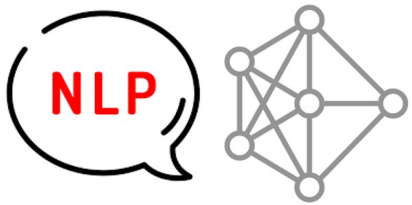
CHULA ENGINEERING
Foundation toward Innovation
COMPUTER



Attention Mechanisms

2110531: Data Science & Data Engineering Tools

(Based on 2110572: Natural Language Processing Systems)

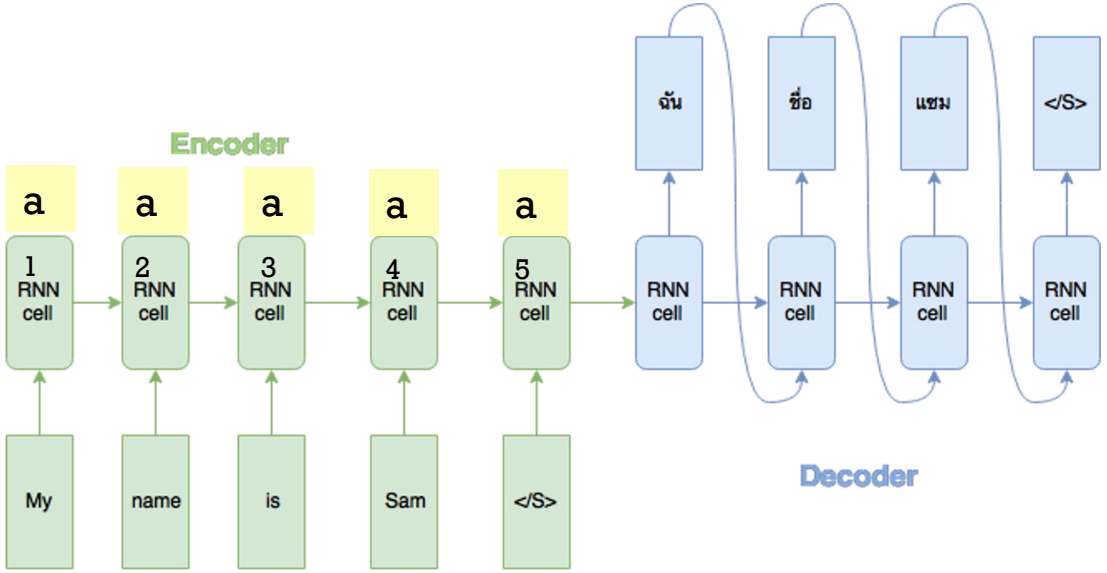
Peerapon Vateekul & Ekapol Chuangsuwanich
Department of Computer Engineering,
Faculty of Engineering, Chulalongkorn University

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?

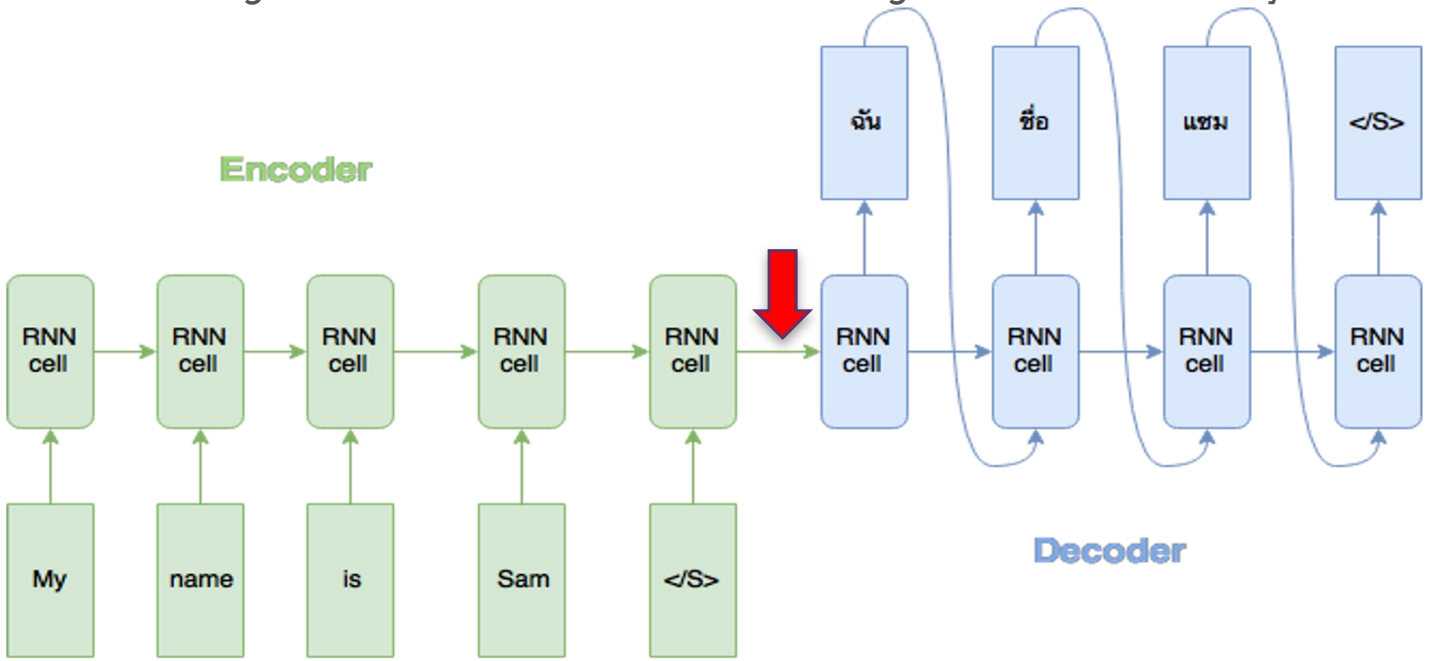


Machine Translation Problem: English to Thai

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

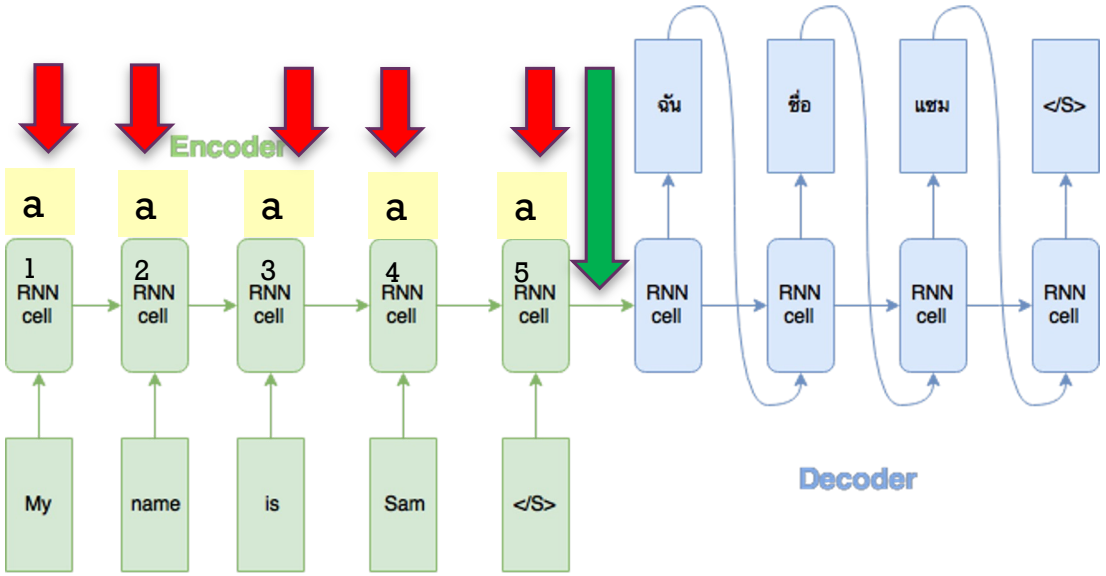
Machine Translation Problem: English to Thai

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.

