Attention in NNs: the advent of Transformers

Roberto Basili, Danilo Croce
Machine Learning, Web Mining & Retrieval 2022/2023

Outline

- Attention Mechanisms in Recurrent Networks
- Trasformers
- Applications to Language Processing
- Perspectives

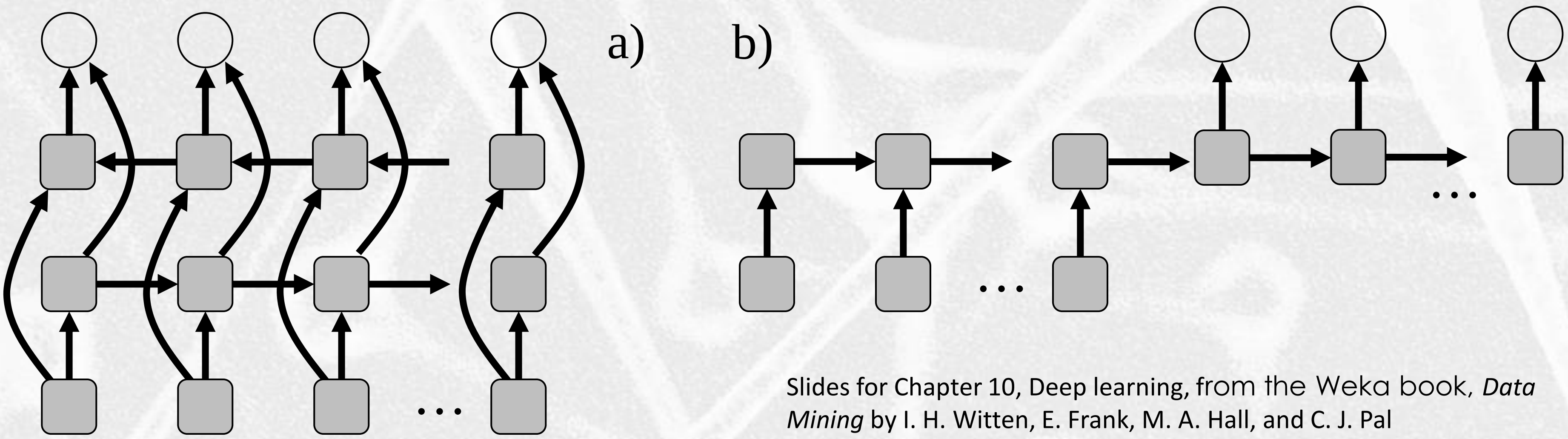
Other RNN architectures

a) Recurrent networks can be made bidirectional, propagating information in both directions

- They have been used for a wide variety of applications, including protein secondary structure prediction and handwriting recognition

b) An "encoder-decoder" network creates a fixed-length vector representation for variable-length inputs, the encoding can be used to generate a variable-length sequence as the output

- Particularly useful for machine translation



Training different Types of RNNs

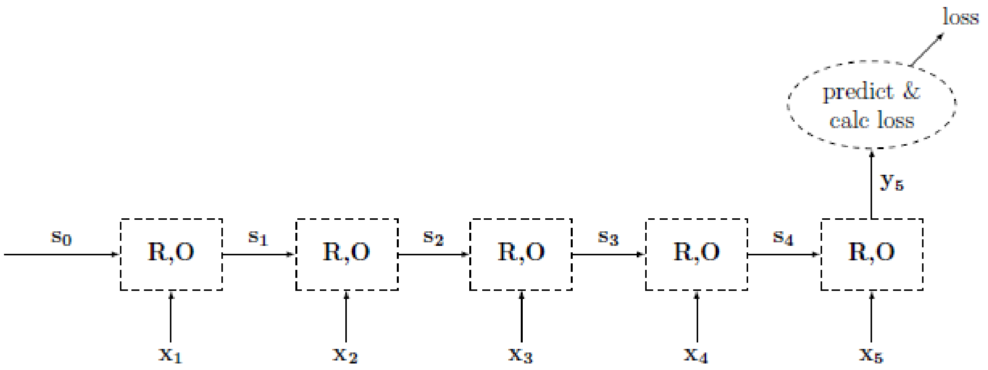


Figure 7: Acceptor RNN Training Graph.

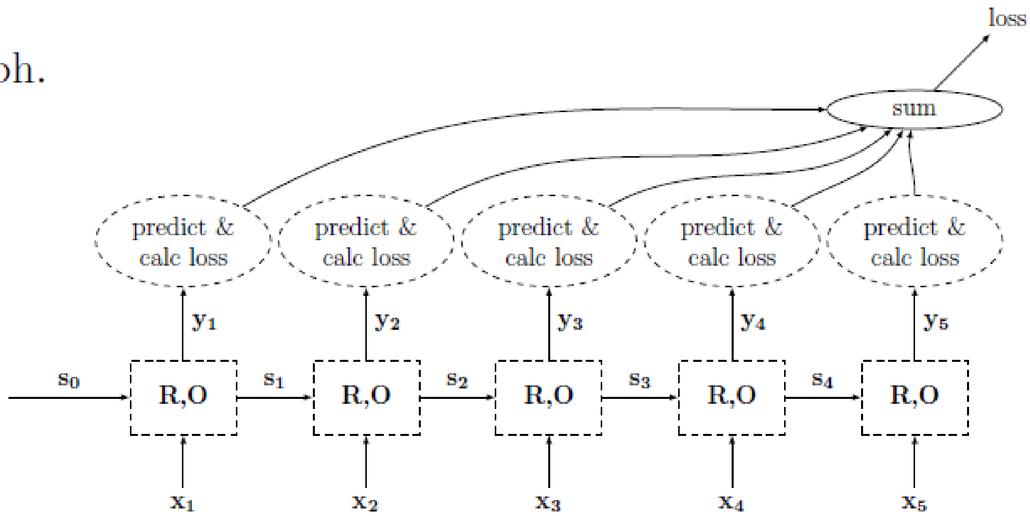


Figure 8: Transducer RNN Training Graph.

Training different Types of RNNs

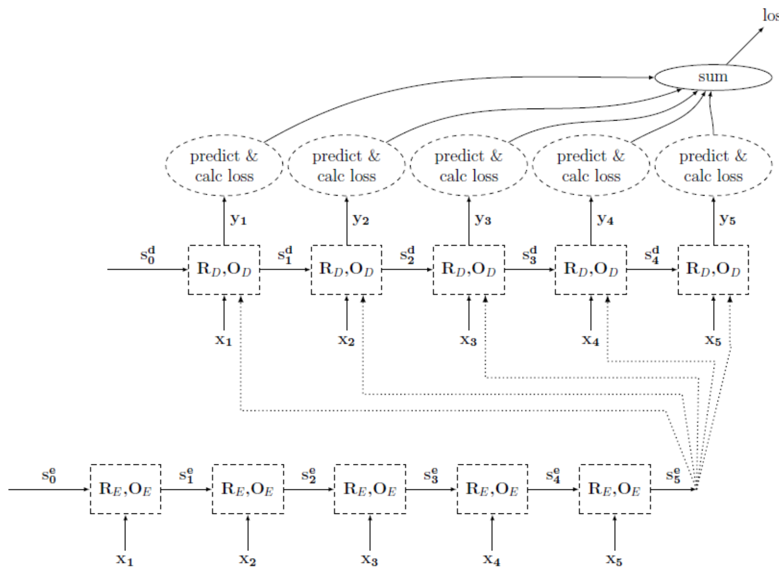


Figure 9: Encoder-Decoder RNN Training Graph.

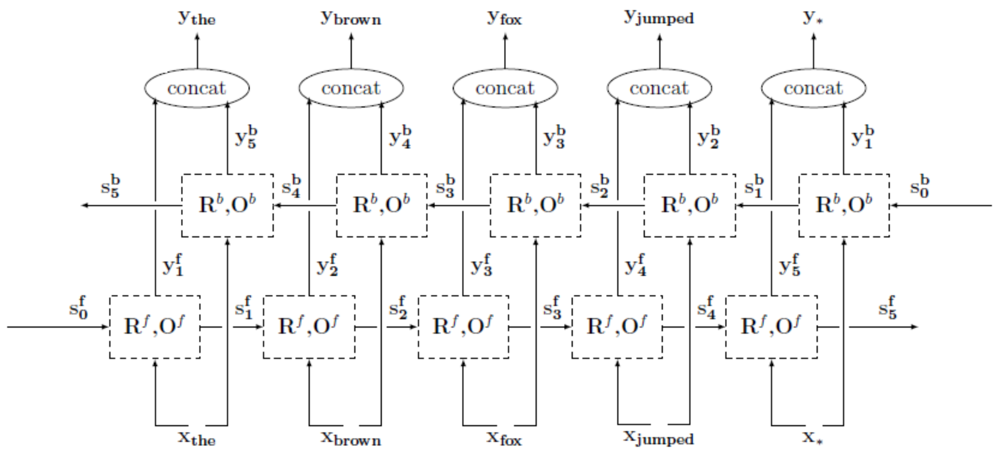
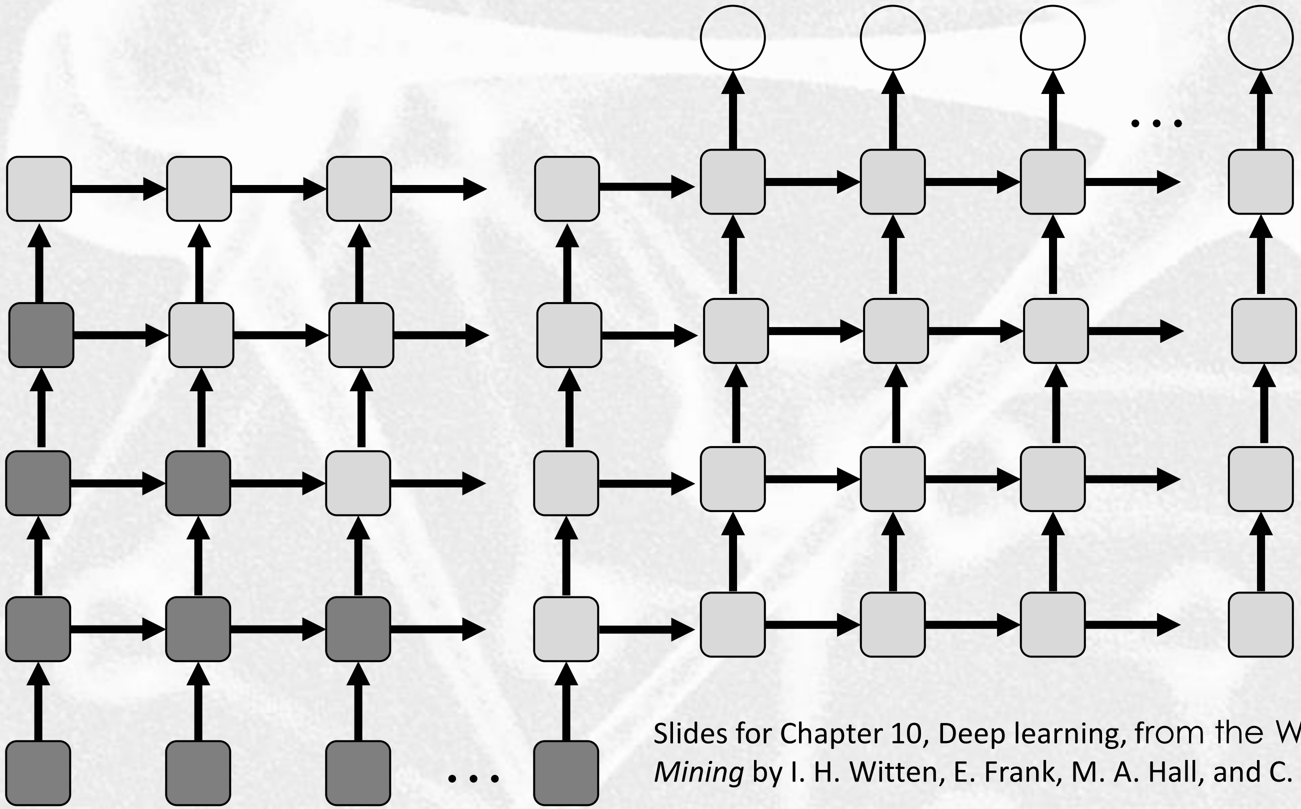


Figure 11: biRNN over the sentence "the brown fox jumped ."

Encoder-decoder deep architectures

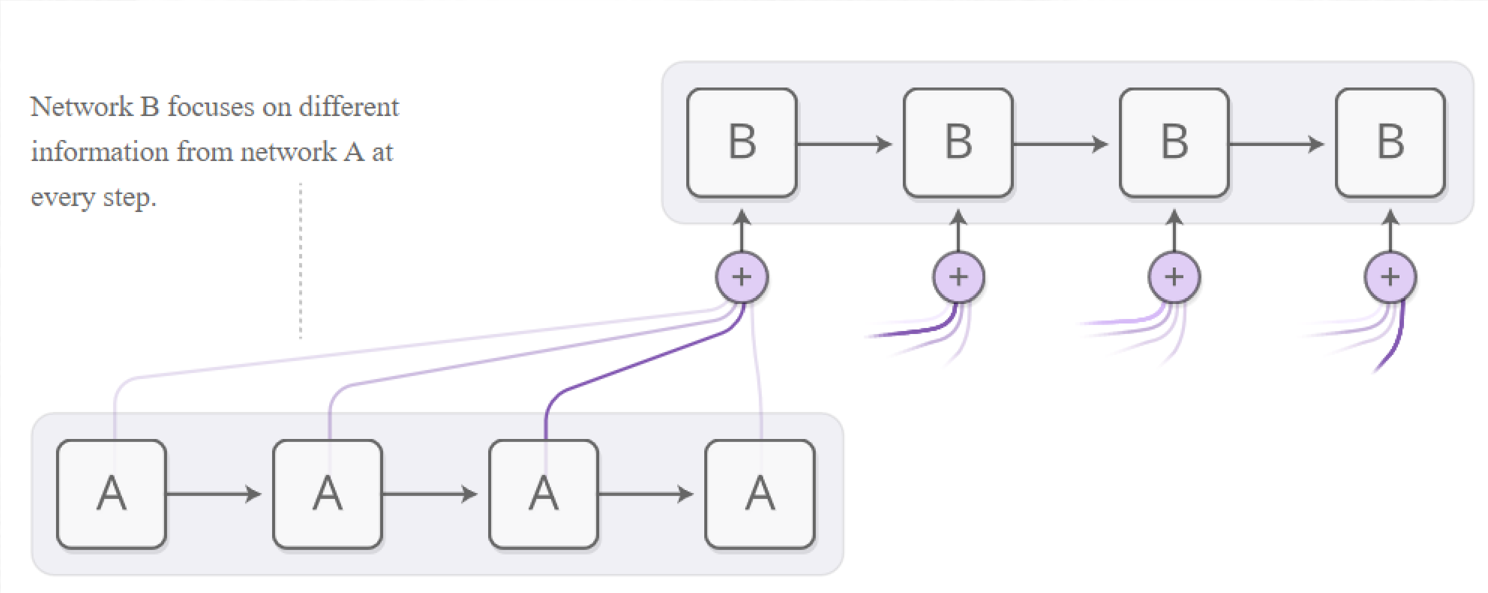
- Given enough data, a deep encoder-decoder architecture (see below) can yield results that compete with hand-engineered translation systems.

