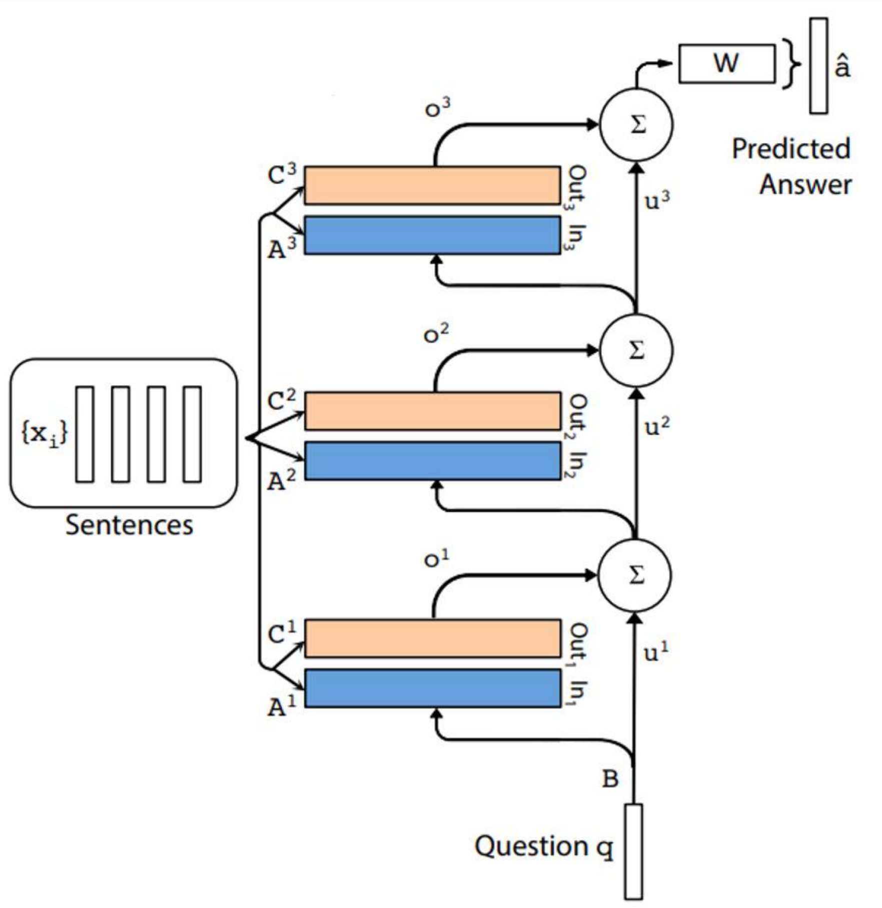
End-To-End Memory Network: Memory Module Key-Value Attention


Weighted Sum
$\sum_i p_i \vec{c}_i$
Context vector
Attention weights / Soft address
$\{p_1, p_2, \ldots, p_N\}$
key
Softmax
To controller (added to controller state)
$\{\vec{c}_1, \vec{c}_2, \ldots, \vec{c}_N\}$
value
Dot Product
$\vec{u}$
Addressing signal (controller state vector)
Embedding C
$\{\vec{m}_1, \vec{m}_2, \ldots, \vec{m}_N\}$
Sentences
Embedding A