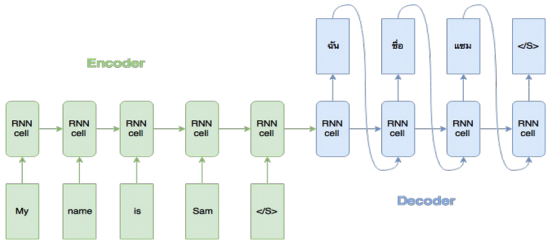


Machine Translation Problem: English to Thai

Graphical Example: English-to-Thai machine translation

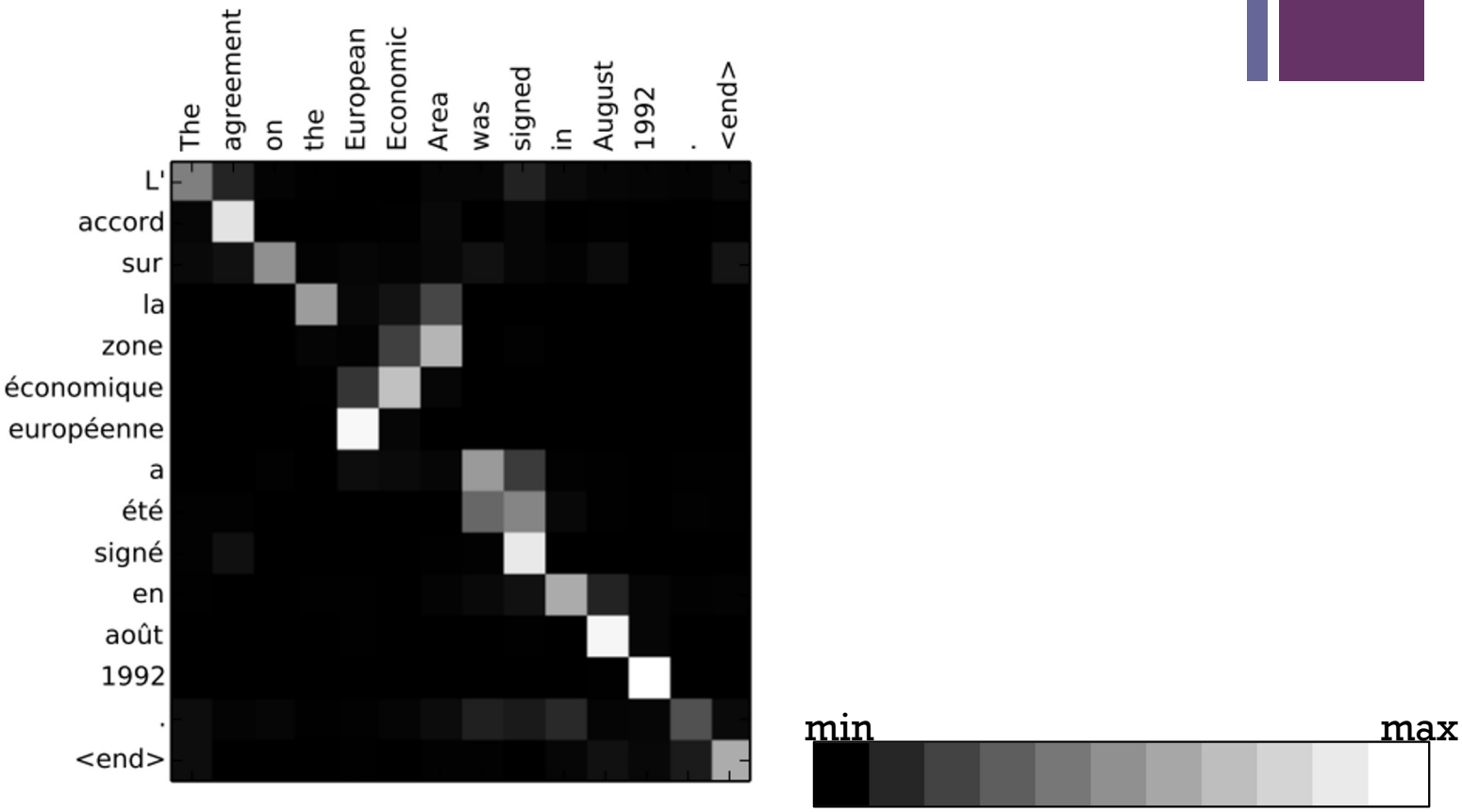
- This is a rough estimate of what might occur for English-to-Thai translation

	My	name	is	Sam	encoder
ฉัน					
ชื่อ					
แซม					
decoder					



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).

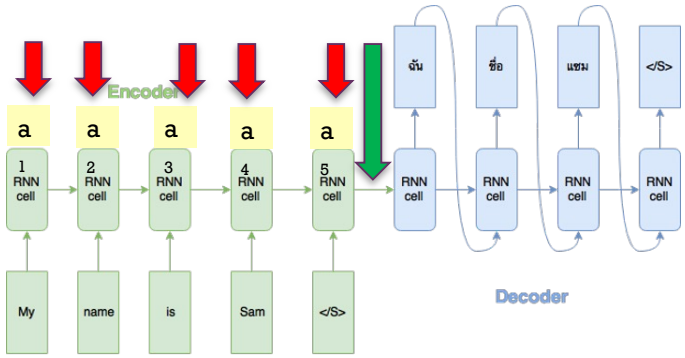
Attention Mechanism: Recap Basic Idea

- 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)

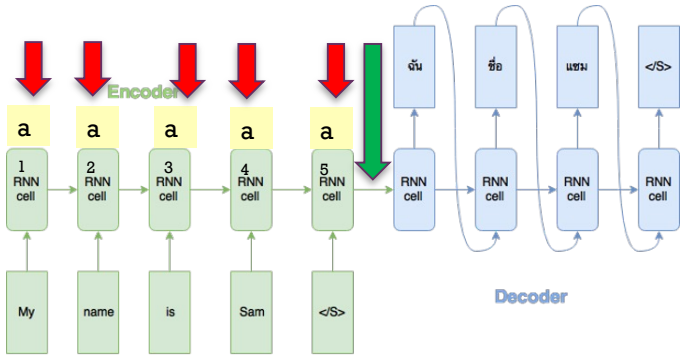
$$\mathbf{c}_i = \sum_j a_{ij}\mathbf{h}_j$$

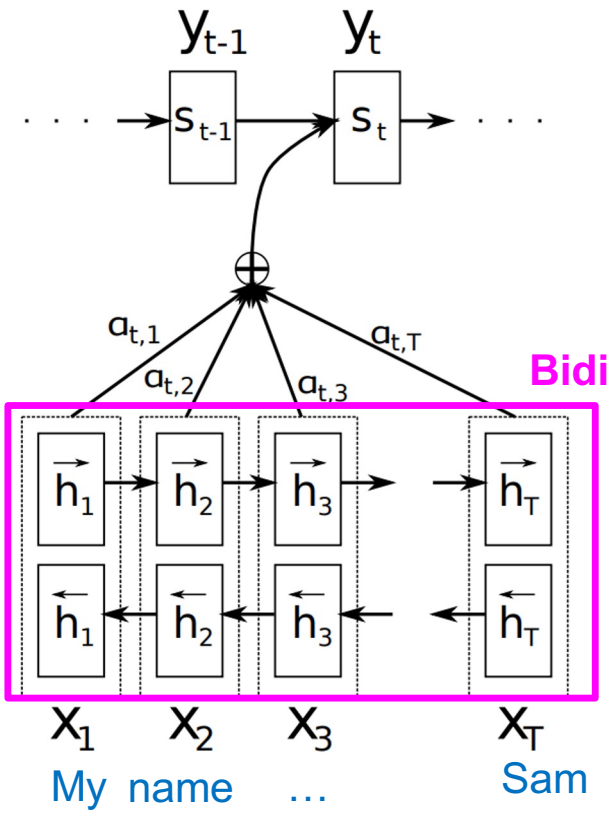
j = each encoder's input
i = each decoder's input

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



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





Decoder

Bidirectional LSTM

Encoder

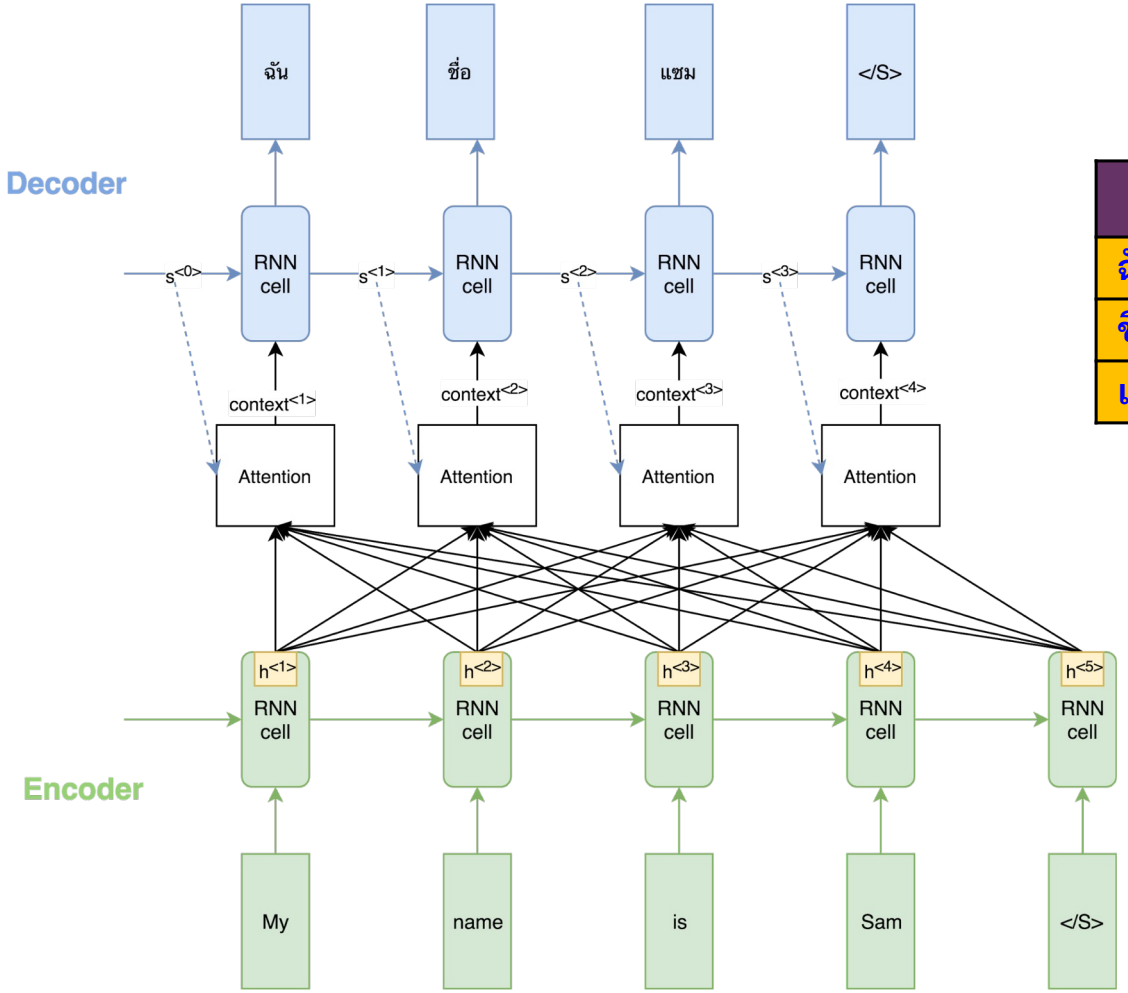
Figure 1: The graphical illustration of the proposed model trying to generate the t-th target word y_t given a source sentence $(x_1, x_2, \ldots, x_T)$.

