From Shallow to Deep Language Representations: Pre-training, Fine-tuning, and Beyond

5. Transformer

Leonard Lausen

gluon-nlp.mxnet.io

aws

Encoder-Decoder

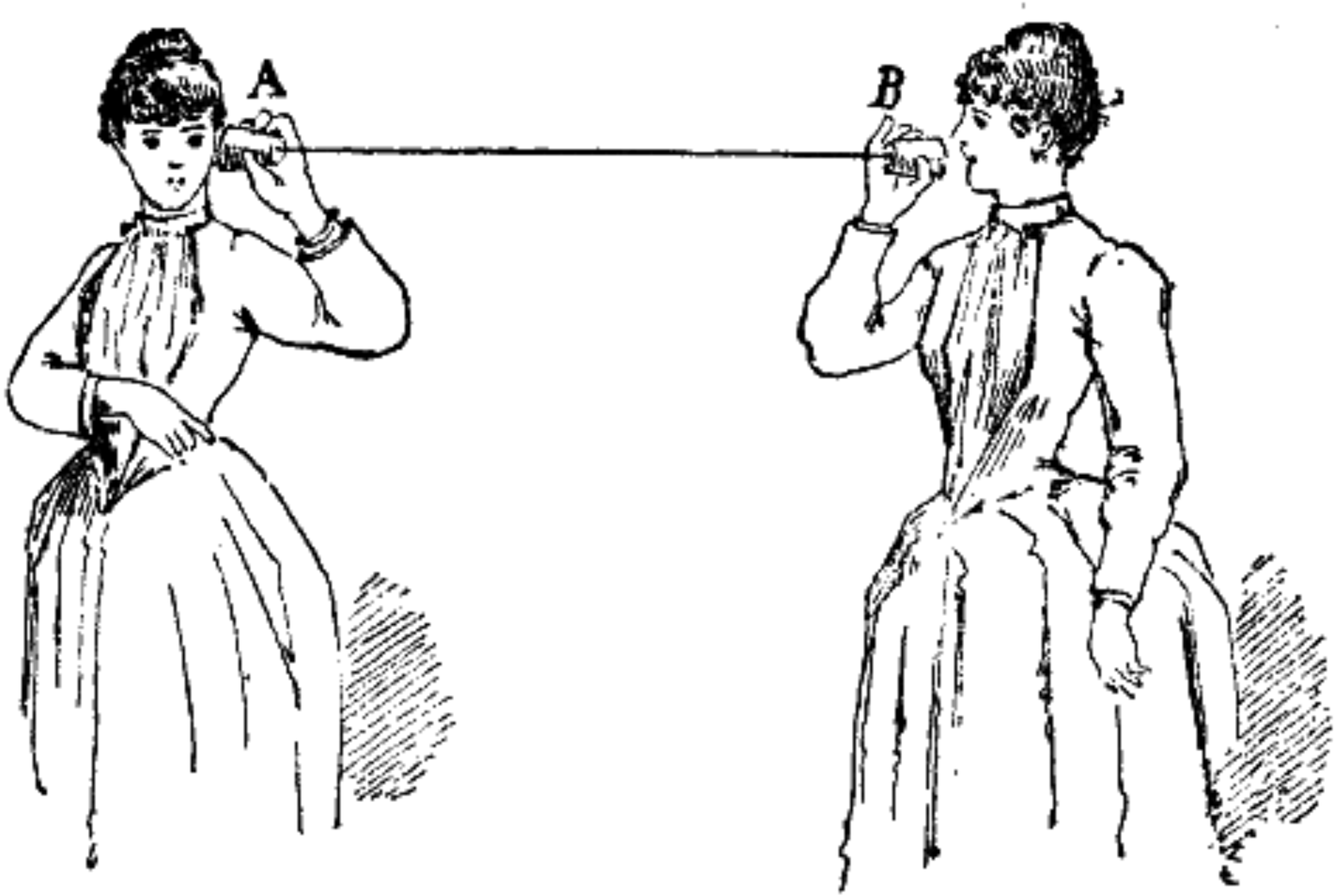
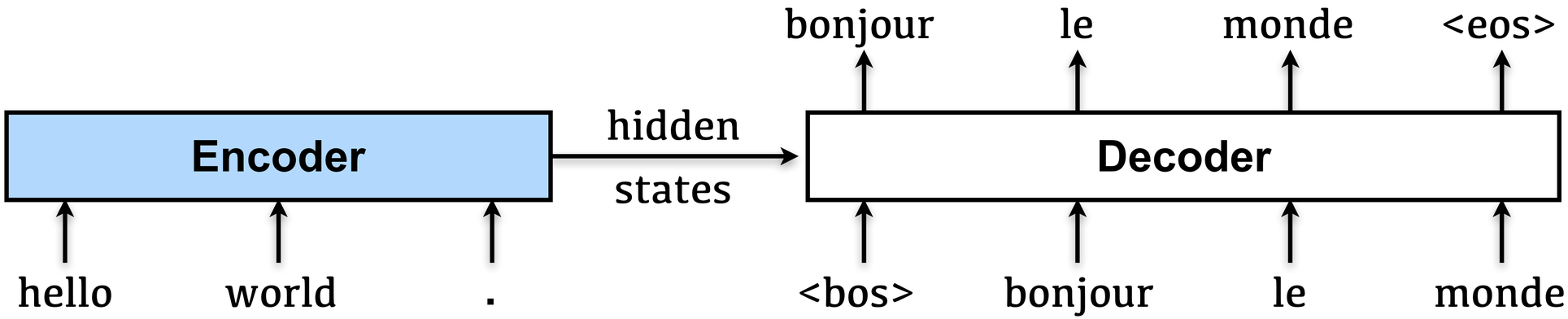