End-To-End Memory Network: Example



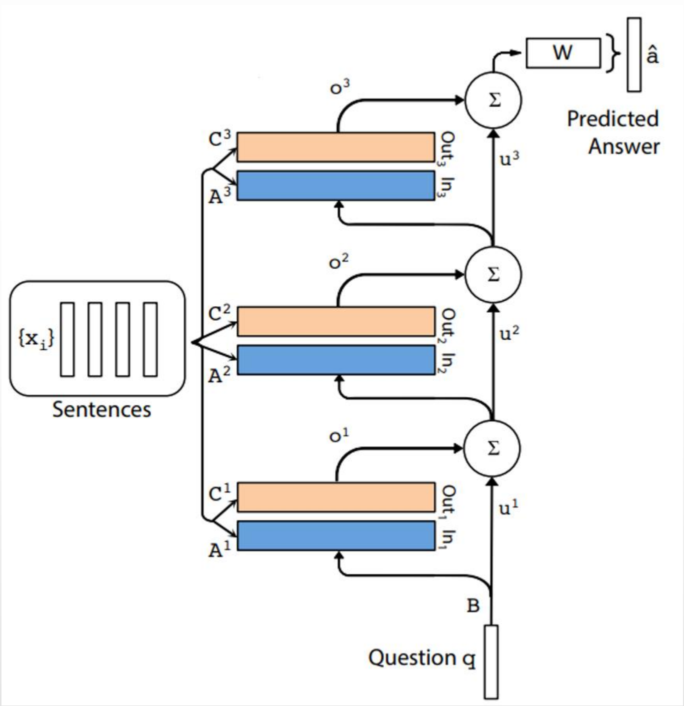
End-To-End Memory Network: Multiple Hops Reasoning (3 hops)



Sukhbaatar, S., Weston, J., & Fergus, R. (2015). End-to-end memory networks. *Advances in neural information processing systems*, *28*.

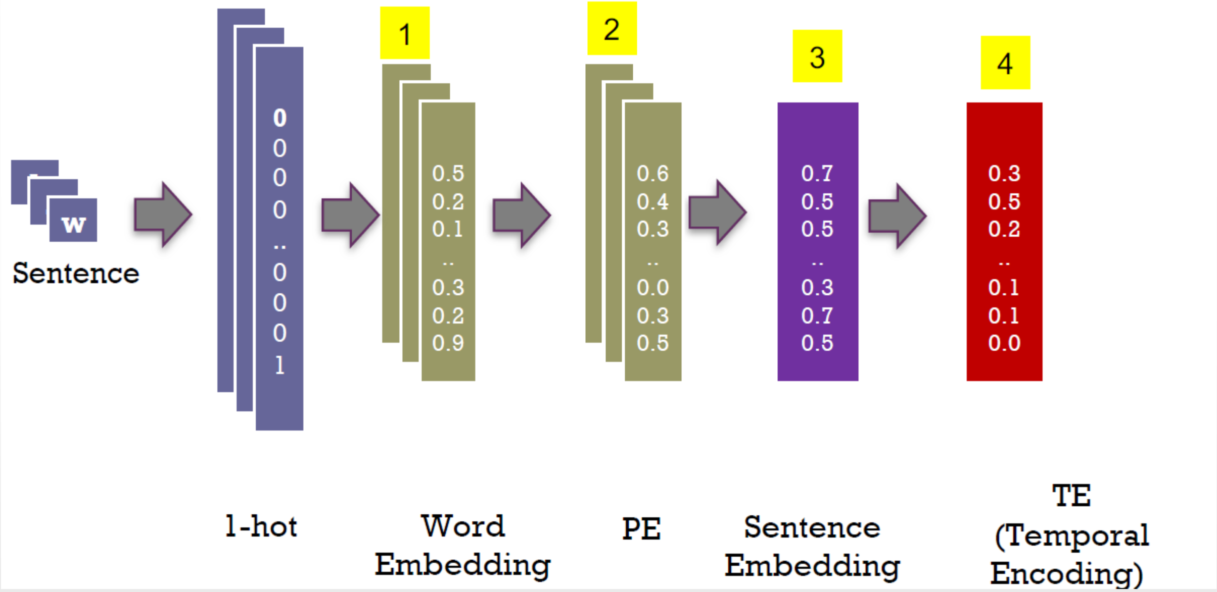
End-To-End Memory Network: Multiple Hops Reasoning

- There are two ways to do the multiple hops reasoning
- 1) Adjacent
  - The query representation (u) is updated every hop (sum): $u_{k+1} = u_k + o_k$
- 2) Layer-wise (RNN-like)
  - The query representation (u) is updated every hop with H linear mapping (dense): $u_{k+1} = Hu_k + o_k$