- The connectivity structure means that partial computations in the model can flow through the graph in a wave (darker nodes in fig.)



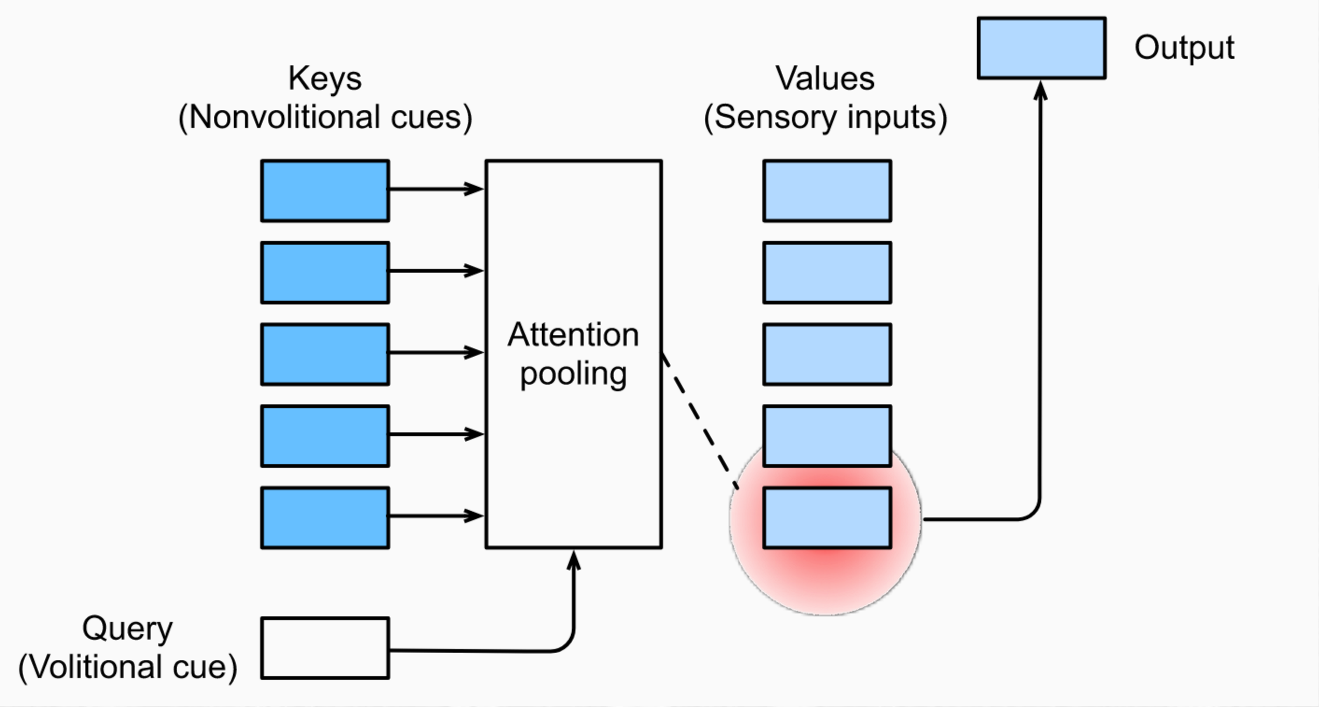
Attention-based RNNs

- A NN (e.g. B) is used to attend the outcome of a second network A, e.g. (Vaswani et al., 2017)

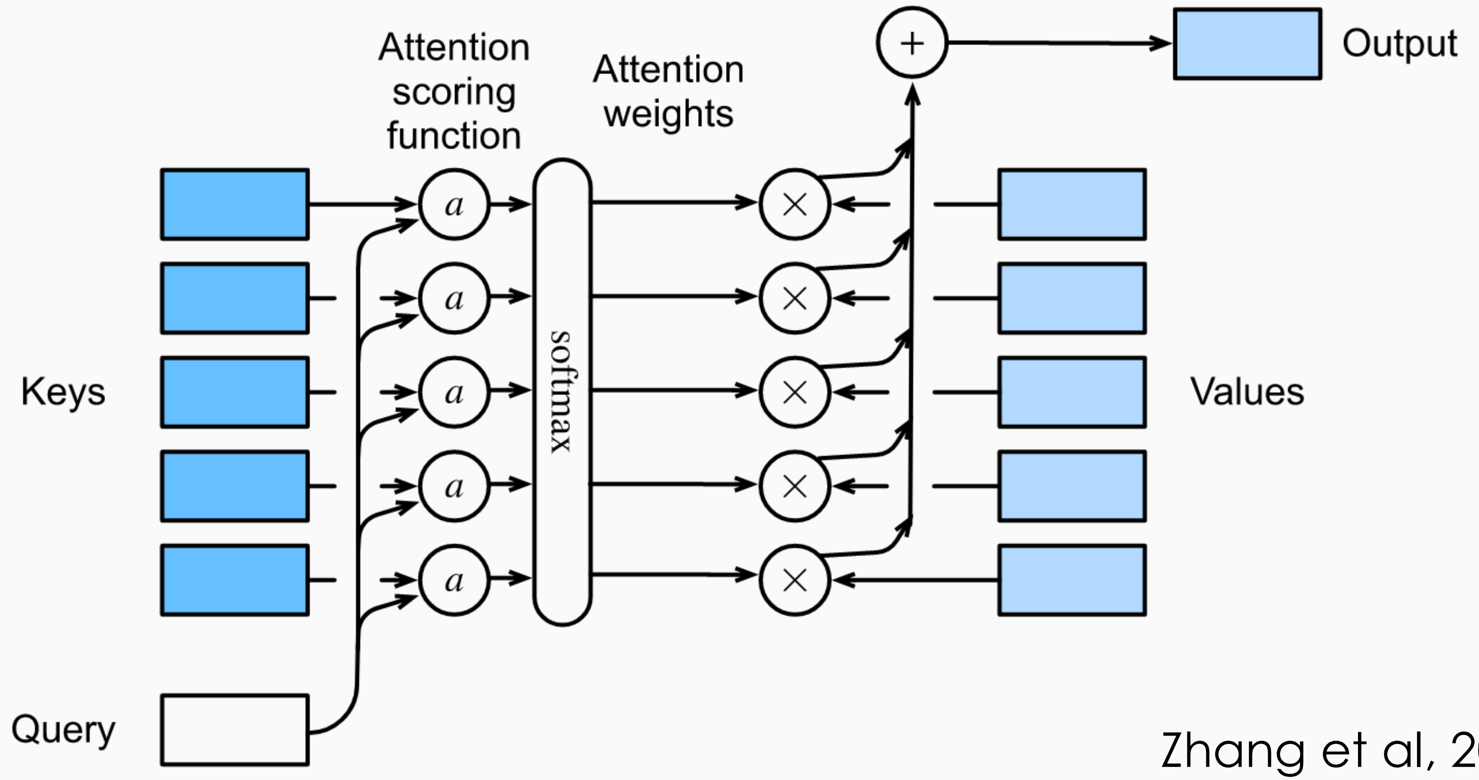


Attention: motivations

- From (*Dive into Deep Learning*, Zhang, Aston and Lipton, Zachary C. and Li, Mu and Smola, Alexander J., 2021).

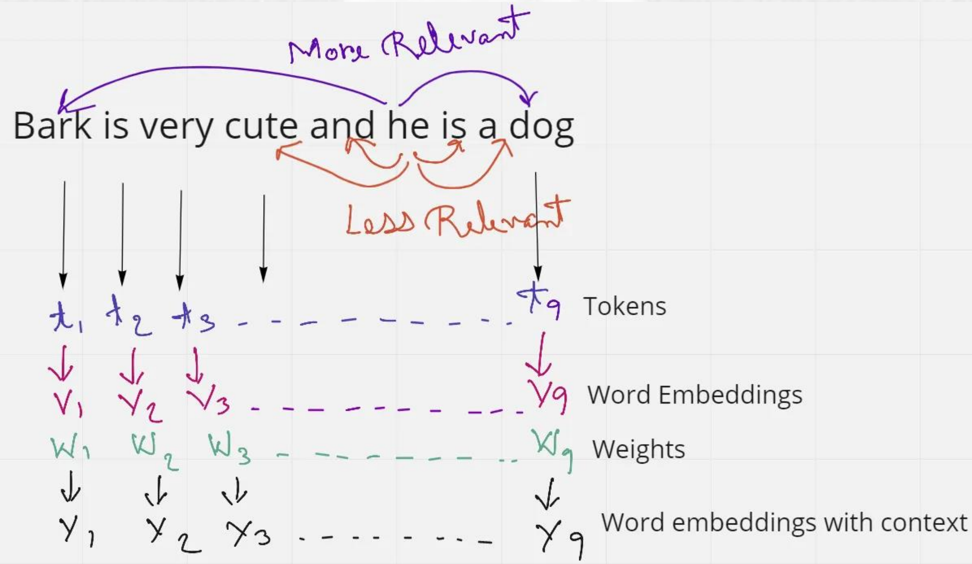


Attention functions



Zhang et al, 2021

Inside Attention



1. Finding the Weights

$V_1 V_1 = W_{11}$

$V_1 V_2 = W_{12}$

$V_1 V_3 = W_{13}$

$\vdots$

$V_1 V_9 = W_{19}$

$\xrightarrow{\text{Normalize}}$

$$\left.\begin{matrix} W_{11} \\ W_{12} \\ W_{13} \\ \vdots \\ W_{19} \end{matrix}\right\}$$ Weights to re-weigh the first vector

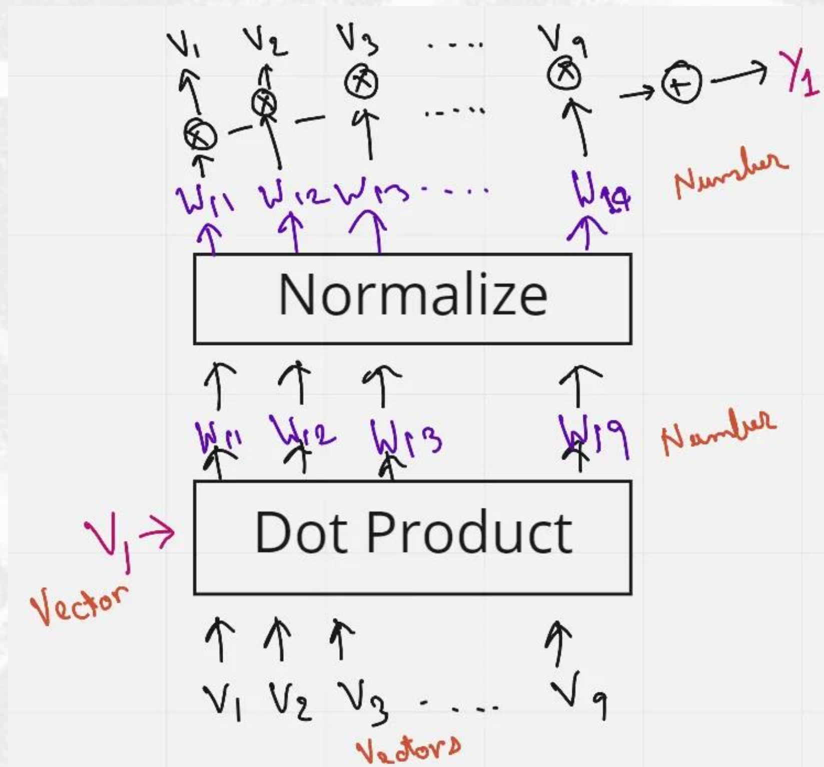
2 Obtaining Embedding with context

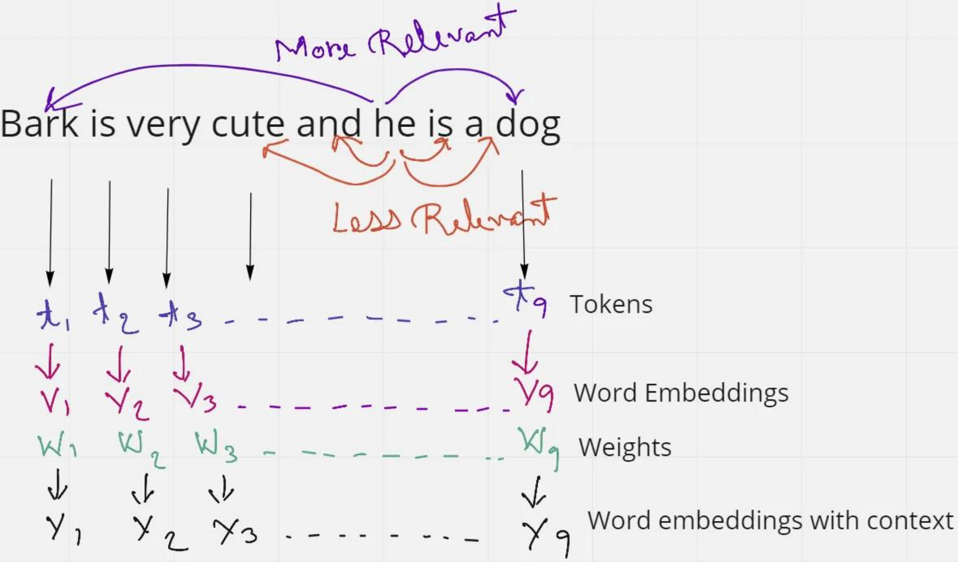
$W_{11}V_1 + W_{12}V_2 + W_{13}V_3 \ldots + W_{19}V_9 = Y_1$