Image Source: https://en.wikipedia.org/wiki/Tin_can_telephone

Encoder-Decoder for Machine Translation

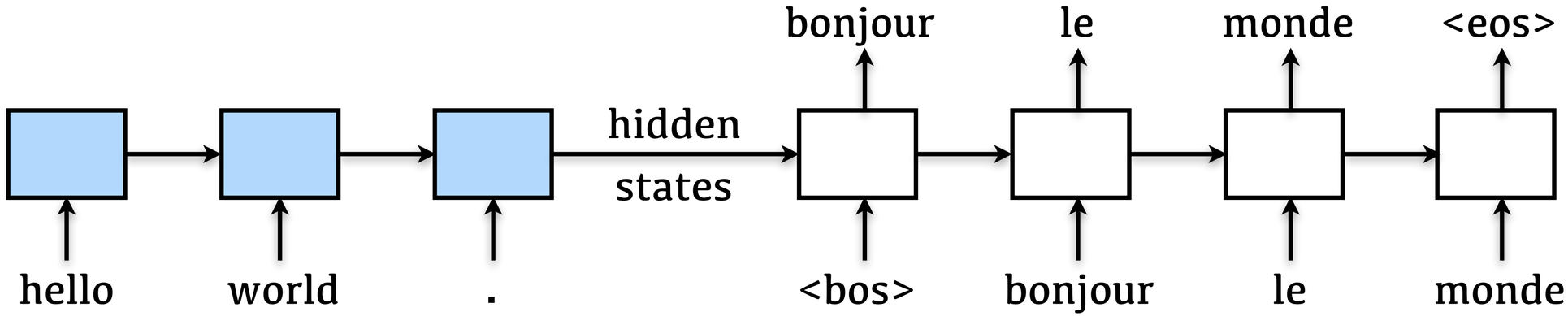


h = Encoder(‘hello’, ‘world’, ‘.’)