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



	My	name	is	Sam
ฉัน				
ชื่อ				
แซม				

Attention Mechanism (1): C_i

We want to calculate a context vector ***c*** based on hidden states $\mathbf{h}_0 \ldots \mathbf{h}_j$ that can be used with the previous state $\mathbf{s}_{i-1}$ 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 $\mathbf{a}_i$.

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

context vector

attention score

encoder state at index j

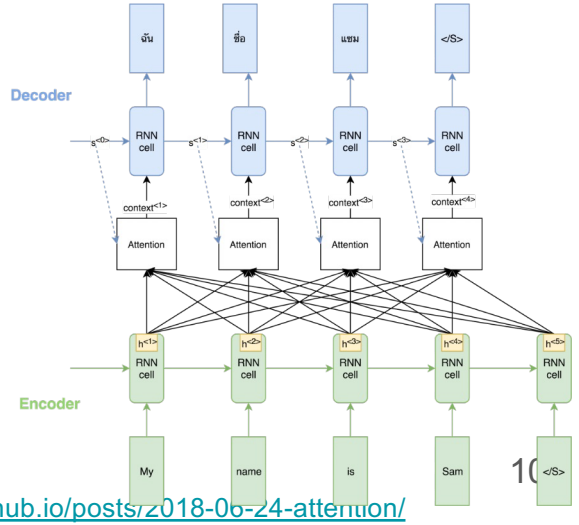
i = decoder index
j = encoder index

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

previous hidden state/ **decoder** state

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

encoder state at index j



Reference: https://lilianweng.github.io/posts/2018-06-24-attention/

Attention Mechanism (2): f_{att}

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

context vector

attention score

encoder state at index j

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

previous hidden state/ **decoder** state

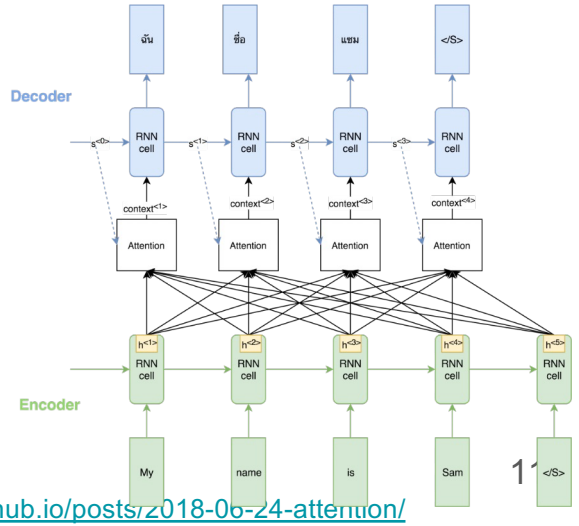
attention score(weight vector) of encoder state at index j

encoder state at index j

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}.