$W_{21}V_1 + W_{22}V_2 + W_{23}V_3 \ldots W_{29}V_9 = Y_2$

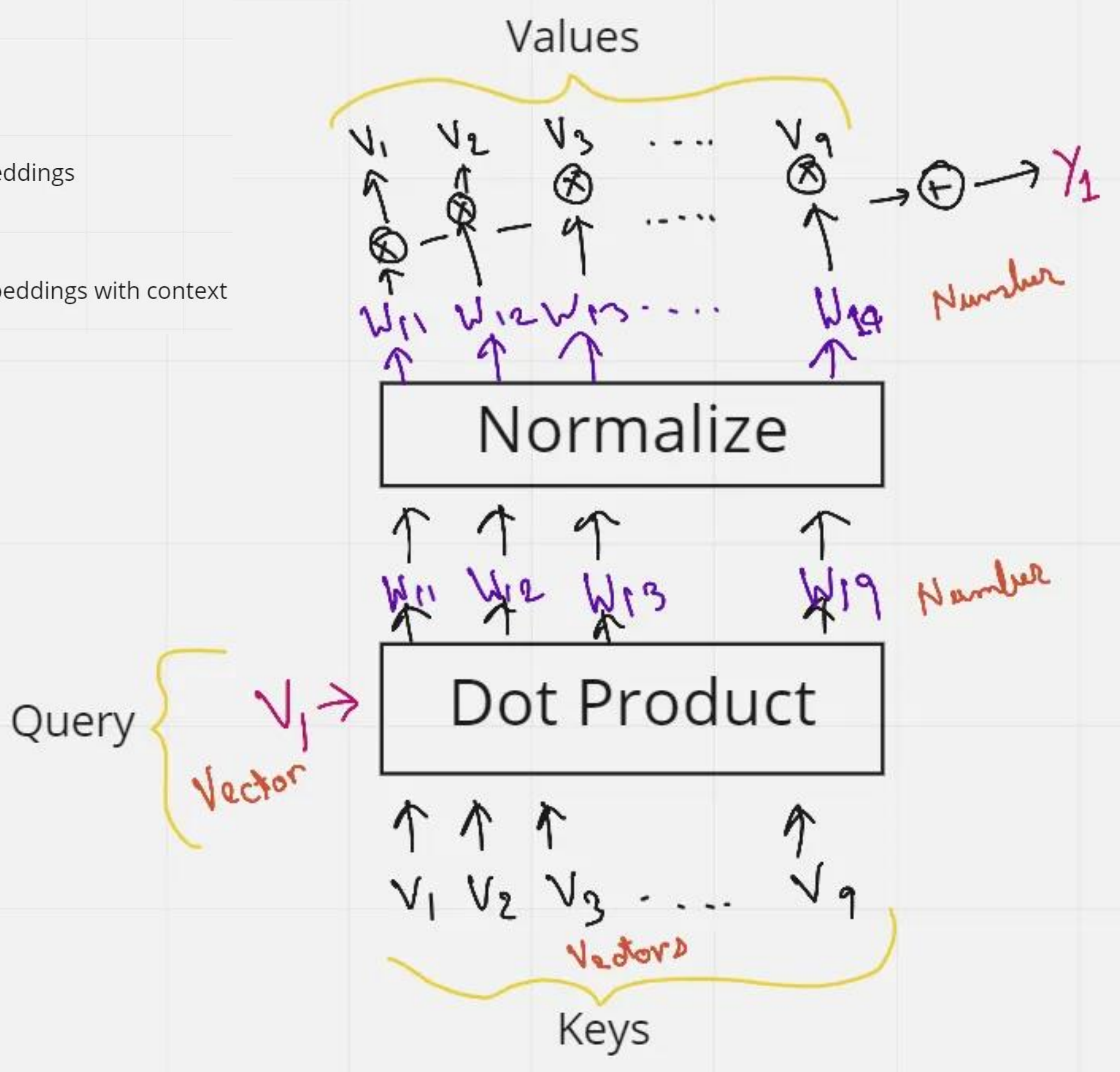
$\vdots$

$W_{91}V_1 + W_{92}V_2 + W_{93}V_3 \ldots W_{99}V_9 = Y_9$

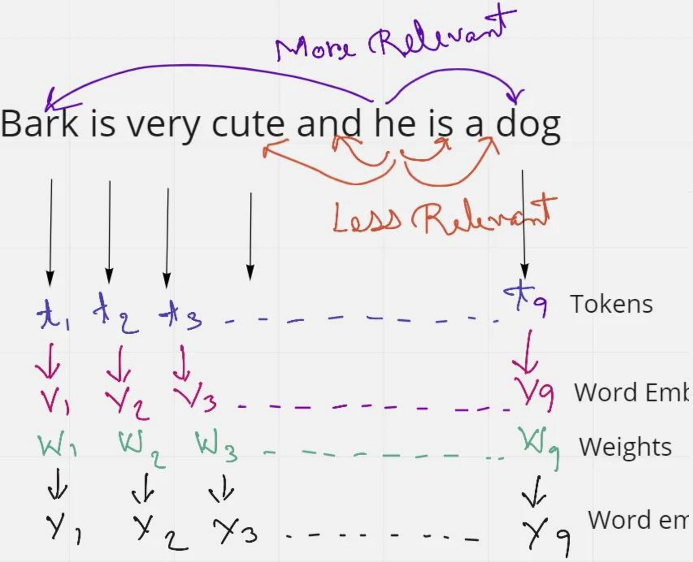




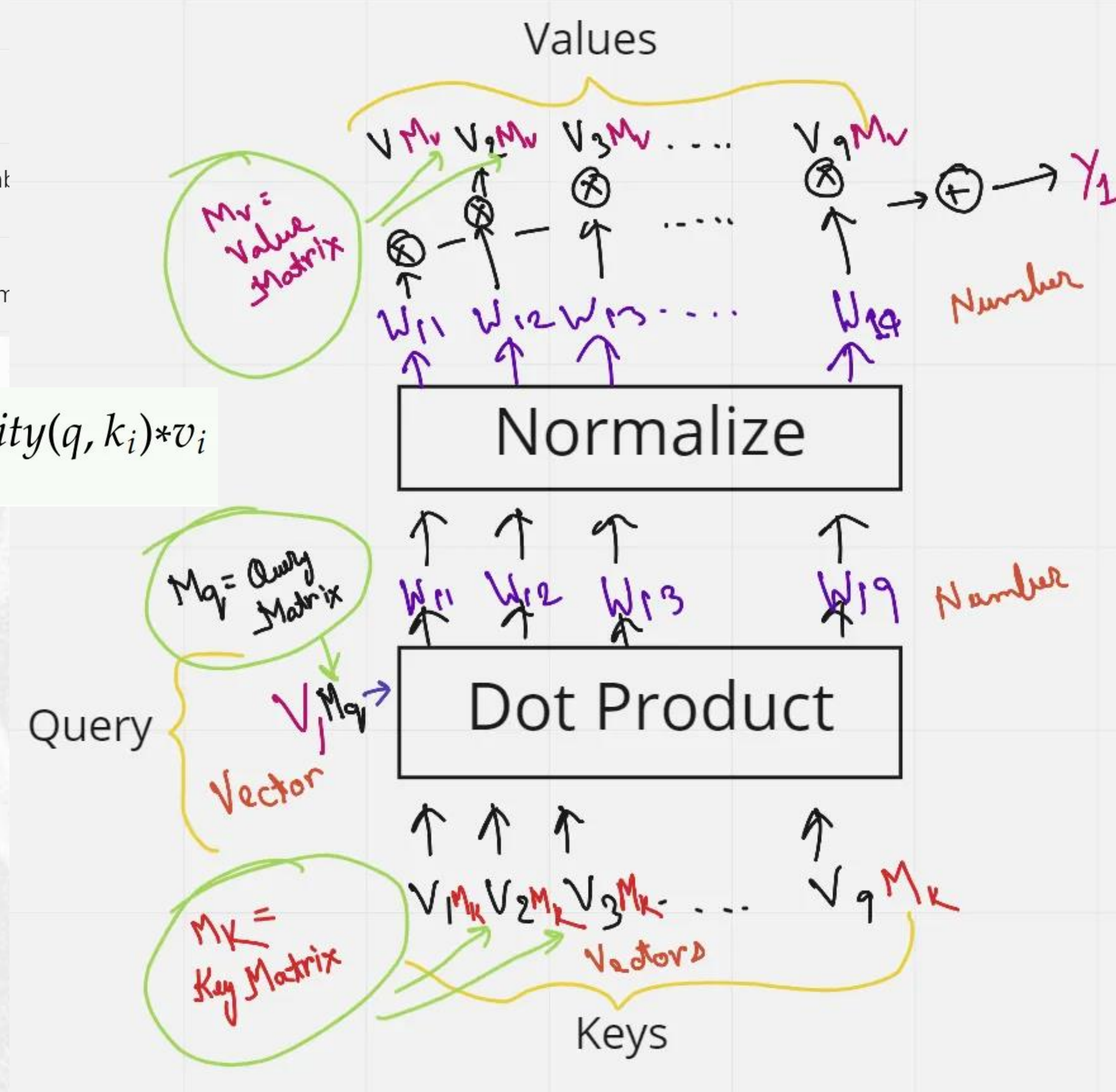
More Relevant
Bark is very cute and he is a dog
Less Relevant
t1 t2 t3 - - - - - - - t9 Tokens
V1 V2 V3 - - - - - - - V9 Word Embeddings
W1 W2 W3 - - - - - - - W9 Weights
Y1 Y2 Y3 - - - - - - - Y9 Word embeddings with context

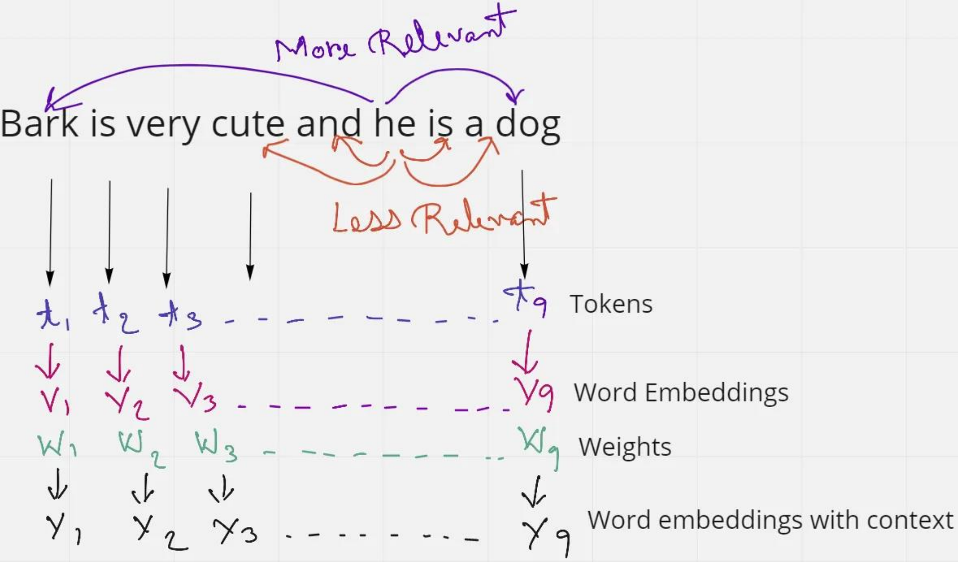


Values
V1 V2 V3 .... V9
Y1
W11 W12 W13 .... W19
Number
Normalize
W11 W12 W13 W19
Number
Query
V1
Vector
Dot Product
V1 V2 V3 .... V9
Vectors
Keys

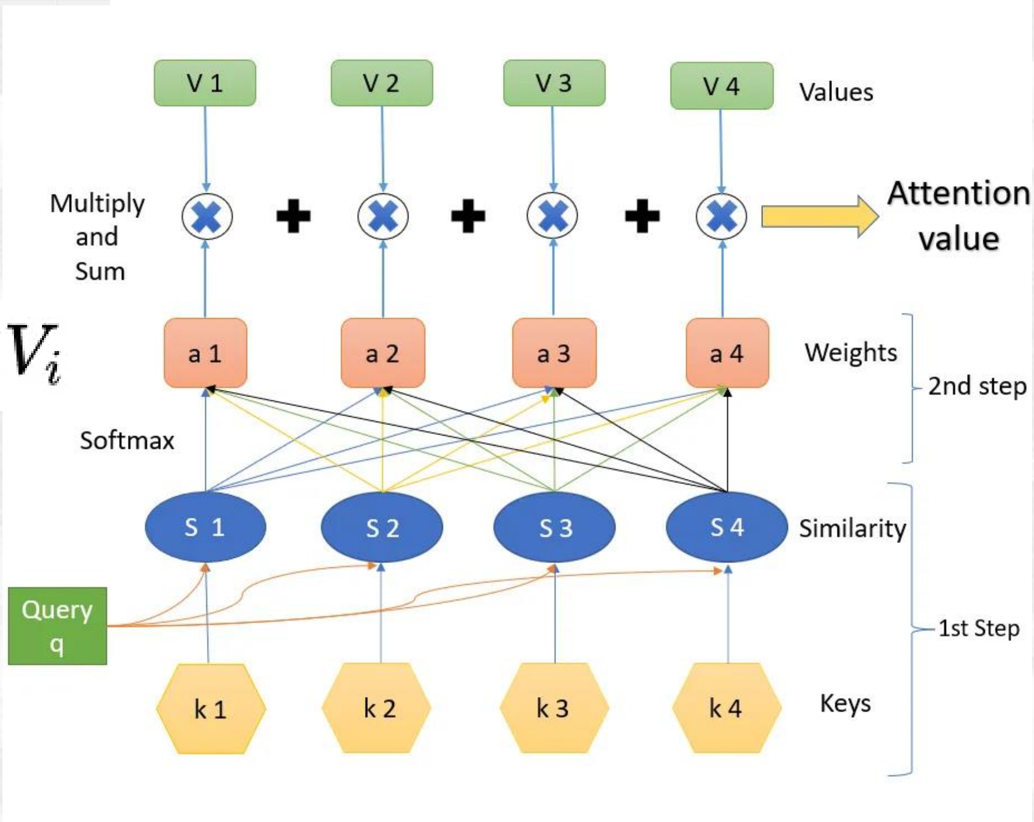


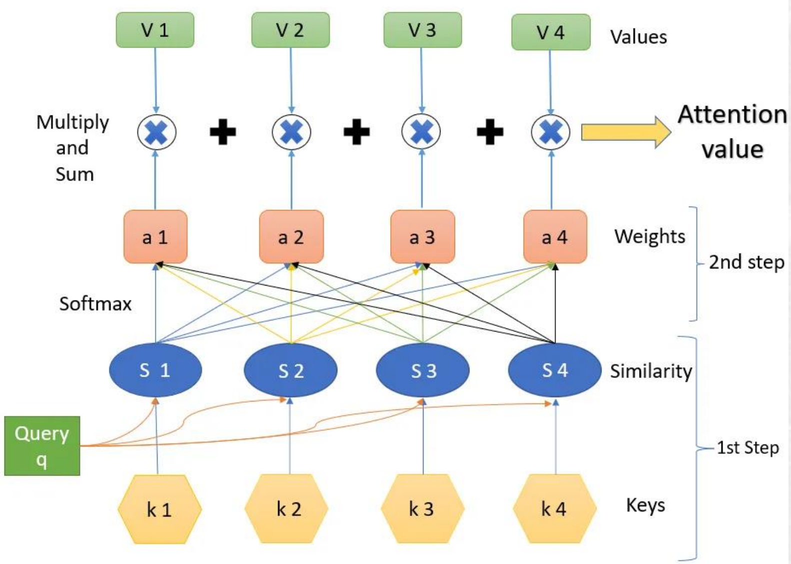
$$attention(q, k, v) = \sum_i similarity(q, k_i) * v_i$$