End-To-End Memory Network: Memory Vector

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

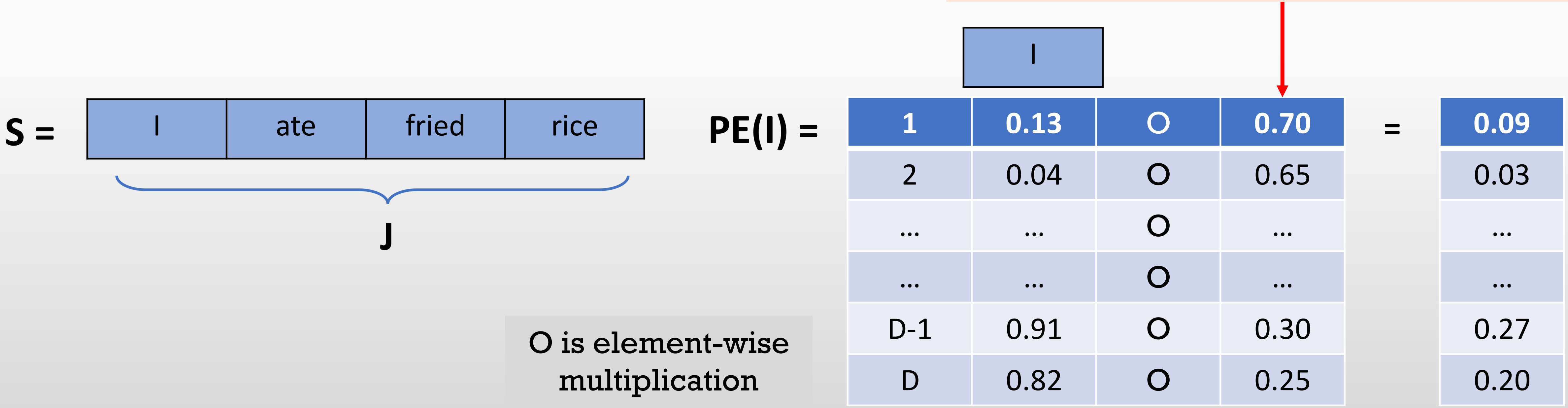


End-To-End Memory Network: Positional Encoding

j=positional index
d=dimensional index

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

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



S = | I | ate | fried | rice |

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	0.016	0.31

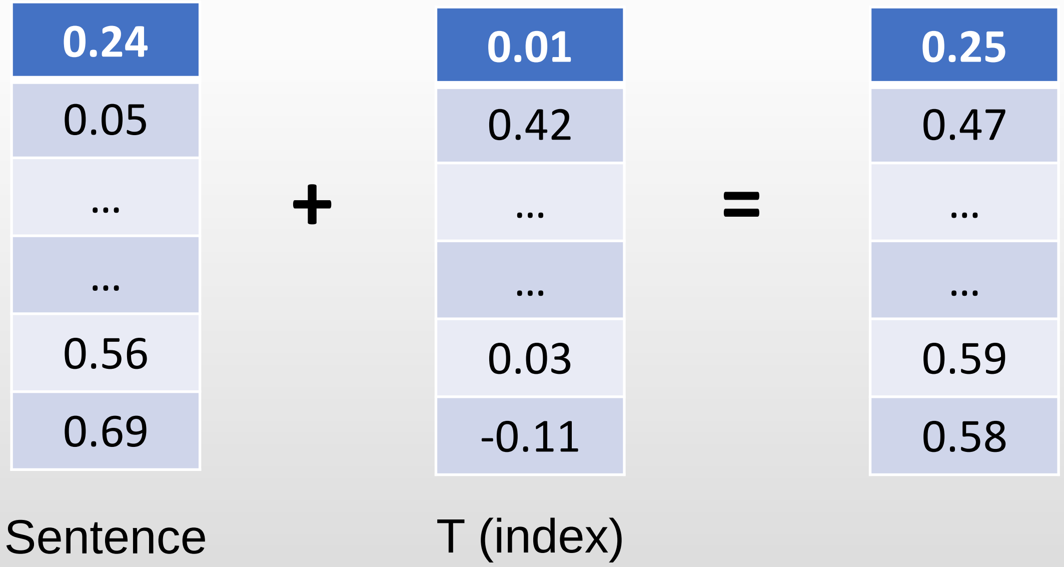
Σ

=

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



0.24
0.05
…
…
0.56
0.69

+