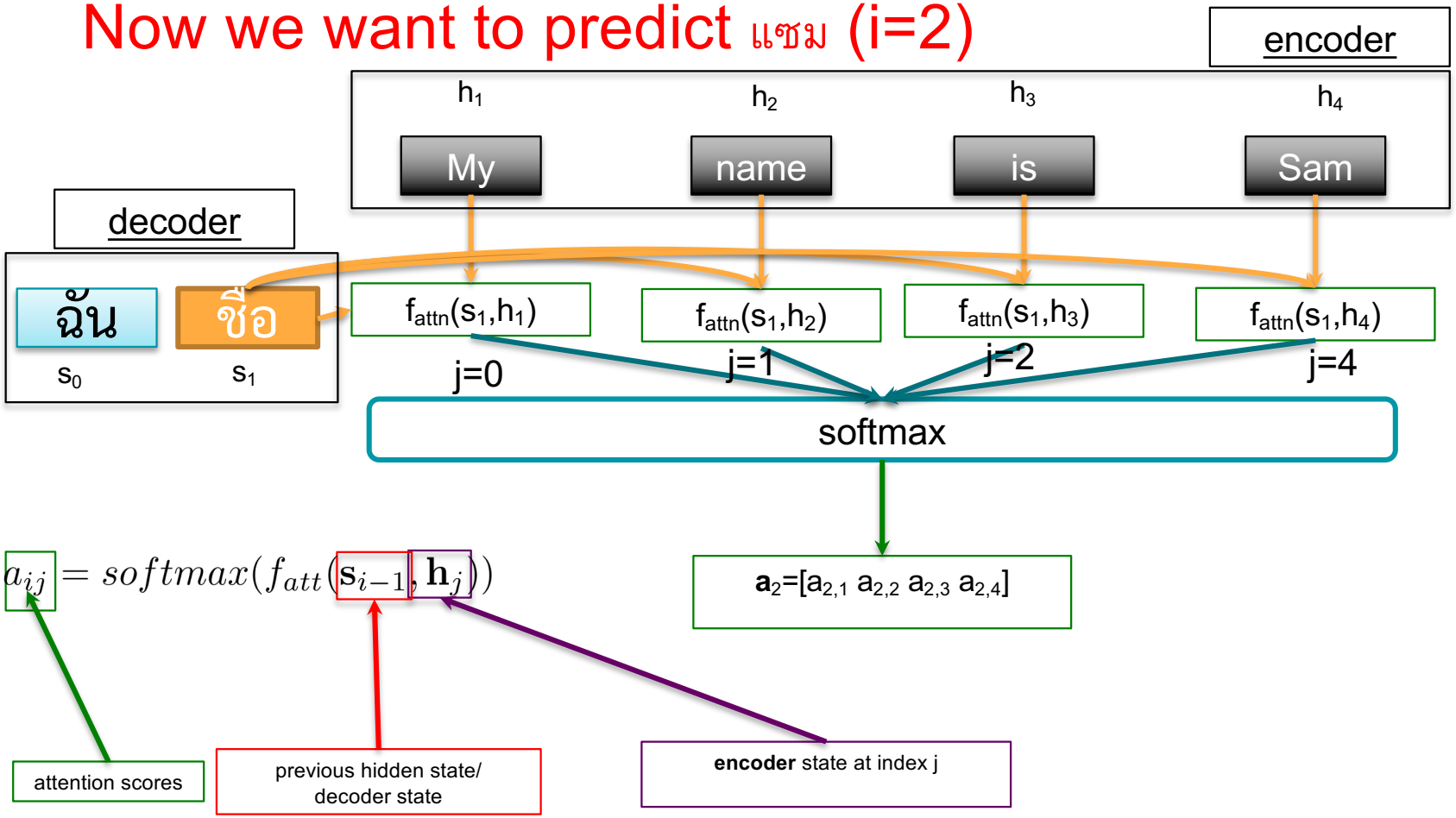
i = decoder index
j = encoder index



Reference: https://lilianweng.github.io/posts/2018-06-24-attention/

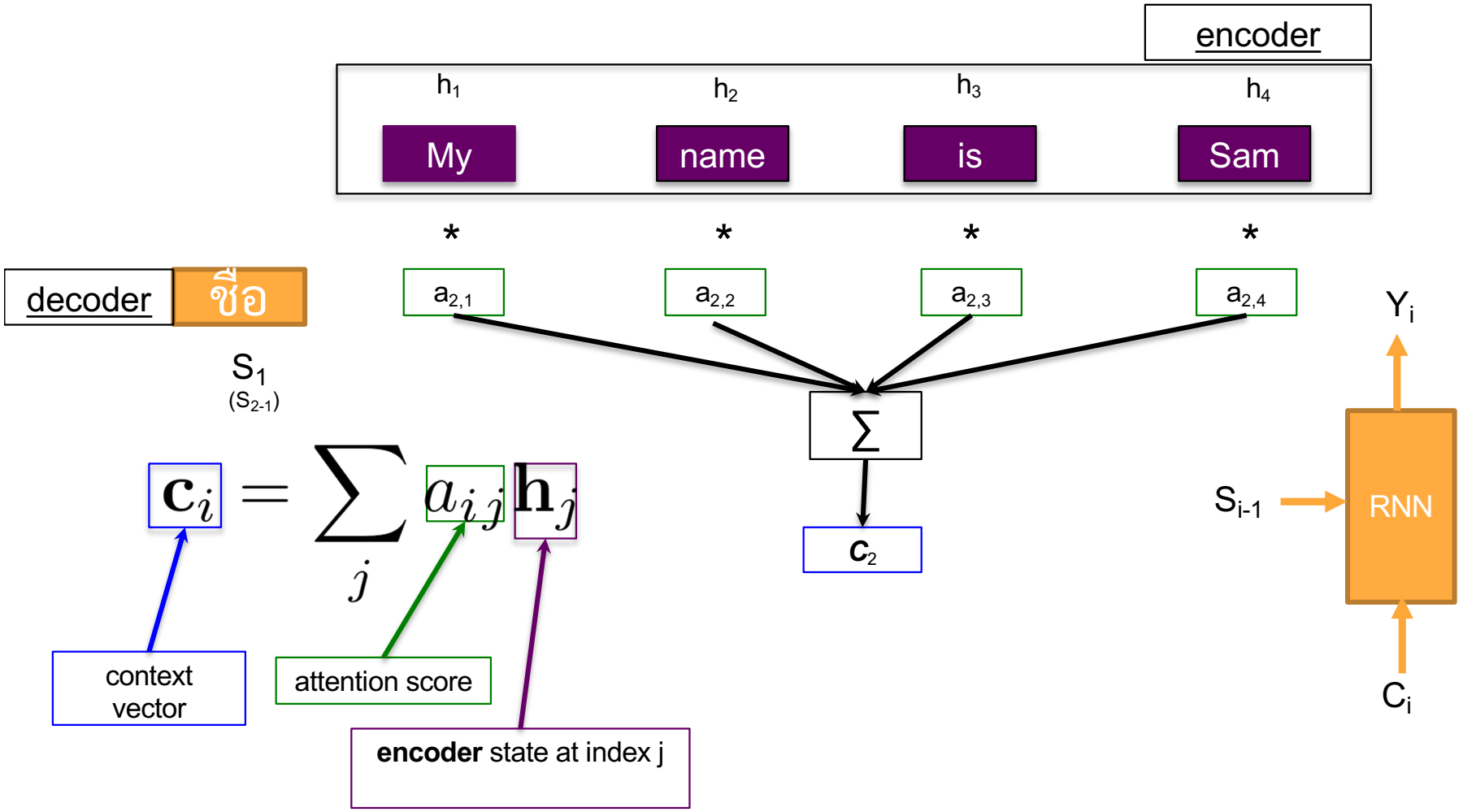
Attention Calculation Example (1): Attention Scores

Now we want to predict แซม (i=2)

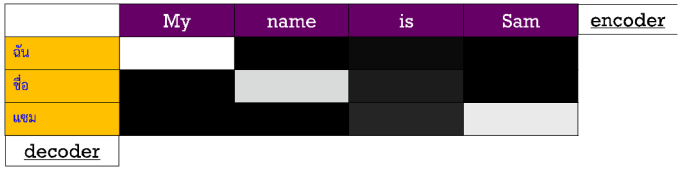


i = decoder index
j = encoder index

Attention Calculation Example (2): Context Vector



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



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).

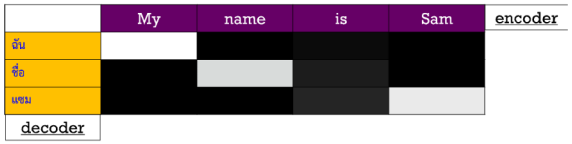
$$\mathbf{a} = softmax(\mathbf{w}_{s_2} tanh(\mathbf{W}_{s_1} \mathbf{H}^T))$$

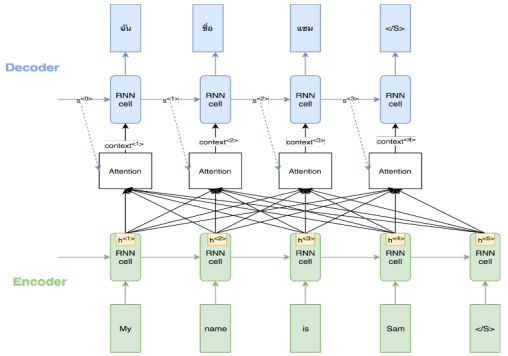
Key-value attention: key-value attention (Daniluk et al., 2017) is a recent attention variant that separates form from function by keeping separate vectors for the attention calculation.

Reference: https://lilianweng.github.io/posts/2018-06-24-attention/

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

1) Additive Attention





- 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)

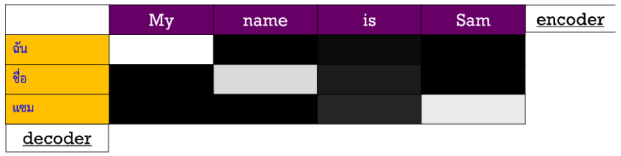
- Where $\mathbf{W}_a$ are learned attention parameters. Analogously, we can also use matrices $\mathbf{W}_1$ and $\mathbf{W}_2$ to learn separate transformations for $\mathbf{s}_{i-1}$ and $\mathbf{h}_j$ respectively, which are then summed (hence the name additive):

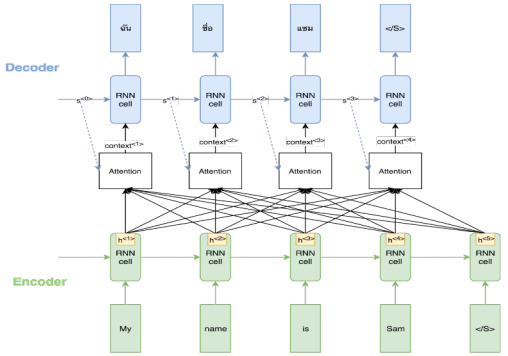
$$f_{attn}(\mathbf{s}_{i-1}, \mathbf{h}_j) = tanh(\mathbf{W}_1\mathbf{s}_{i-1} + \mathbf{W}_2\mathbf{h}_j)$$

Reference: https://lilianweng.github.io/posts/2018-06-24-attention/

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

2) Multiplicative Attention





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

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

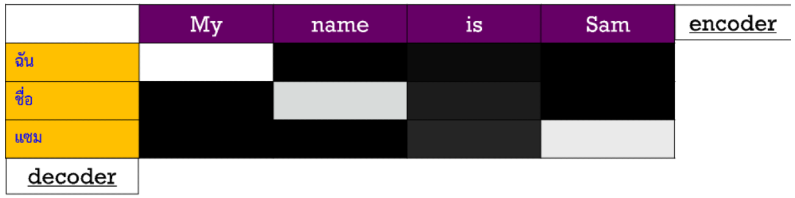
- Faster, more efficient than additive attention BUT additive attention performs better for larger dimensions
- One way to mitigate this is to scale f_{attn} by $\frac{1}{\sqrt{d_s}}$

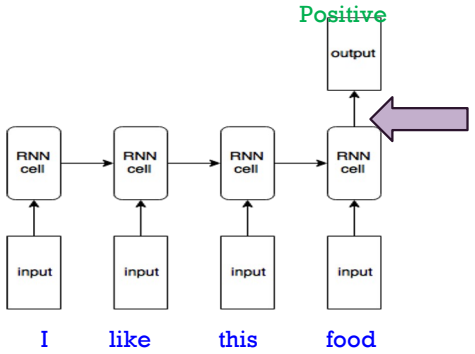
d_s = #dimensions of hidden states in LSTM (context vector; latent factors)

  - Dot product of high dimensional vectors has high variance -> softmax is peaky -> small gradient -> harder to train

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

3) 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)

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

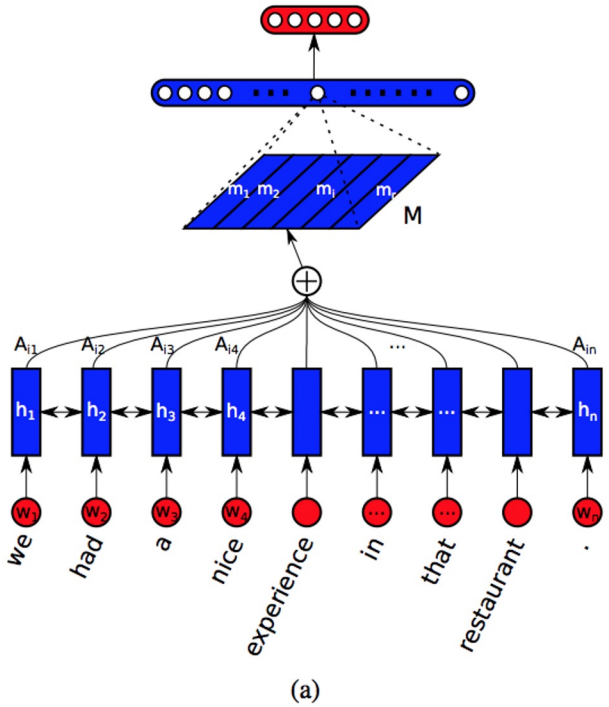
Fully connected layer

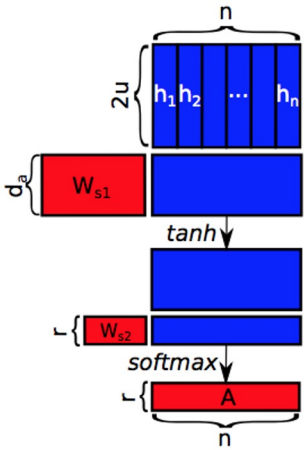
$$\mathbf{a} = softmax(\mathbf{w}_{s_2} tanh(\mathbf{W}_{s_1} \mathbf{H}^T))$$