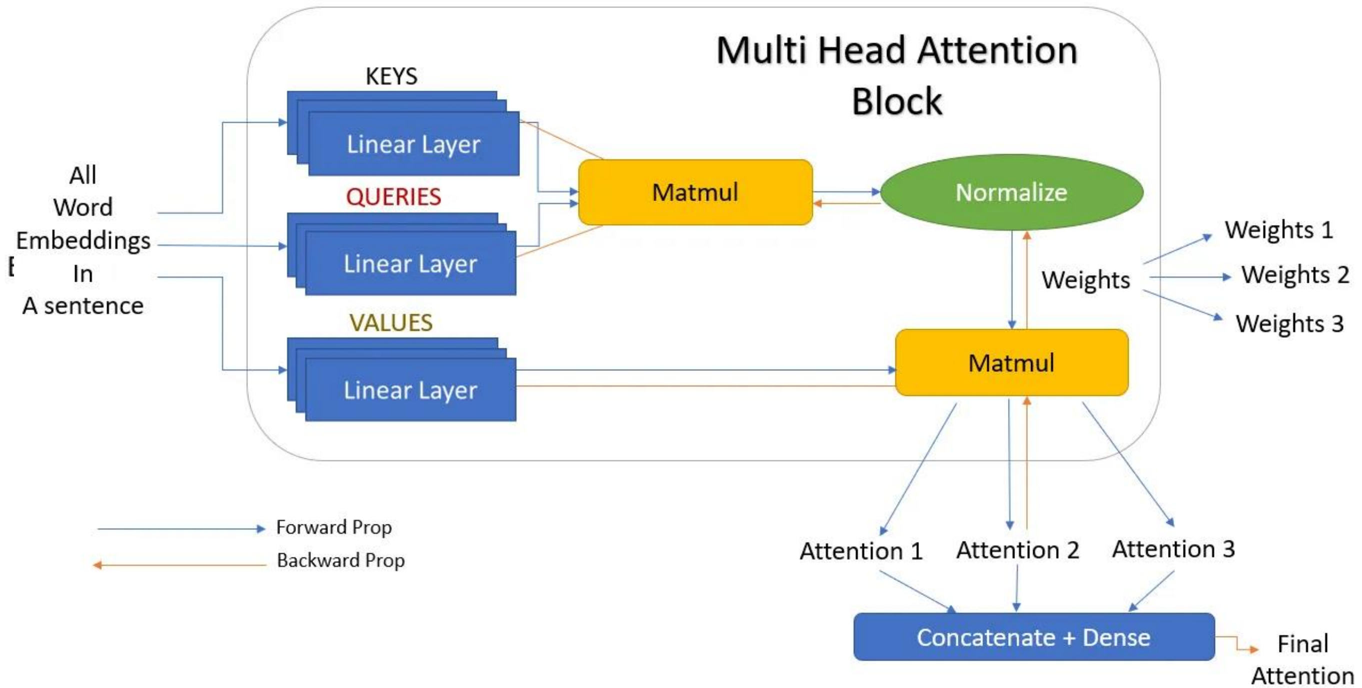


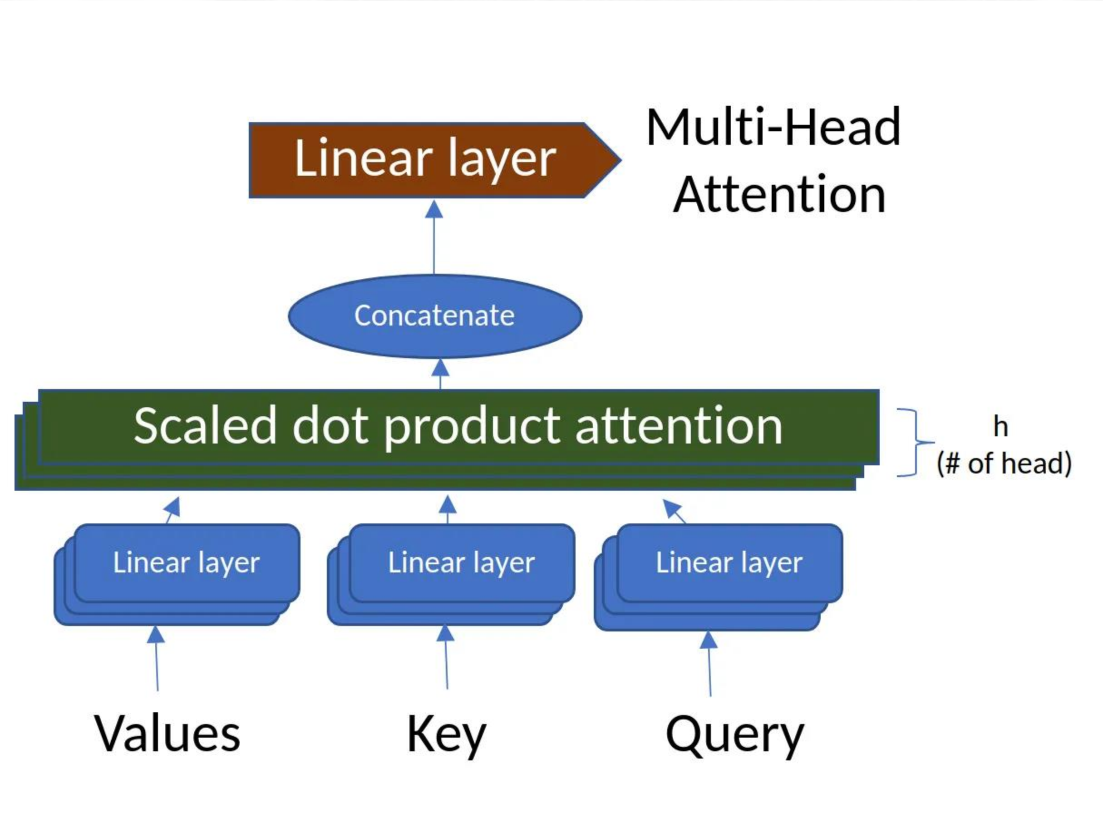


$$attention\ value = \sum_i a_i V_i$$




V 1
V 2
V 3
V 4
Values
Multiply and Sum
Attention value
a 1
a 2
a 3
a 4
Weights
2nd step
Softmax
S 1
S 2
S 3
S 4
Similarity
Query q
1st Step
k 1
k 2
k 3
k 4
Keys


Multi Head Attention Block
KEYS
Linear Layer
QUERIES
Linear Layer
VALUES
Linear Layer
All Word Embeddings In A sentence
Matmul
Normalize
Weights
Weights 1
Weights 2
Weights 3
Matmul
Attention 1
Attention 2
Attention 3
Concatenate + Dense
Final Attention
Forward Prop
Backward Prop


Linear layer
Multi-Head
Attention
Concatenate
Scaled dot product attention
h
(# of head)
Linear layer
Linear layer
Linear layer
Values
Key
Query

Attention functions: examples (1)

- In general, when queries and keys are vectors of different lengths, we can use additive attention as the scoring function. Given a query $\mathbf{q} \in \mathbb{R}^q$ and a key $\mathbf{k} \in \mathbb{R}^k$, the *additive attention* scoring function

$$a(\mathbf{q}, \mathbf{k}) = \mathbf{w}_v^\top \tanh(\mathbf{W}_q\mathbf{q} + \mathbf{W}_k\mathbf{k}) \in \mathbb{R},$$

- where learnable parameters $\mathbf{W}_q \in \mathbb{R}^{h\times q}$, $\mathbf{W}_k \in \mathbb{R}^{h\times k}$ and $\mathbf{w}_v \in \mathbb{R}^h$.

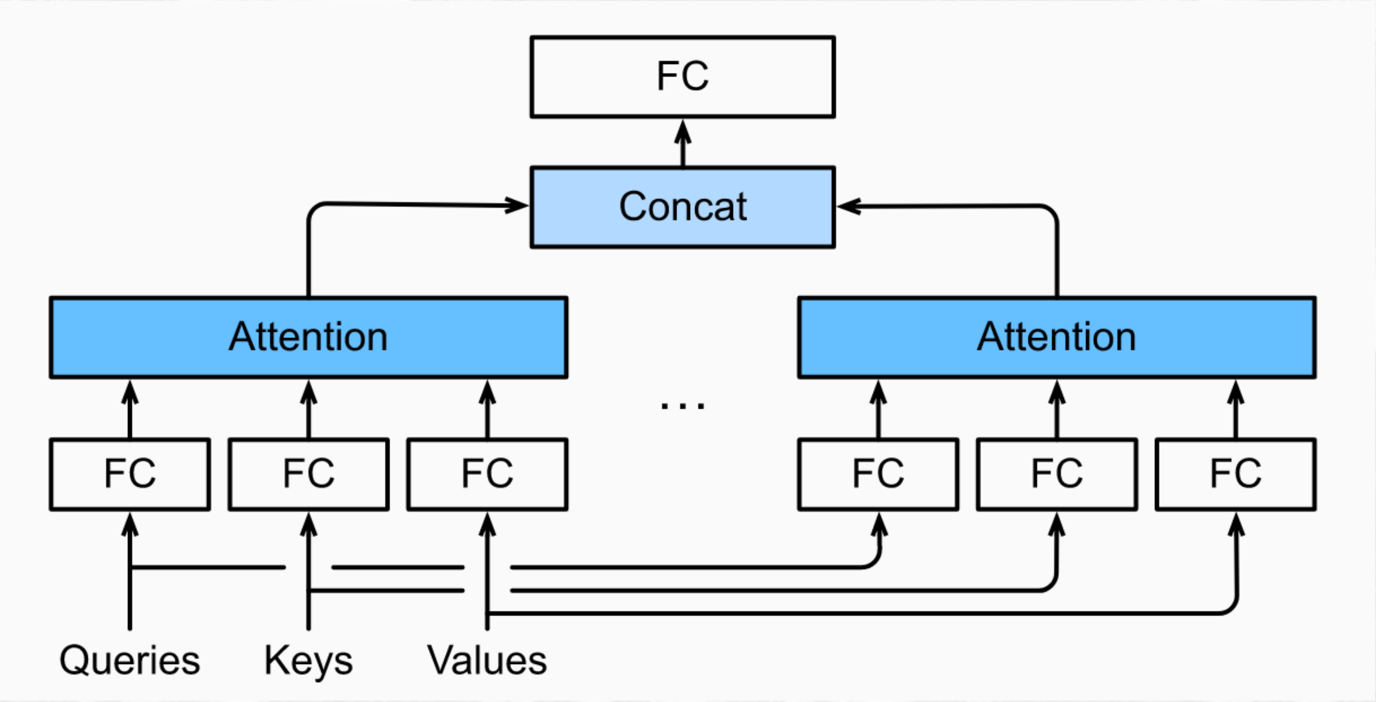
- In a learnable setting, the query and the key are concatenated and fed into an MLP with a single hidden layer whose number of hidden units is *h*, a hyperparameter. By using as the activation function and disabling bias terms, we implement additive attention in the following

Attention functions: scaled dot-product (2)

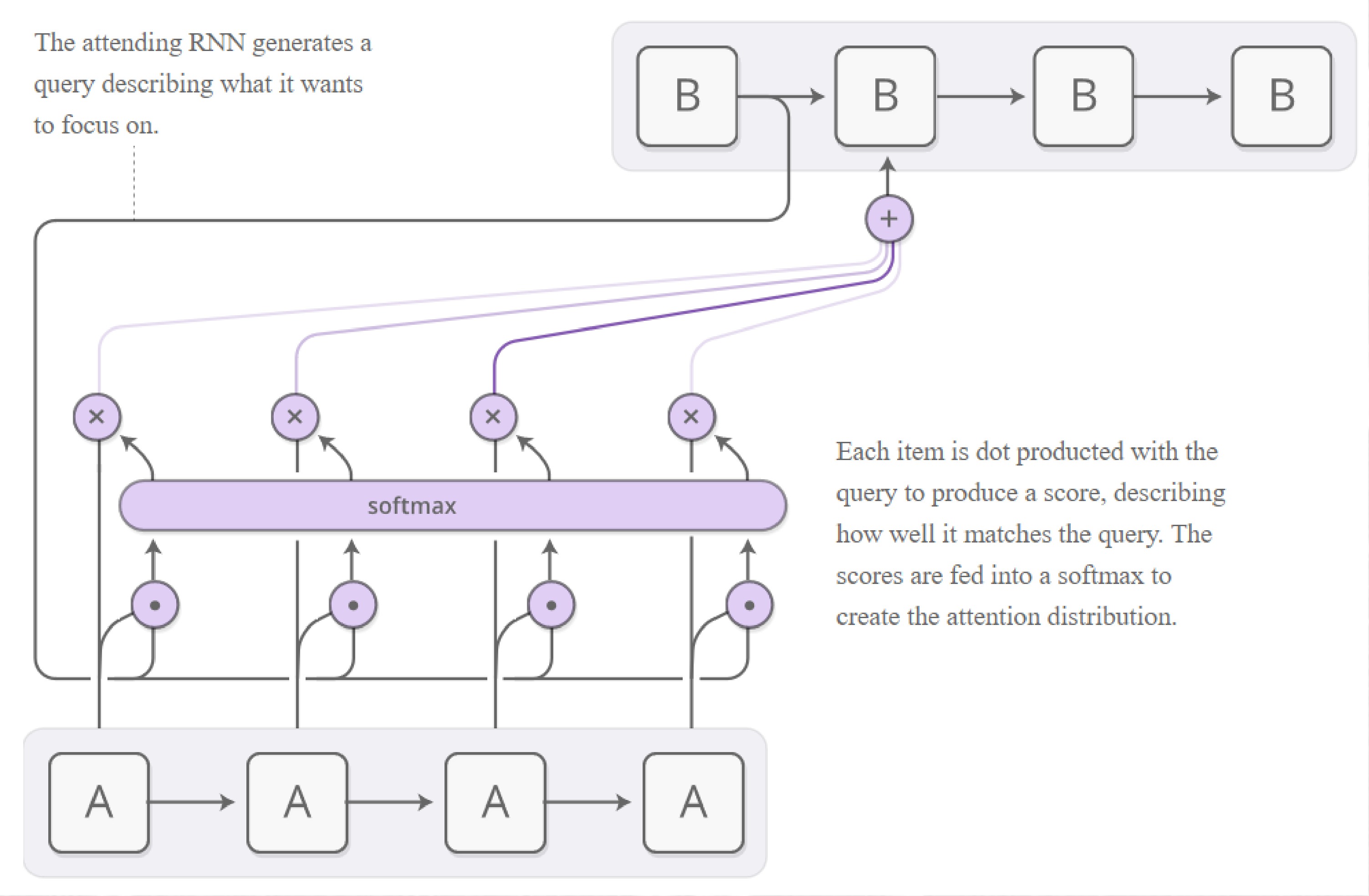
- When *q* and *k* are *d*-dimensional vectors whose independent dimensions have mean=0 and variance=1, their dot product has mean = 0 and a variance = *d*. To ensure that the variance of the dot product still remains one regardless of vector length, the *scaled dot-product attention* scoring function is adopted $a(\mathbf{q}, \mathbf{k}) = \mathbf{q}^\top \mathbf{k}/\sqrt{d}$
- It divides the dot product by $\sqrt{d}$. In practice, we often think in minibatches for efficiency, such as computing attention for *n* queries and *m* key-value pairs, where queries and keys are of length *d* and values are of length *v*. The scaled dot-product attention of queries $\mathbf{Q} \in \mathbb{R}^{n\times d}$, keys $\mathbf{K} \in \mathbb{R}^{m\times d}$, and values $\mathbf{V} \in \mathbb{R}^{m\times v}$ is

$$\mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right)\mathbf{V} \in \mathbb{R}^{n\times v}.$$