‘le’ = Decoder(‘bonjour’, ‘<bos>’, h)

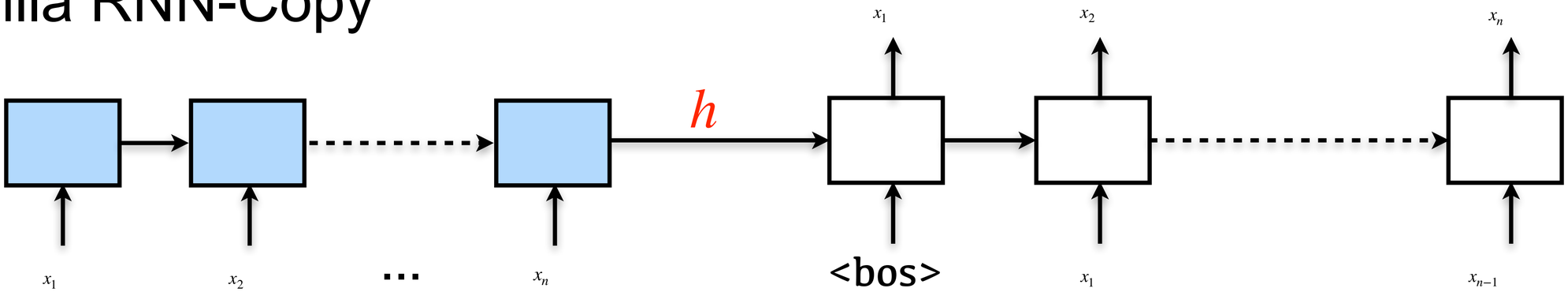
Difficult to generate a sequence. Decoders are modeled to be single-step-ahead.