One-hidden layer (Dense)

- 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.**





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

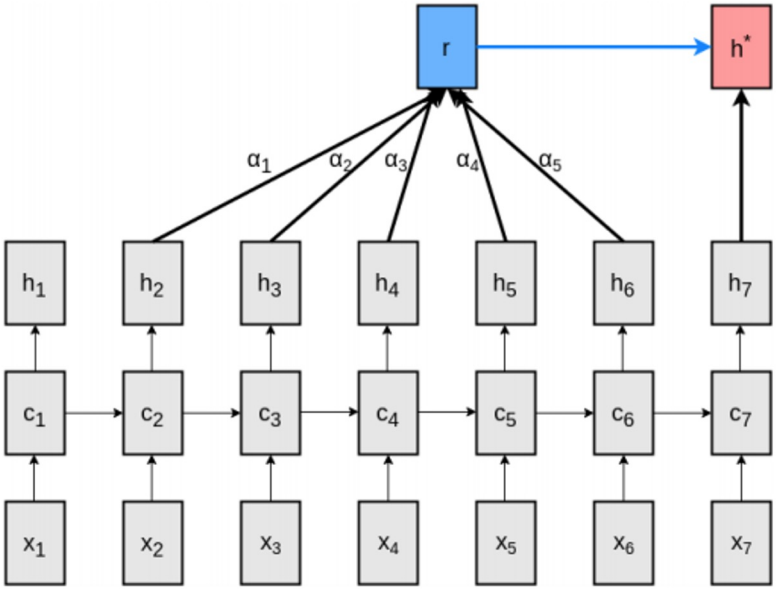
Figure 2: Heatmap of Yelp reviews with the two extreme score.

4) Key-value attention (1)

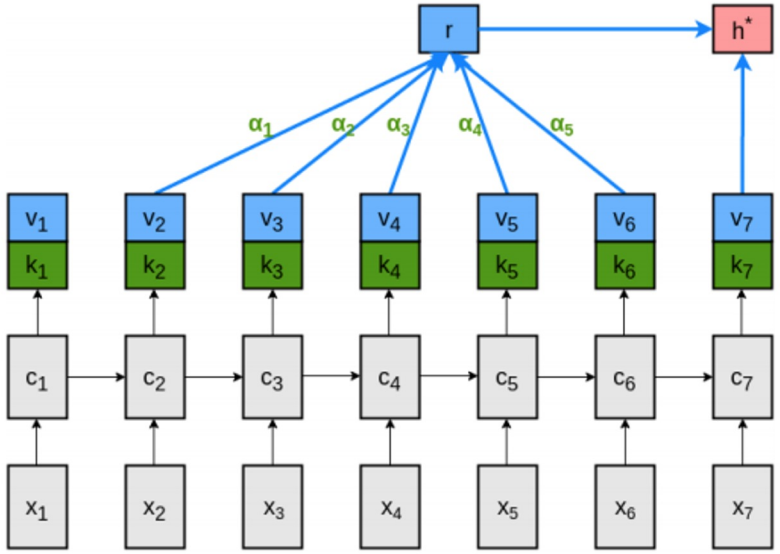
$$a_{ij} = softmax(f_{att}(\mathbf{s}_{i-1}, \mathbf{h}_j))$$

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

search value



(a) Neural language model with attention.



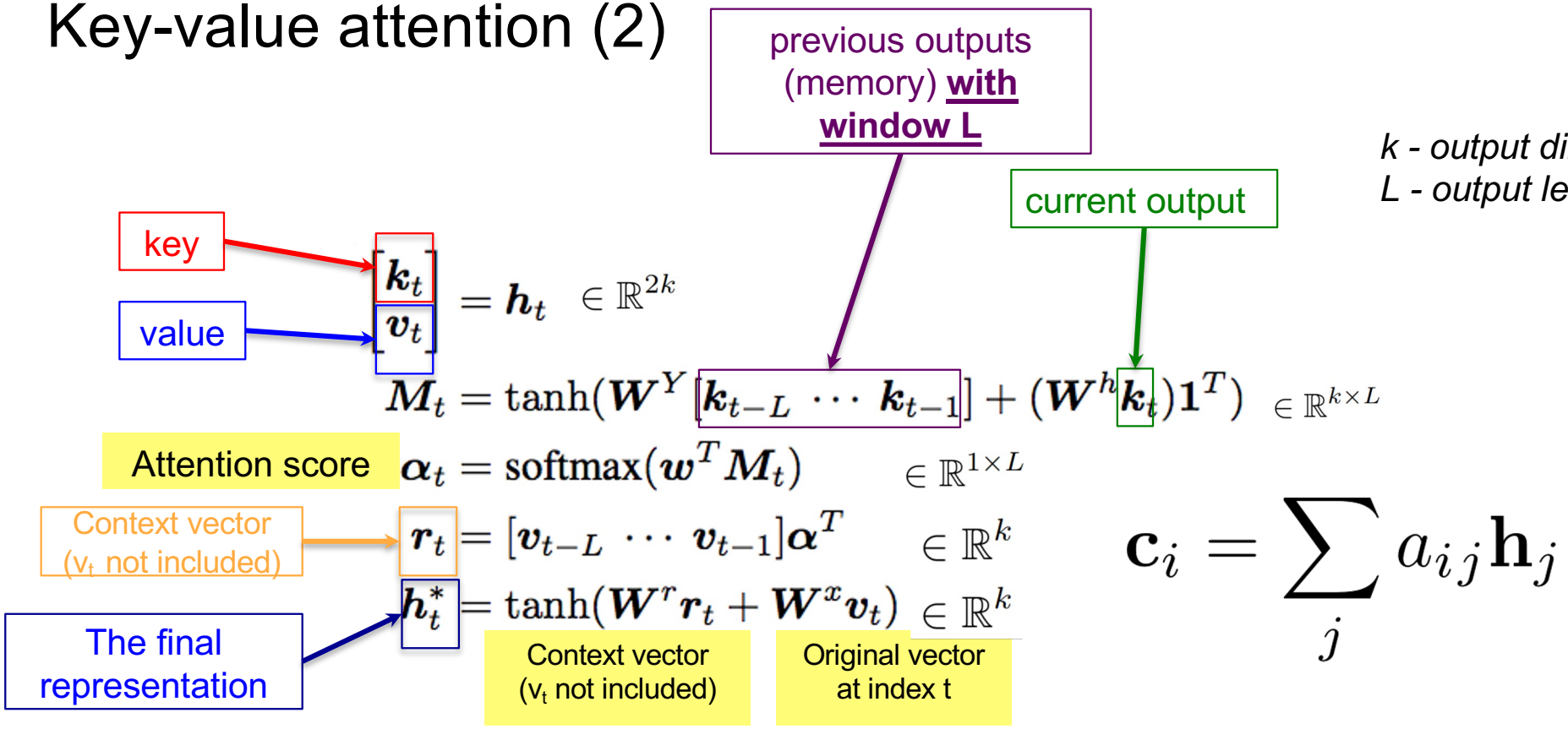
(b) Key-value separation.

Value=encoded vector

Key=used for attention score calculation

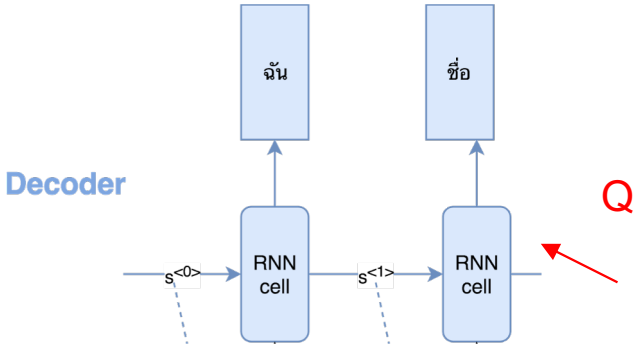
Reference: Daniluk, M., Rockt, T., Welbl, J., & Riedel, S. (2017). Frustratingly Short Attention Spans in Neural Language Modeling. In ICLR 2017.