Attention: multihead



Attention-based RNNs



Attention mechanisms in Machine Translation

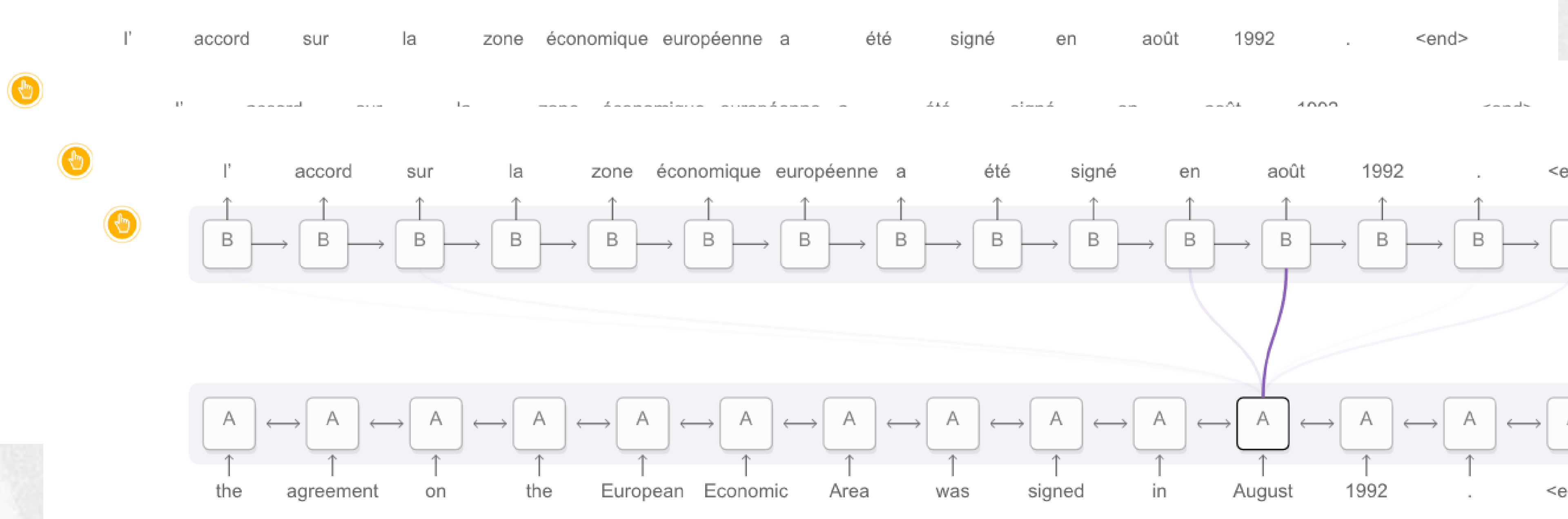
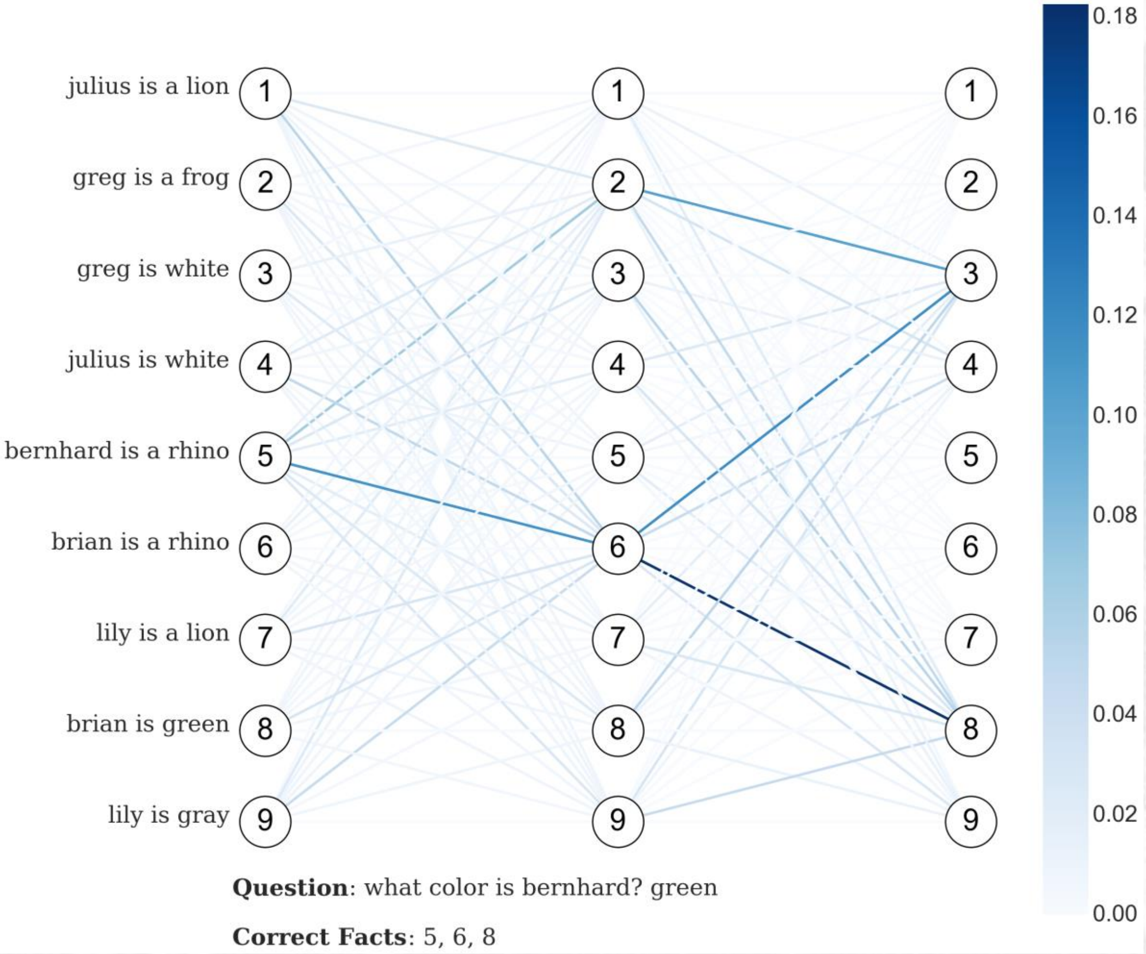


Diagram derived from Fig. 3 of Bahdanau, *et al.* 2014

Visualization of the attention distribution in QA

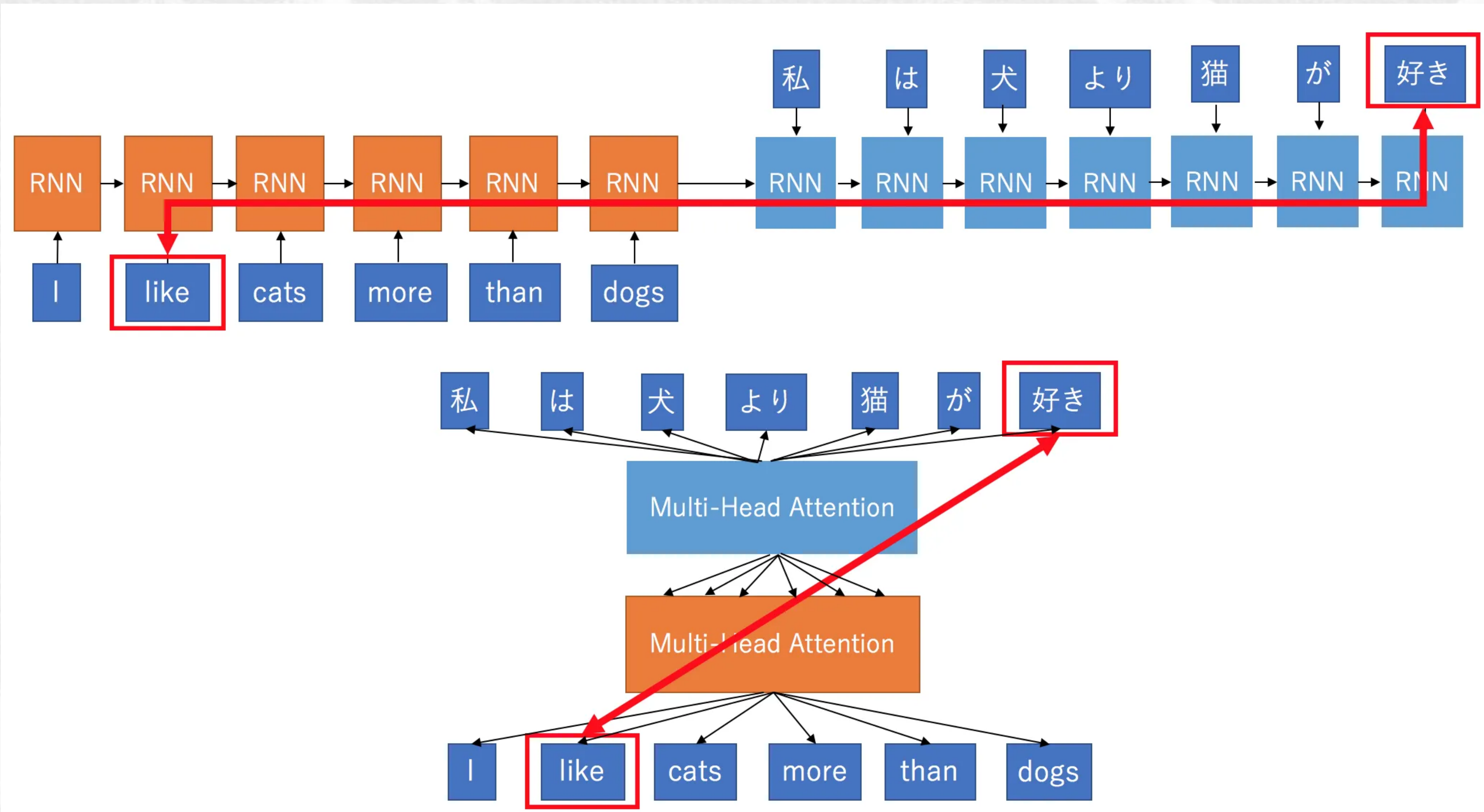
- Supporting fact sequences for an example question
- On the right the attentions over facts for individual sequences
  - Each sequence is mapped into a Markov process



Attention & enconding

- IN a decoding process (e.g. machine translation) there are **three** kinds of dependencies for neural architectures
- Dependencies can establish between
- (1) the ***input and output*** tokens
- (2) the ***input tokens themselves***
- (3) the ***output tokens themselves***
- Examples:
  - MT
  - QA where the query the answer paragraph is the input and the matched answer is the output

Attention in MT:
long distance dependencies



From RNNs to Transformers

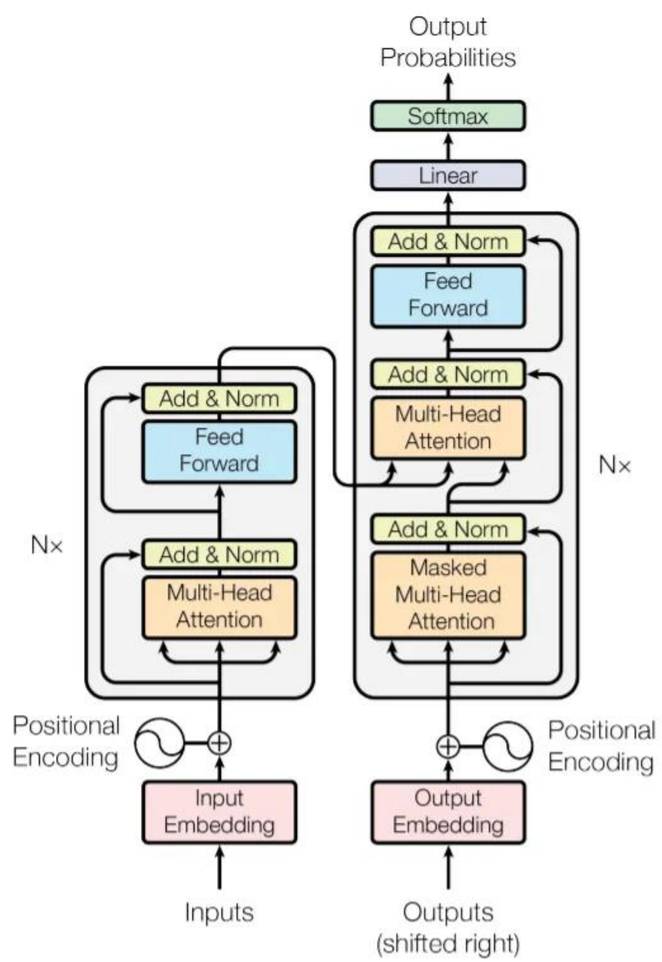
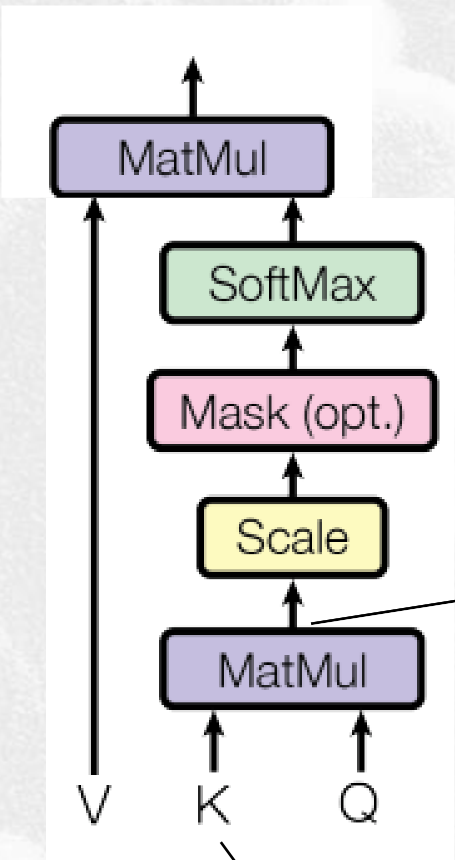


Figure 1: The Transformer - model architecture.

Bidirectional Encoder Representations from **BERT** - Transformers (Devlin et al. '18)