0.01
0.42
…
…
0.03
-0.11

=

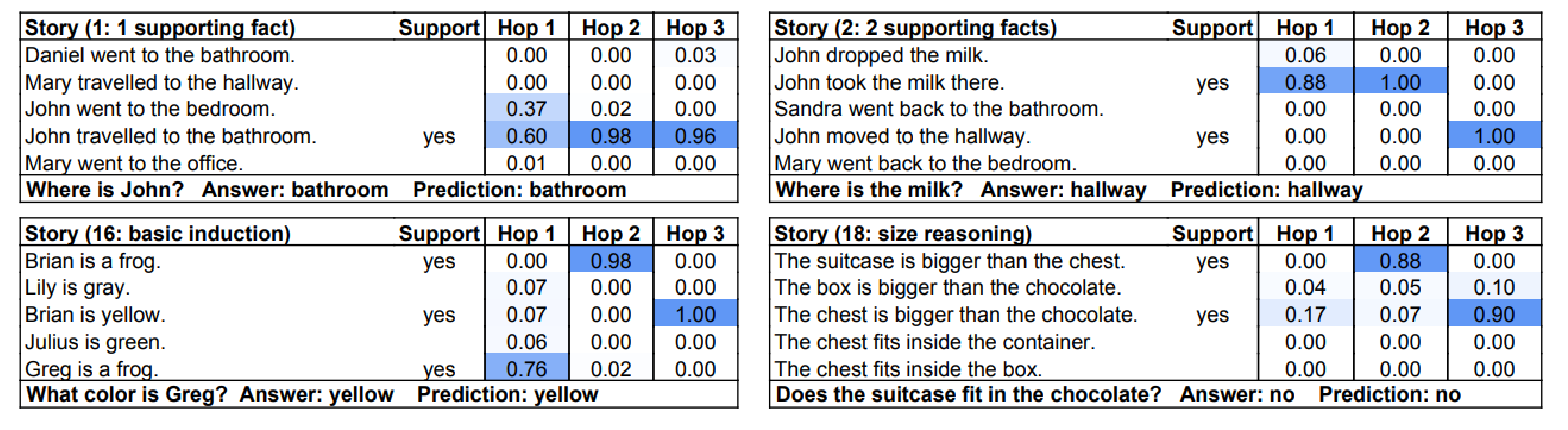
0.25
0.47
…
…
0.59
0.58

Sentence

T (index)

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				

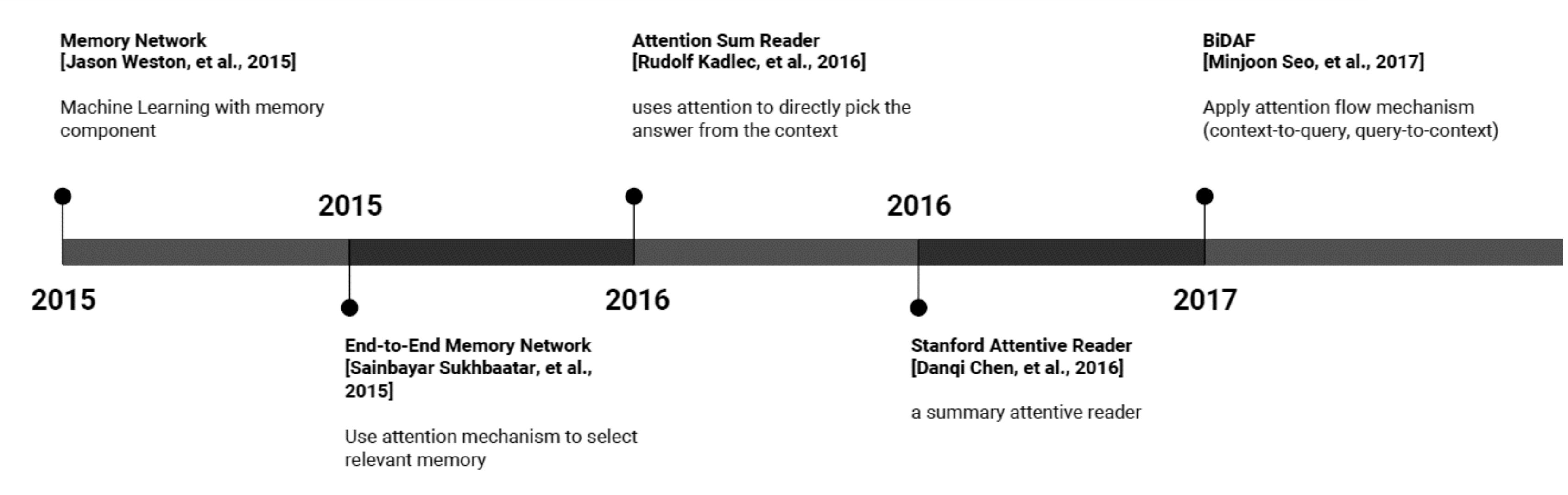
Story (2: 2 supporting facts)	Support	Hop 1	Hop 2	Hop 3
John dropped the milk.		0.06	0.00	0.00
John took the milk there.	yes	0.88	1.00	0.00
Sandra went back to the bathroom.		0.00	0.00	0.00
John moved to the hallway.	yes	0.00	0.00	1.00
Mary went back to the bedroom.		0.00	0.00	0.00
Where is the milk?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