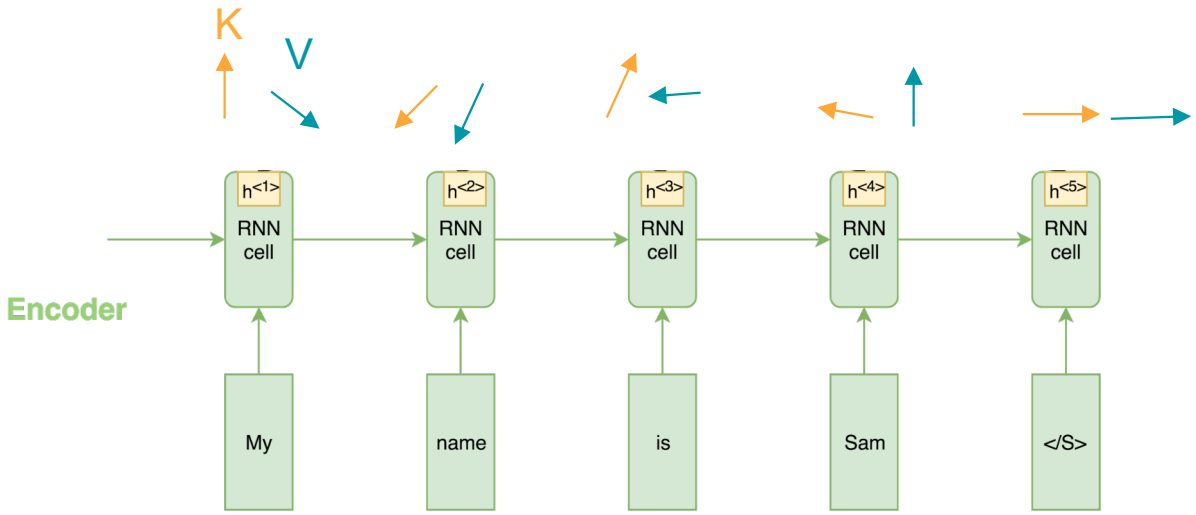
Key-value attention (2)



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

Pictorial view of KV attention



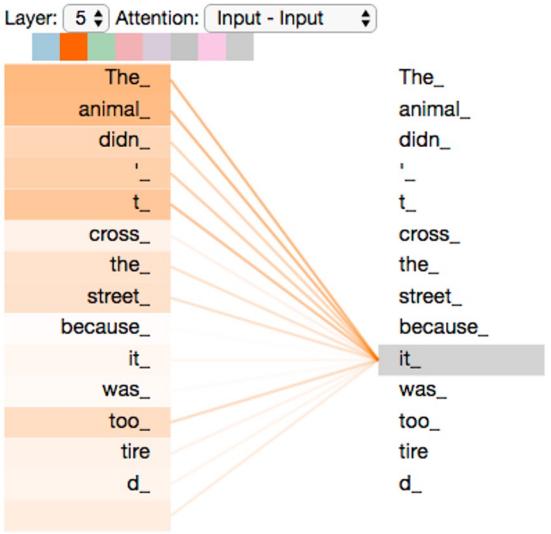


Scaled Dot-Product Attention (1)

- Introduced in Attention is all you need (Viswani et al., 2017)
- **NO** recurrence nor convolution
- Widely used today in all Transformer-based model
- "Relating different positions of a single sequence in order to compute a representation of the sequence"

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

where the query, keys, values, and output are all vectors and d_k is a number.



Attention Is All You Need. 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

Scaled Dot-Product Attention (2)

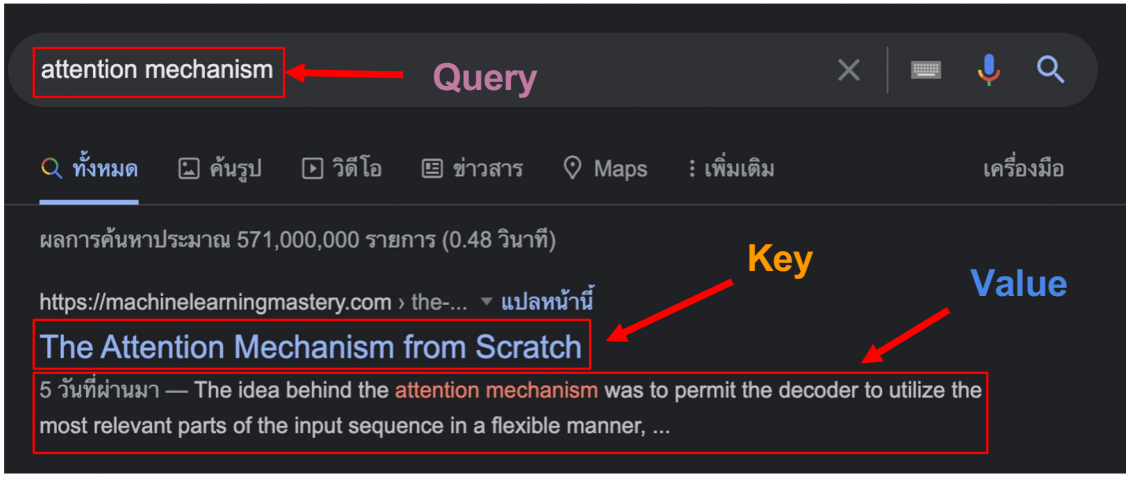
What are the “query”, “key”, and “value” vectors?

As an analogy, think of Google Search.