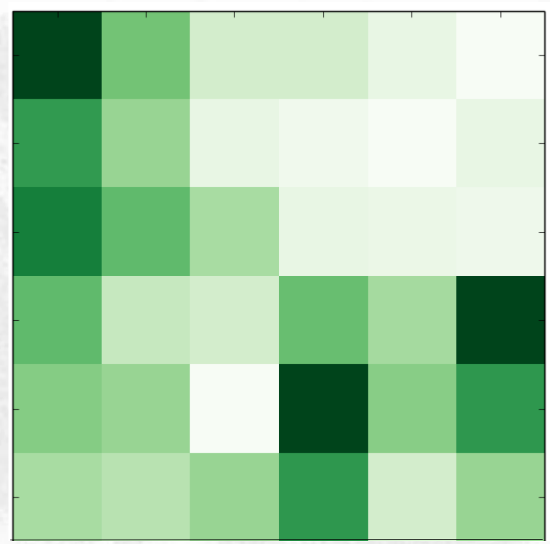
Scaled Dot-Product Attention

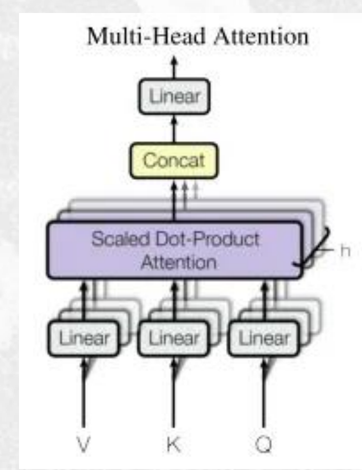


$L \times d_e$

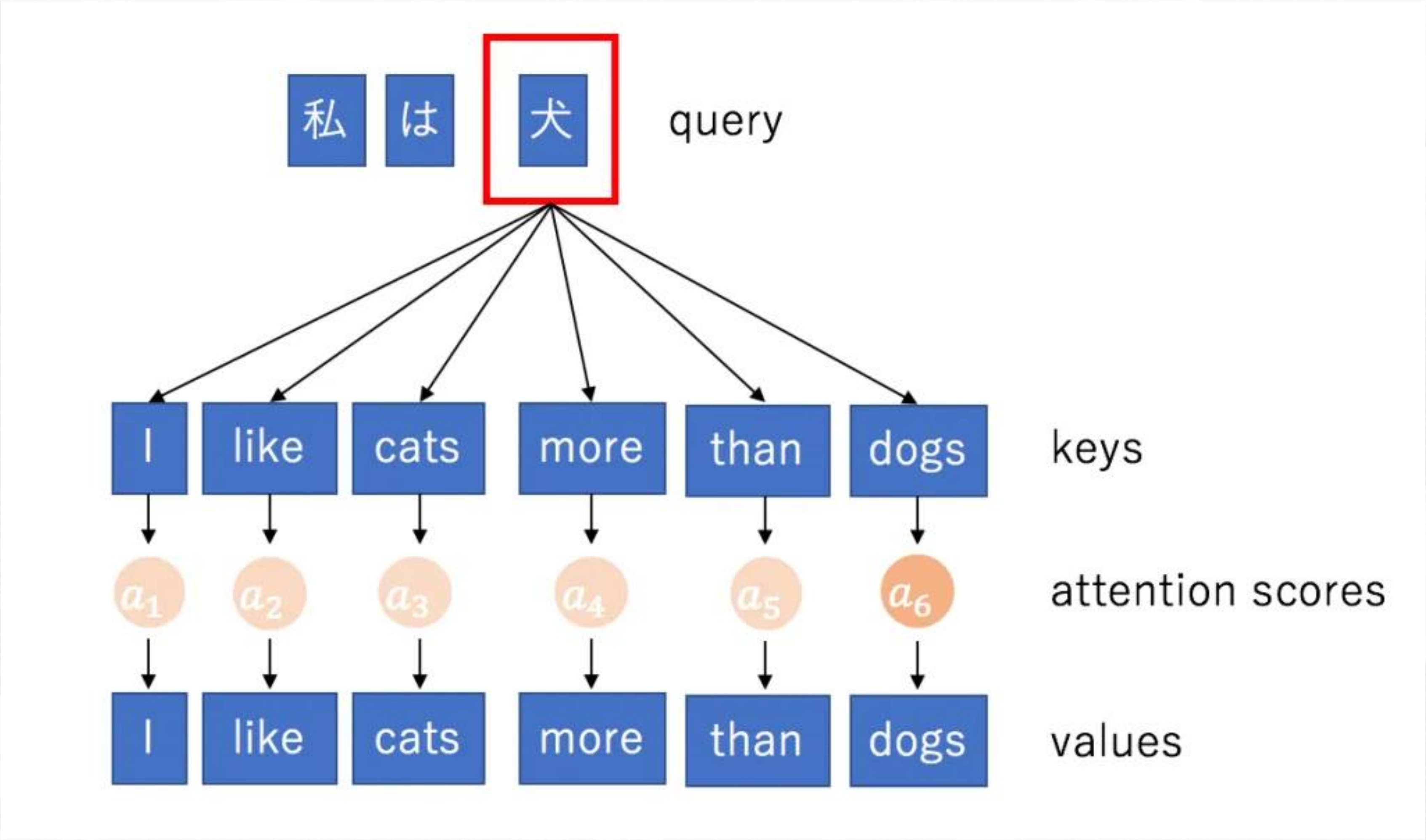
Attention is a function that maps a query Q and a set of key-value pairs <K,V> to an output

L x L



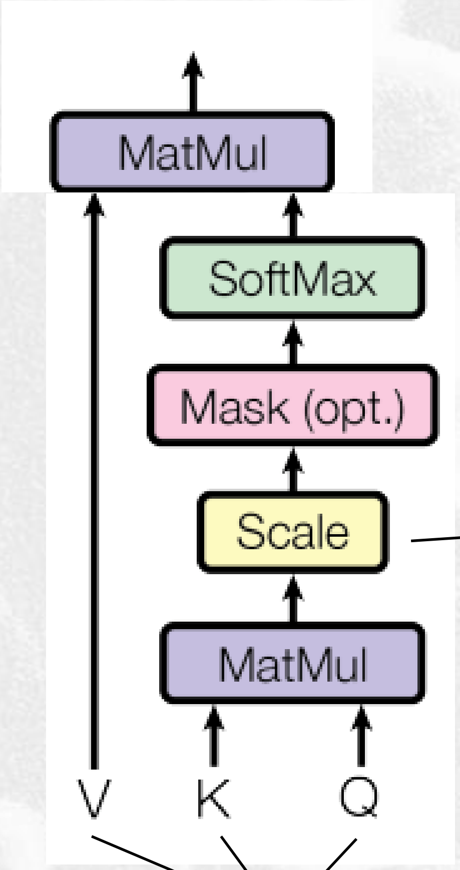


Input-Output Attention



BERT (Devlin et al. '18)

Scaled Dot-Product Attention

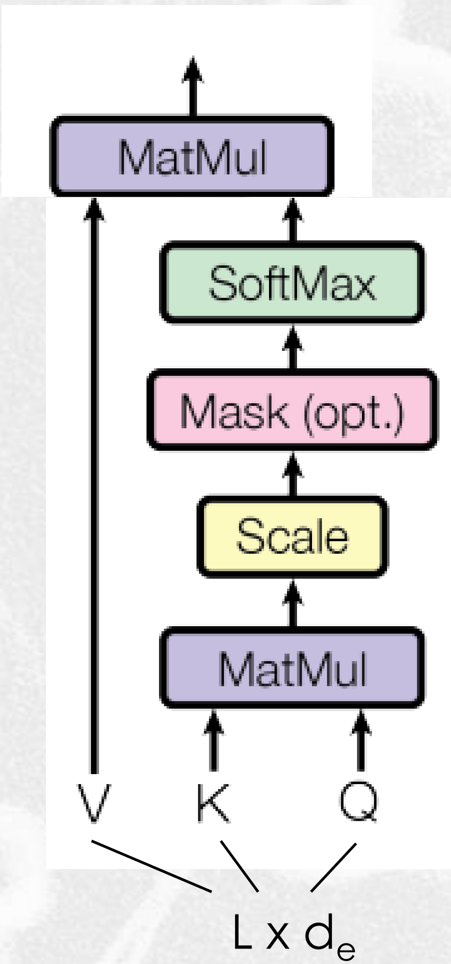


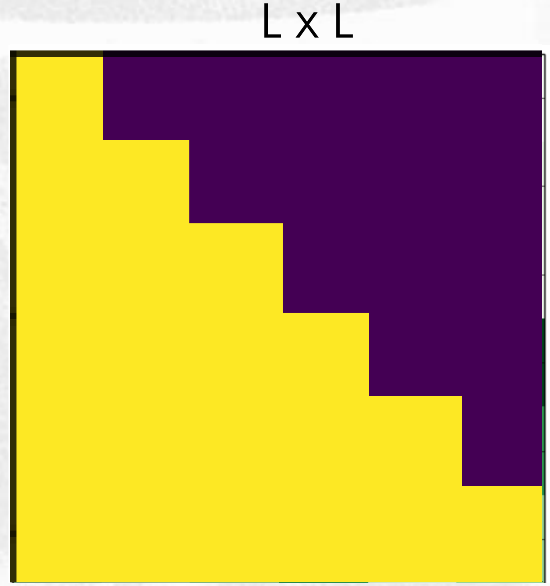
$L \times d_e$

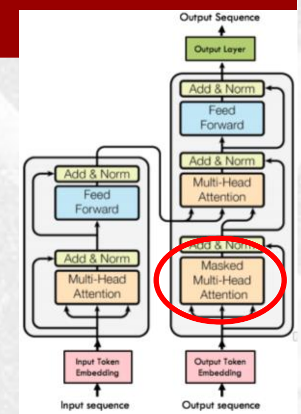
Division by $\sqrt{d_e}$
Only for numerical stability

BERT (Devlin et al. '18)

Scaled Dot-Product Attention

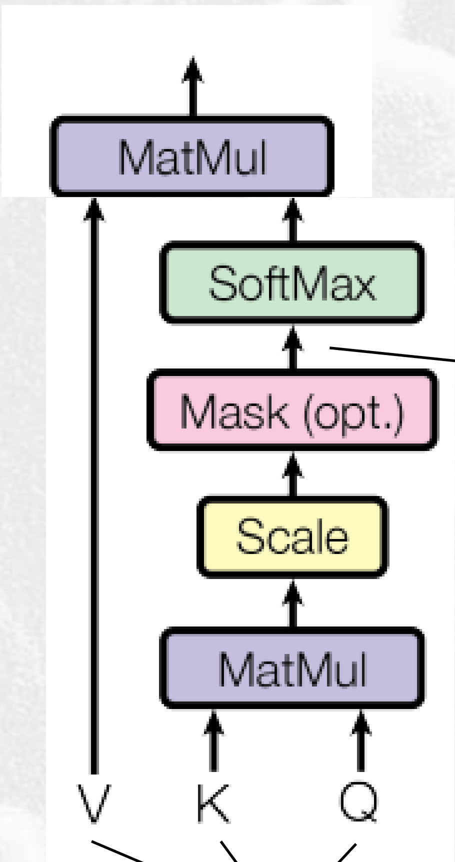






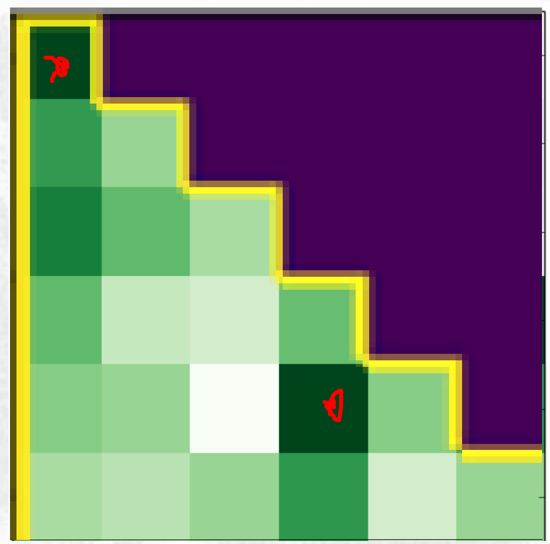
BERT (Devlin et al. ’18)