Sukhbaatar, S., Weston, J., & Fergus, R. (2015). End-to-end memory networks. *Advances in neural information processing systems*, *28*.

Recent Deep QA models



Memory Network
[Jason Weston, et al., 2015]

Machine Learning with memory component

End-to-End Memory Network
[Sainbayar Sukhbaatar, et al., 2015]

Use attention mechanism to select relevant memory

Attention Sum Reader
[Rudolf Kadlec, et al., 2016]

uses attention to directly pick the answer from the context

Stanford Attentive Reader
[Danqi Chen, et al., 2016]

a summary attentive reader

BiDAF
[Minjoon Seo, et al., 2017]

Apply attention flow mechanism (context-to-query, query-to-context)

- 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

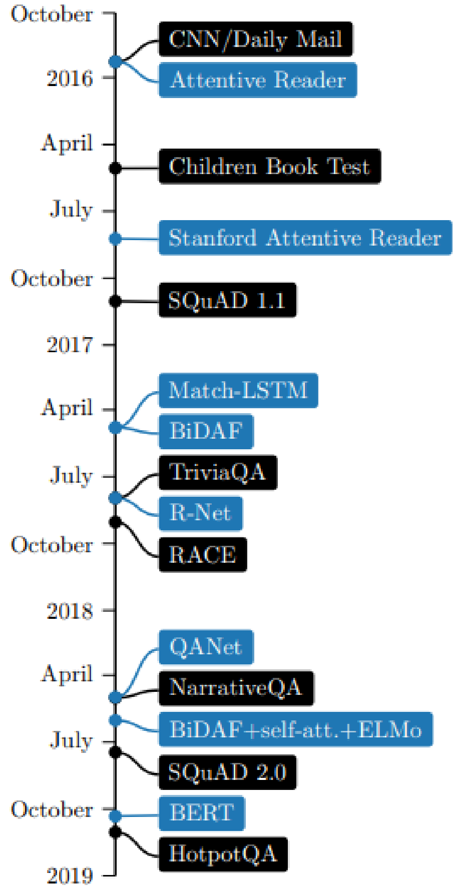


Figure 2.2: The recent development of datasets (black) and models (blue) in neural reading comprehension. For the timeline, we use the date that the corresponding papers were published, except BERT (Devlin et al., 2018).

https://stacks.stanford.edu/file/druid:gd576xb1833/thesis-augmented.pdf

Demo: Text generation

https://drive.google.com/file/d/11p2euvE5l2iMwKU5fIa40s0ur_05inVk/view?usp=share_link

Demo: Neural Machine Translation with Attention (Additive Attention)

https://drive.google.com/file/d/1RvyeQWDca99CO4WEddKYqa0CpkSYq-L6/view?usp=share_link

Demo: QA (AllenNLP)

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