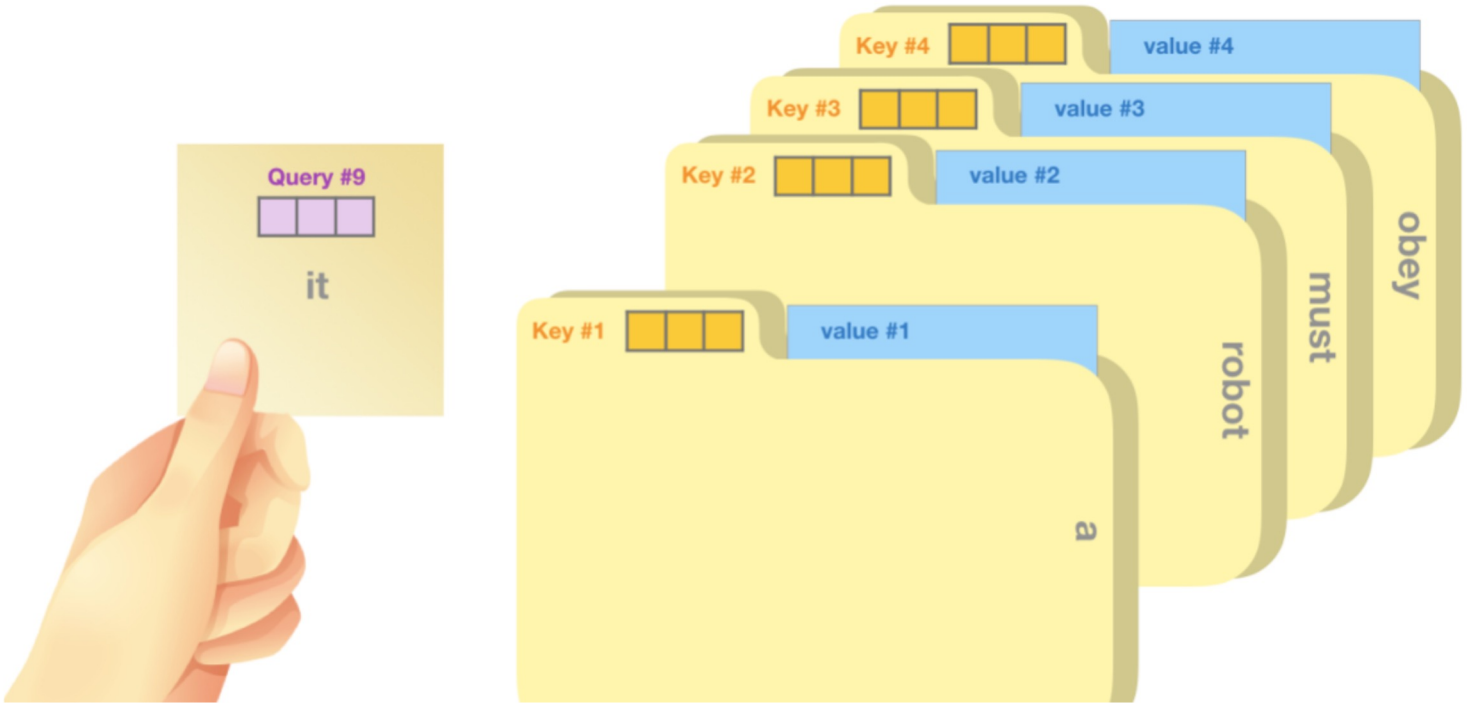
Query - what we want to know

Key - how to index information

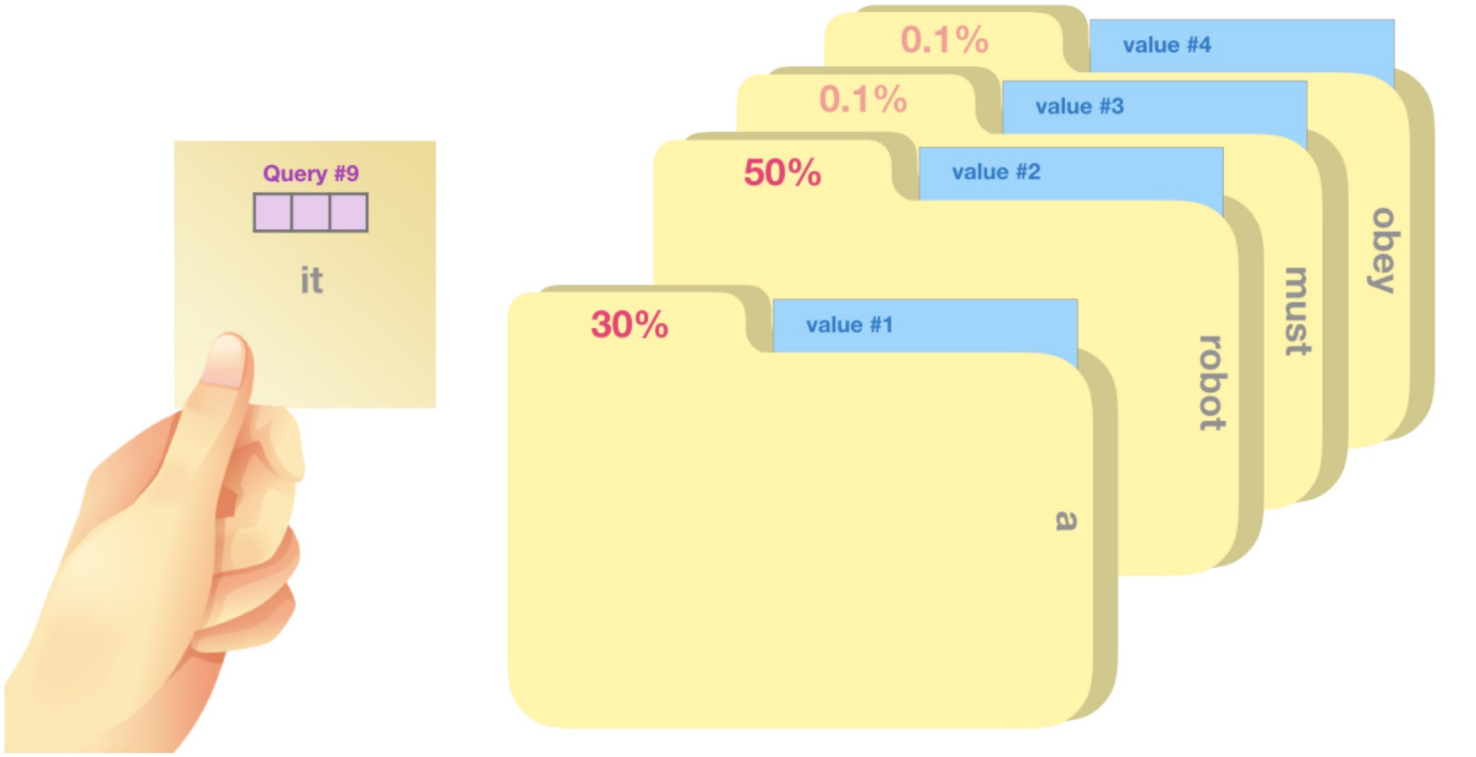
Value - what kind of info is in each website



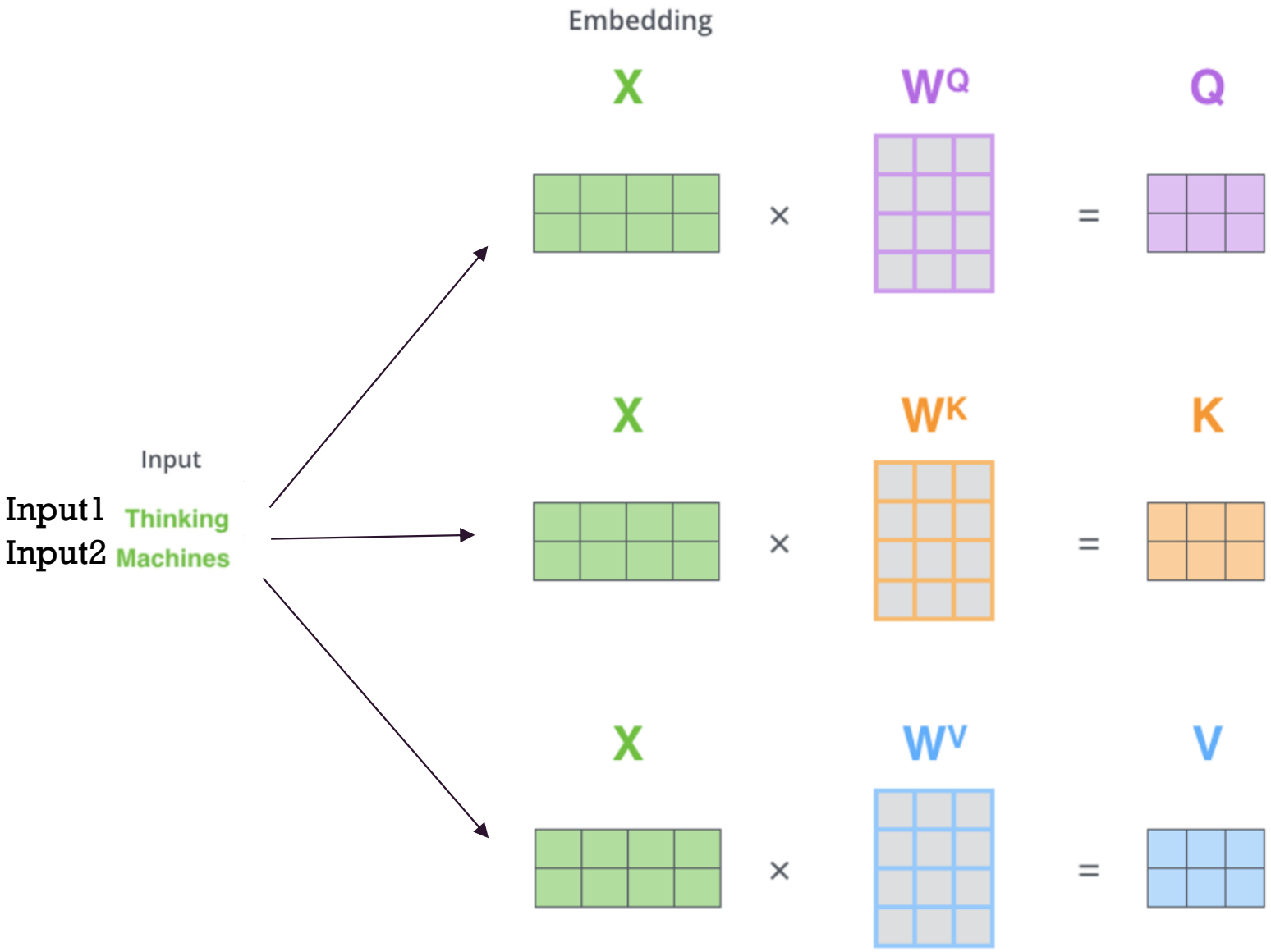
Attention Is All You Need. 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.