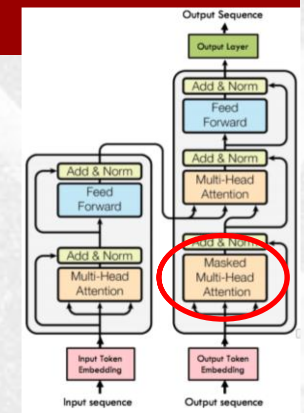
Scaled Dot-Product Attention



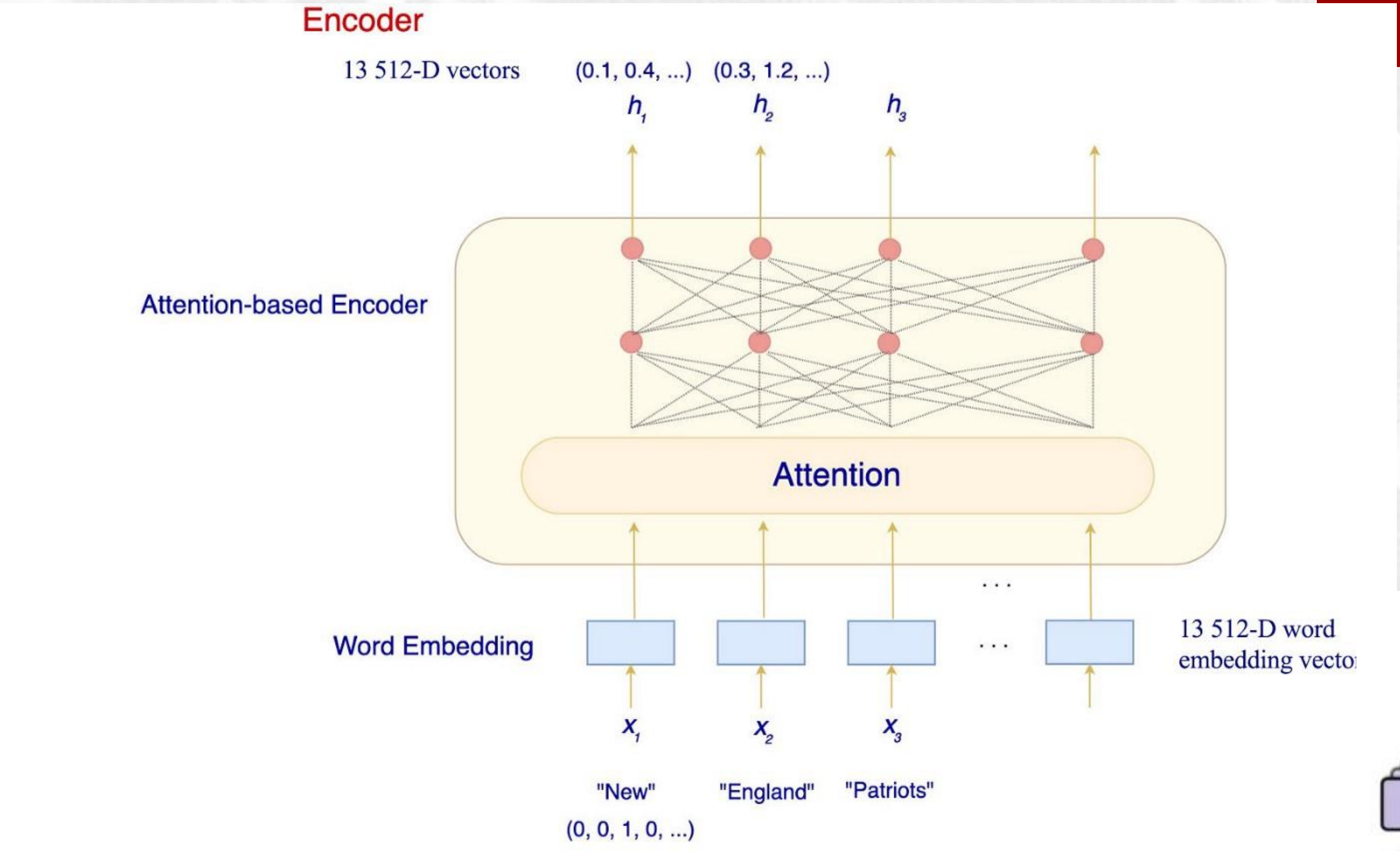
L x d_e

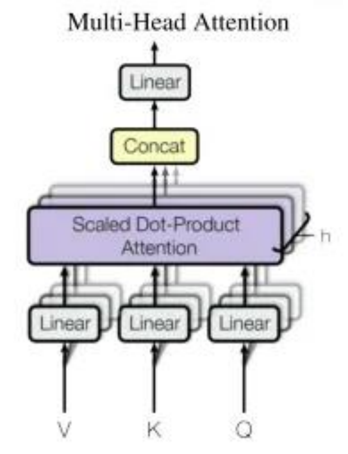
L x L





BERT & NLP

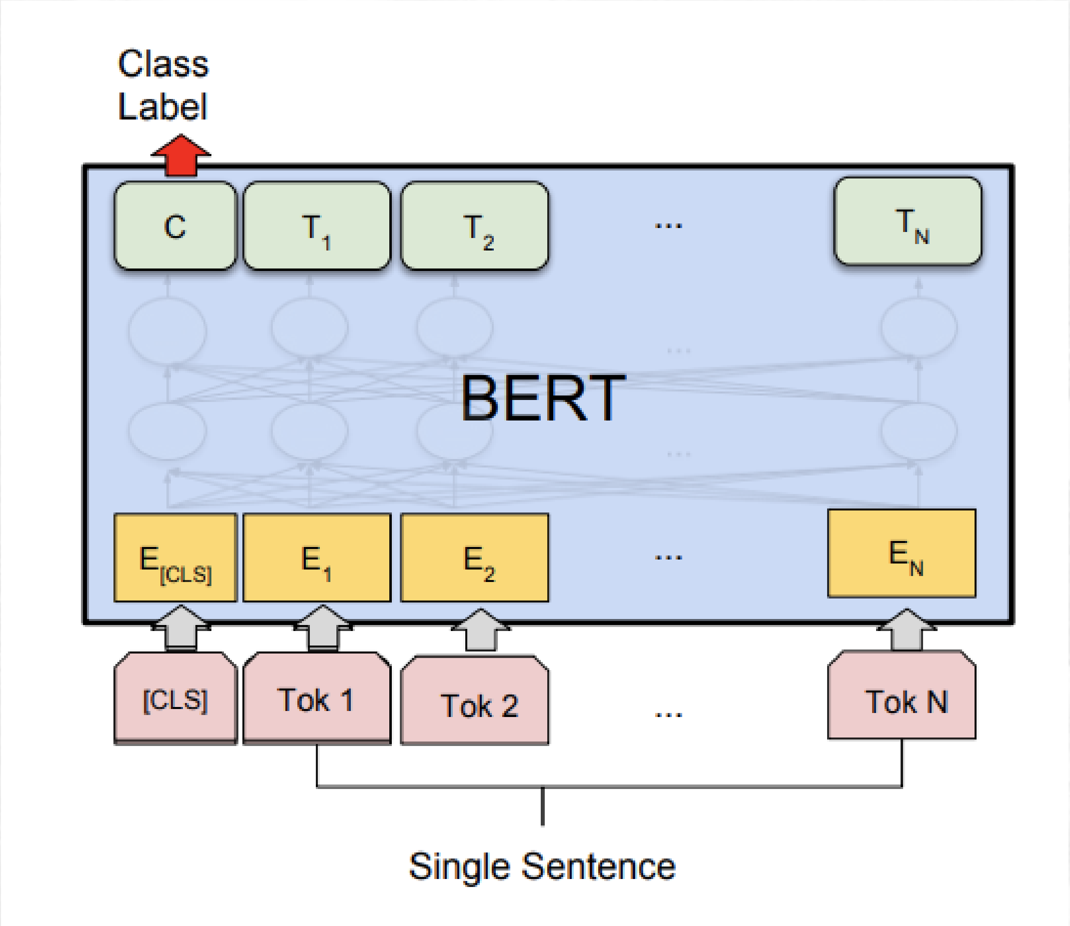

Encoder
13 512-D vectors
(0.1, 0.4, ...)
(0.3, 1.2, ...)
h1
h2
h3
Attention-based Encoder
Attention
...
Word Embedding
13 512-D word
embedding vecto
x1
x2
x3
"New"
"England"
"Patriots"
(0, 0, 1, 0, ...)


Multi-Head Attention
Linear
Concat
Scaled Dot-Product
Attention
h
Linear
Linear
Linear
V
K
Q

BERT & NLP (2)

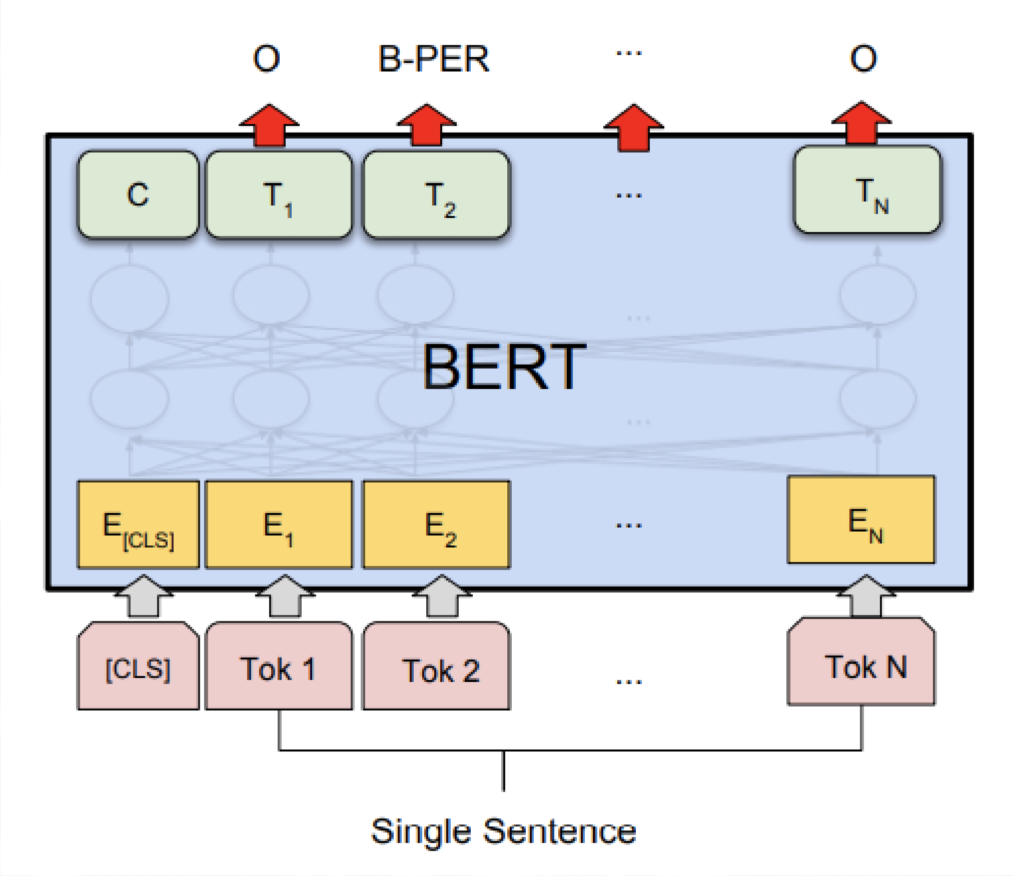
- How to optimize the encoding?
- General and complex tasks defined in (Devlin et al., 2018) are
  - Masked Language Modeling (15%)
    - Inpired by Distributional Hypothesis
    - Can be Simulated and does not require any labeling
  - Next Sentence Prediction
    - Inspired by Textual Inference tasks (e.g. Textual Entailment)
    - Can be Simulated and does not require any labeling
- Source Representations
  - Words? And why not subword (in the BERT jargon: word pieces)?
    - Useful to deal with out-of-vocabulary phenomena

BERT (Devlin et al. '18)



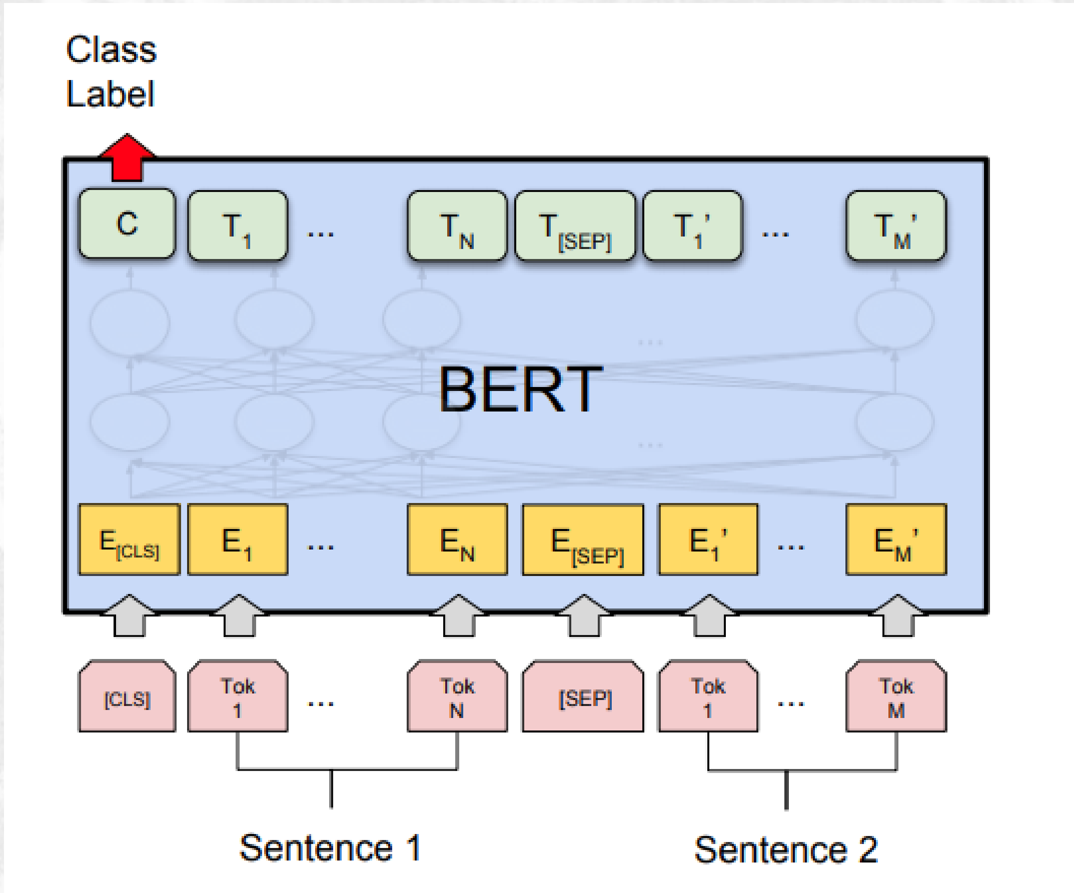
BERT for single sentence classification (Sentiment analysis, Intent Classification, etc.)

BERT (Devlin et al. '18)



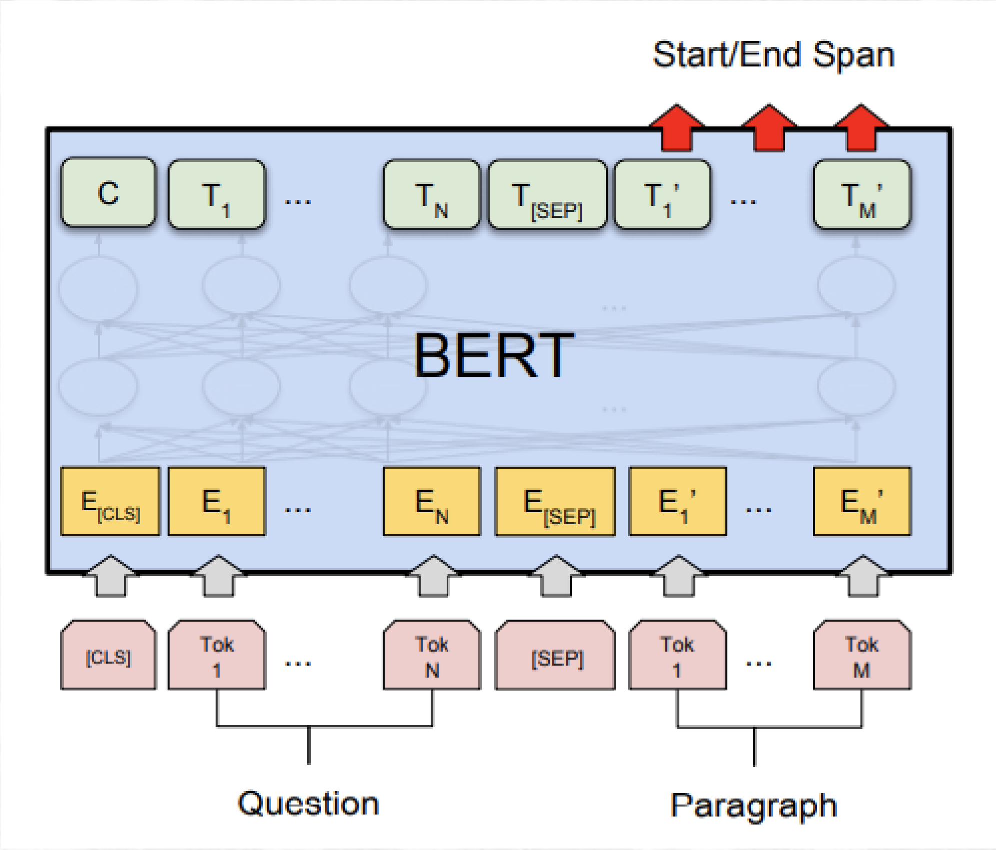
BERT for Sequence Tagging Tasks (e.g., POS tagging, Named Entity Recognition, etc.)

BERT (Devlin et al. '18)



BERT for sentence pairs classification (Paraphrase Identification, answer selection in QA, Recognizing Textual Entailment)

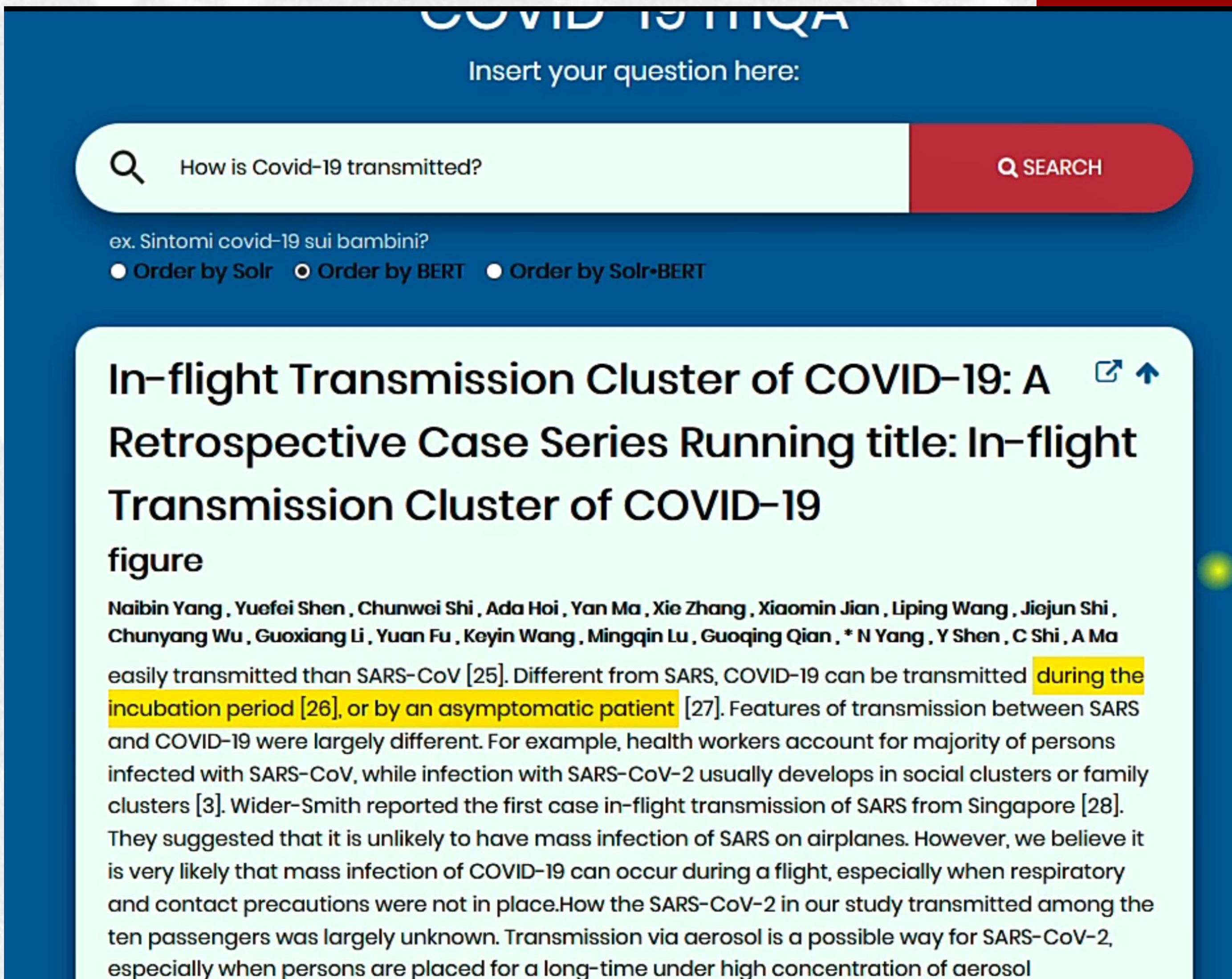
BERT (Devlin et al. '18)