http://jalammar.github.io/illustrated-gpt2/

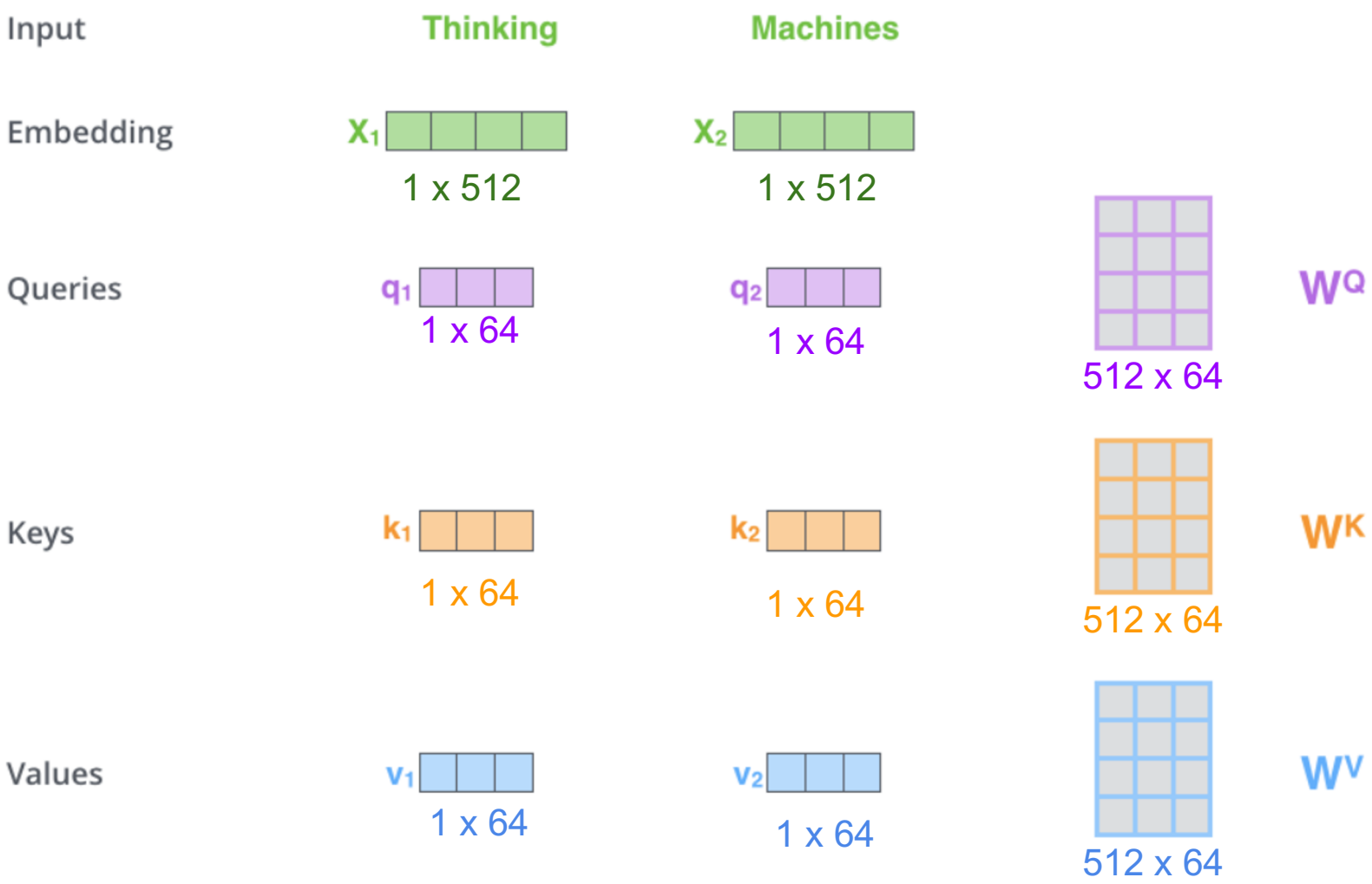


http://jalammar.github.io/illustrated-gpt2/



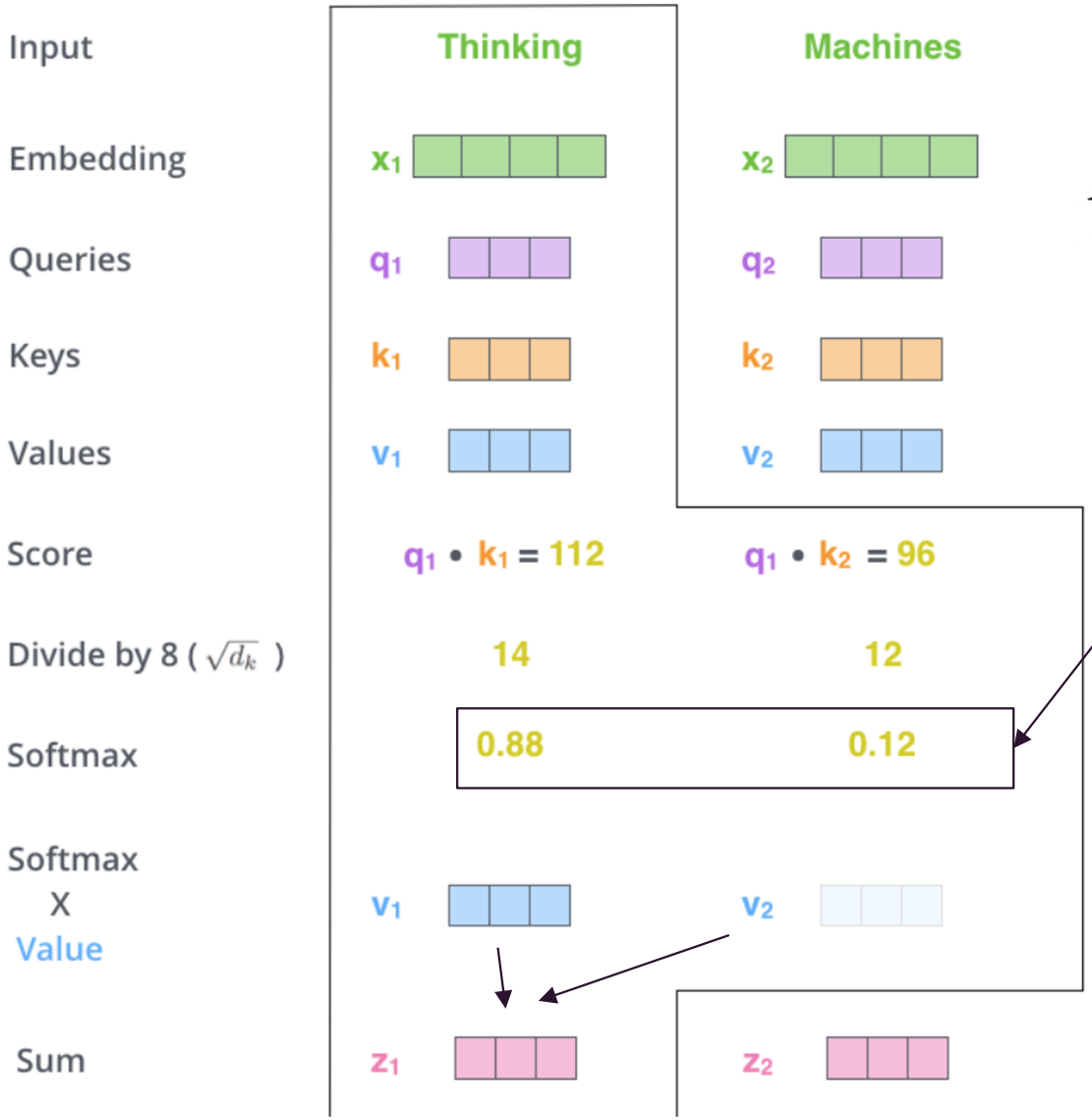
http://jalammar.github.io/illustrated-transformer/

W^Q, W^K, and W^V are linear layers



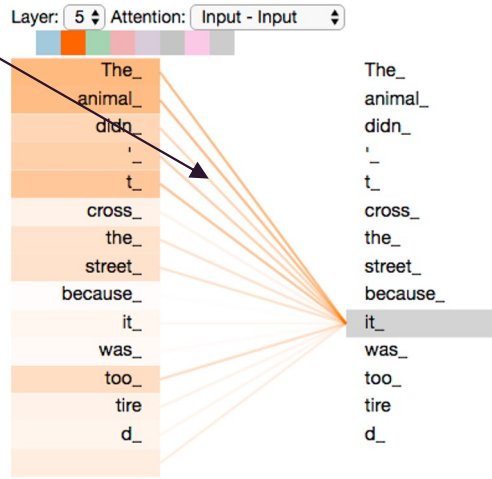
Multiplying x1 by the WQ weight matrix produces q1, the "query" vector associated with that word. We end up creating a "query", a "key", and a "value" projection of each word in the input sentence.

http://jalammar.github.io/illustrated-transformer/



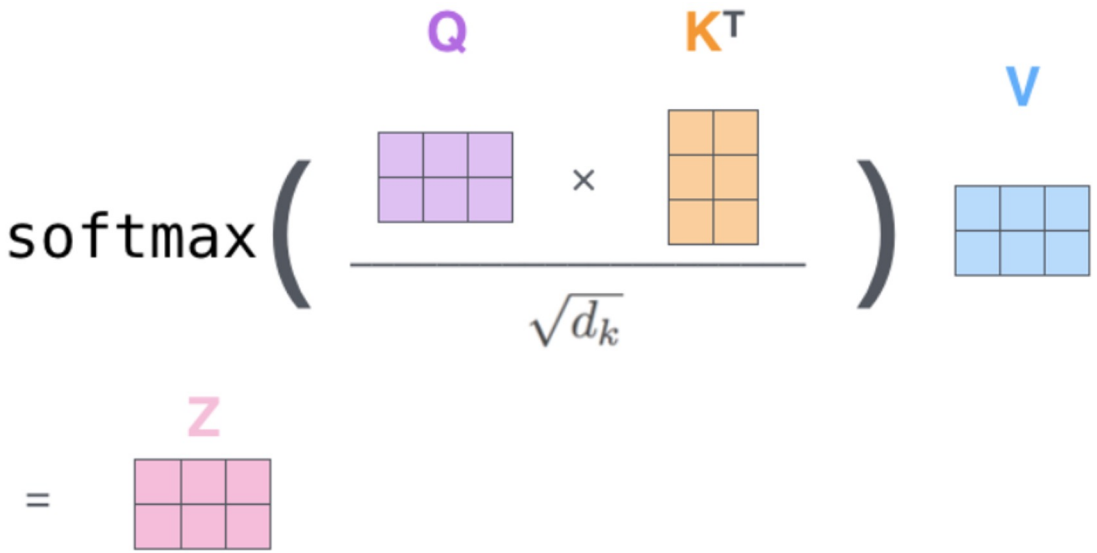
$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

Attention scores

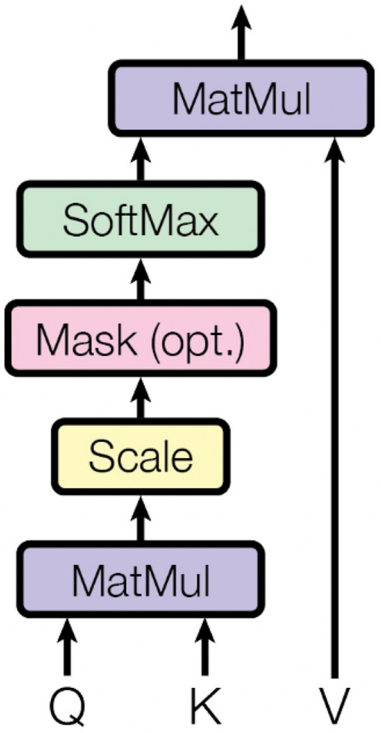


http://jalammar.github.io/illustrated-transformer/

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$



Scaled Dot-Product Attention



http://jalammar.github.io/illustrated-transformer/