BERT for Answer Span Selection in Question Answering

A QA example on SquAD

- Cross-lingual Question Answering



BERT (Devlin et al. '18)

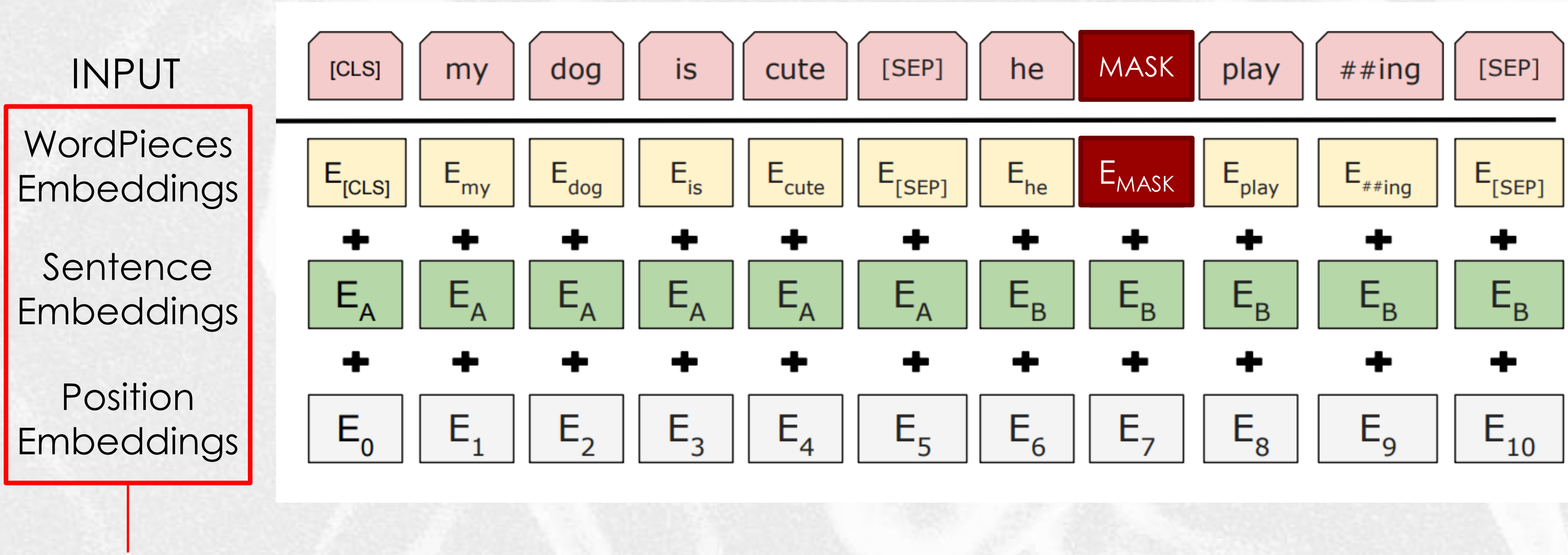
Pretraining on two unsupervised prediction tasks:

- **Masked Language Model**: given a sentence *s* with missing words, reconstruct *s*
  - Example: Amazon <MASK> amazing → Amazon is amazing
  - In BERT the language modeling is deeply Bidirectional, while in ELMo the forward and backward LMs were two independent branches of the NN

- **Next Sentence Prediction**: given two sentences s_1 and s_2, the task is to understand whether s_2 is the actual sentence that follows s_1
  - 50% of the training data are positive examples: s_1 and s_2 are actually consecutive sentences
  - 50% of the training data are negative examples: s_1 and s_2 are randomly chosen from the corpus

BERT pretraining:

Input representations



All these embeddings are learned during the (pre)training process

In pre-training 15% of the input tokens are masked for the masked LM task

Attention mechanisms in Speech Recognition

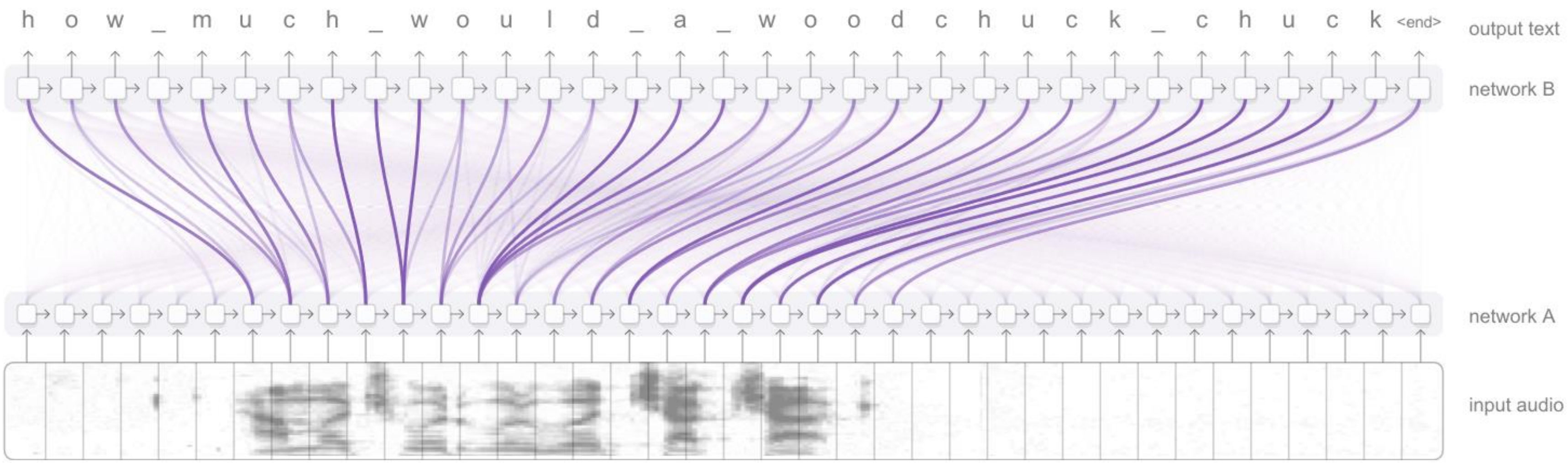


Figure derived from Chan, *et al.* 2015

https://arxiv.org/pdf/1508.01211.pdf

A complex application of LSTM (and recently Transformers): Image captioning

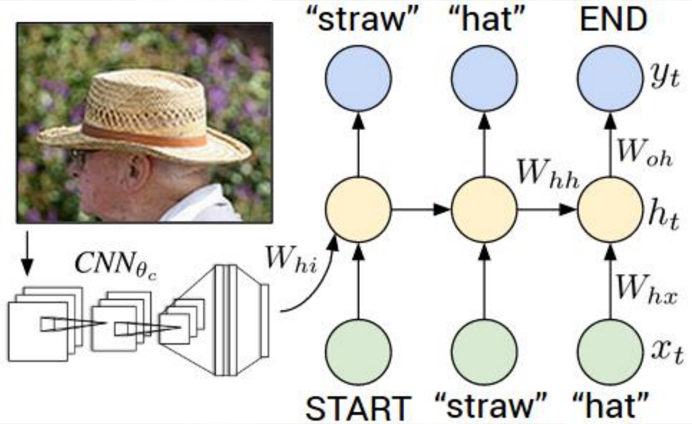
A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

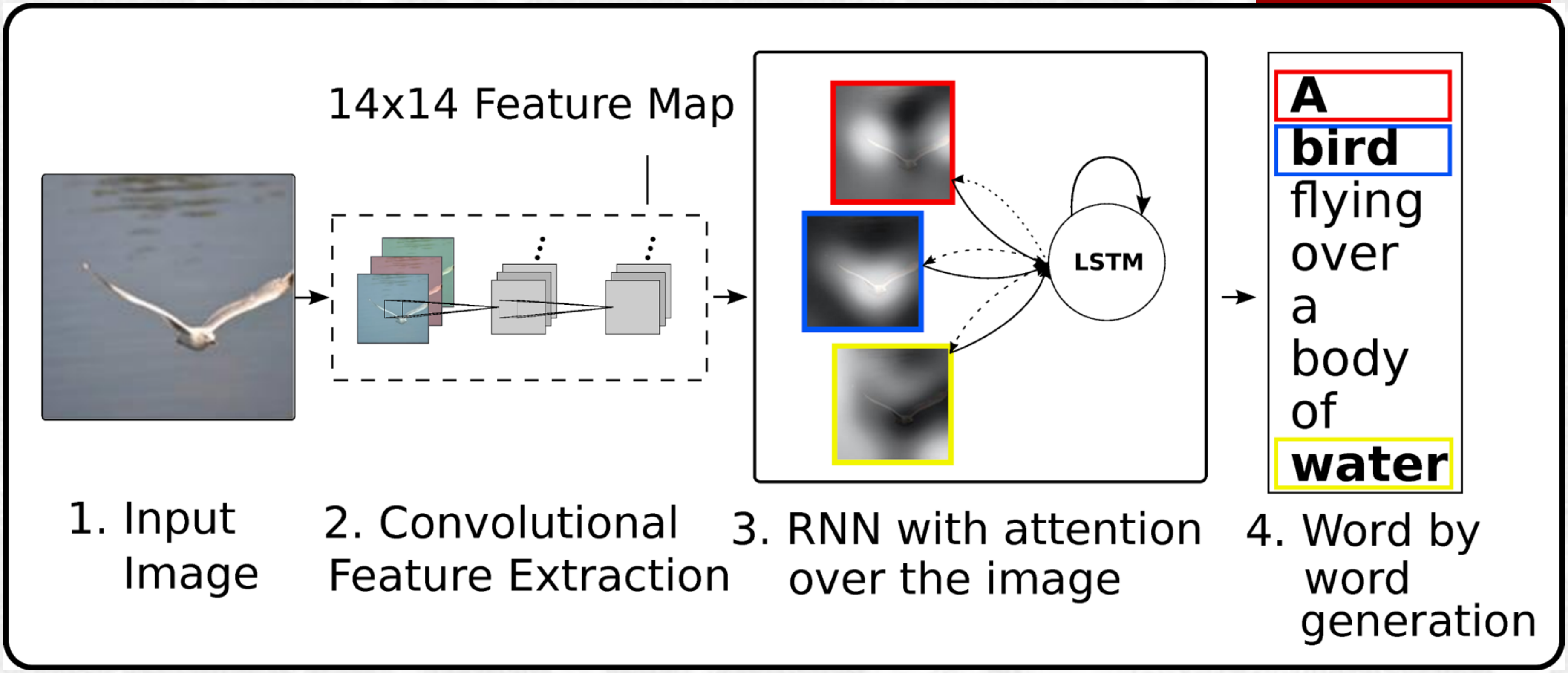
A stop sign is on a road with a mountain in the background.

Image Captioning

- Image to captions
  - Convolutional Neural Network to learn a representation of the image
  - (Bi-directional) Recurrent Neural Network to generate a caption describing the image
    - its input is the representation computed from the CNN
    - its output is a sequence of words, i.e. the caption



"baseball player is throwing ball in game."

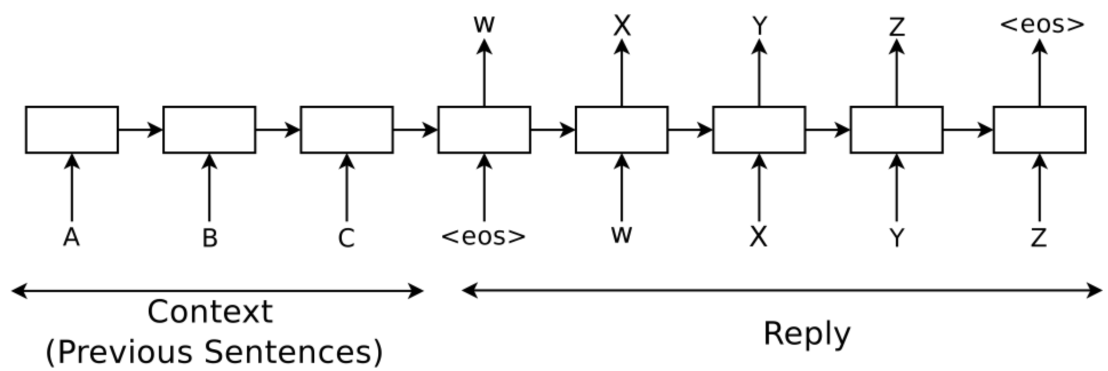

14x14 Feature Map
LSTM
A
bird
flying
over
a
body
of
water
1. Input Image
2. Convolutional Feature Extraction
3. RNN with attention over the image
4. Word by word generation

Attention: a dynamic rendering

RNNs: Neural Conversation Model

- Learn to answer question in a conversation
- For example, an IT help desk
  - model the input sequence
  - produce a output sequence

Machine: *hi*
Human: *hello*
Machine: *could you please let me know what are the operating systems you are using ?*
Human: *i am using linux*
Machine: *ok it was in the ssl vpn on which you stated it was broken*
Machine: *please check if you are logged in to corp network*
Machine: *what is the error that you are running please*
Human: *i am seeing an error related to vpn*
Machine: *what is the error message that you are getting when connecting to vpn using network connect ?*
Human: *connection refused or something like that*



Perspectives

- Injecting bias (e.g. linguistic structures) within the learning architectures
- Making use of hybrid architectures integrating visual and linguistic knowledge
- Extending the epistemological transparency of current architectures: Explainable AI
- Making natural language data to work as a representation layer for different cognitive functions (e.g HRI in robotics but also vision)

Transformers

- (Vaswani 2017), Attention is all you need, https://arxiv.org/abs/1706.03762
- (Devlin et al 2018), BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, https://arxiv.org/abs/1810.04805
- An interesting introduction to the attention mechanism:
  - **All you need to know about 'Attention' and 'Transformers' — In-depth Understanding — Part 1**, A. Sarkar, URL: https://towardsdatascience.com/all-you-need-to-know-about-attention-and-transformers-in-depth-understanding-part-1-552f0b41d021#4c16
- Other Task specific works:
  - Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. CoRR, abs/1409.0473, 2014.
  - Effective Approaches to Attention-based Neural Machine Translation, Minh-Thang Luong Hieu Pham Christopher D. Manning, 2015, https://arxiv.org/abs/1508.04025v5
  - Yoon Kim, Carl Denton, Luong Hoang, and Alexander M. Rush. Structured attention networks. In International Conference on Learning Representations, 2017.