RNN as Encoder & Decoder



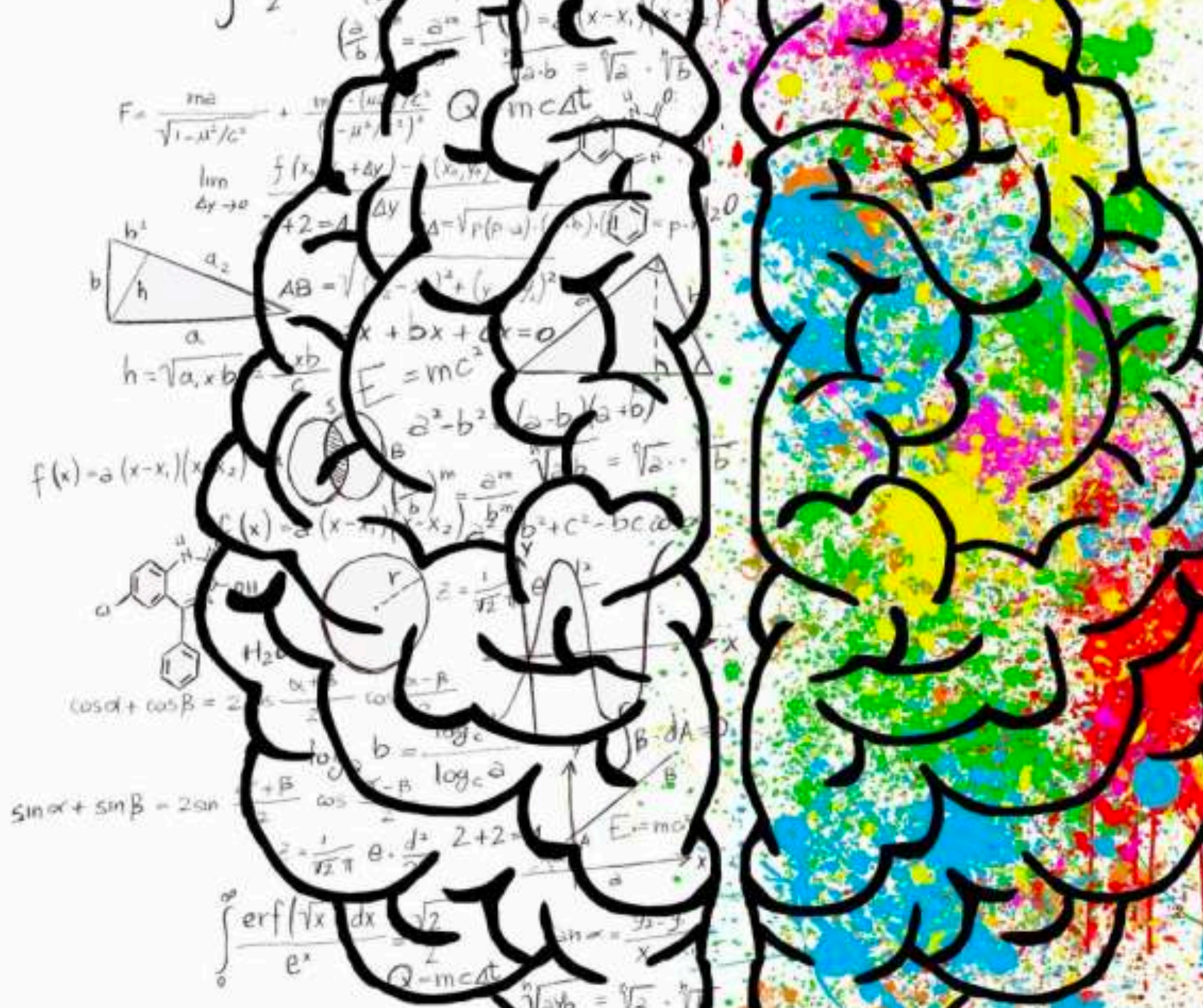
The Limitation of RNN Encoder-Decoder

- Mental Experiment: Sequence Copy
  - $x_1, x_2, \ldots, x_n \rightarrow h \rightarrow x_1, x_2, \ldots, x_n$
- Vanilla RNN-Copy



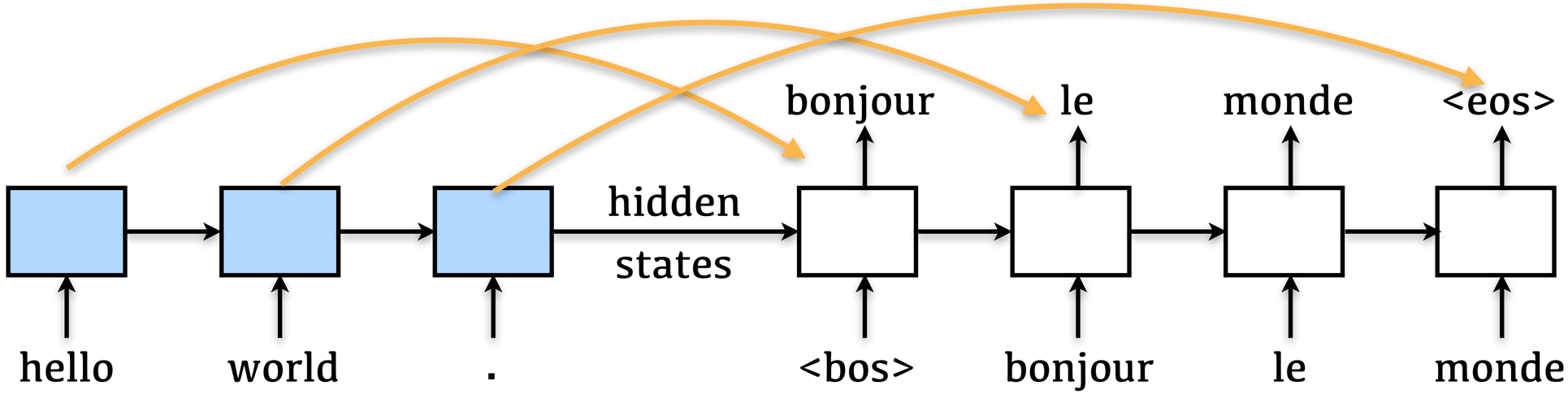
- What if the sequence length approaches infinity?
  - Encoding to a fixed-dimensional vector will have information loss
  - Can never solve the problem due to the memory capacity

Attention



Motivation

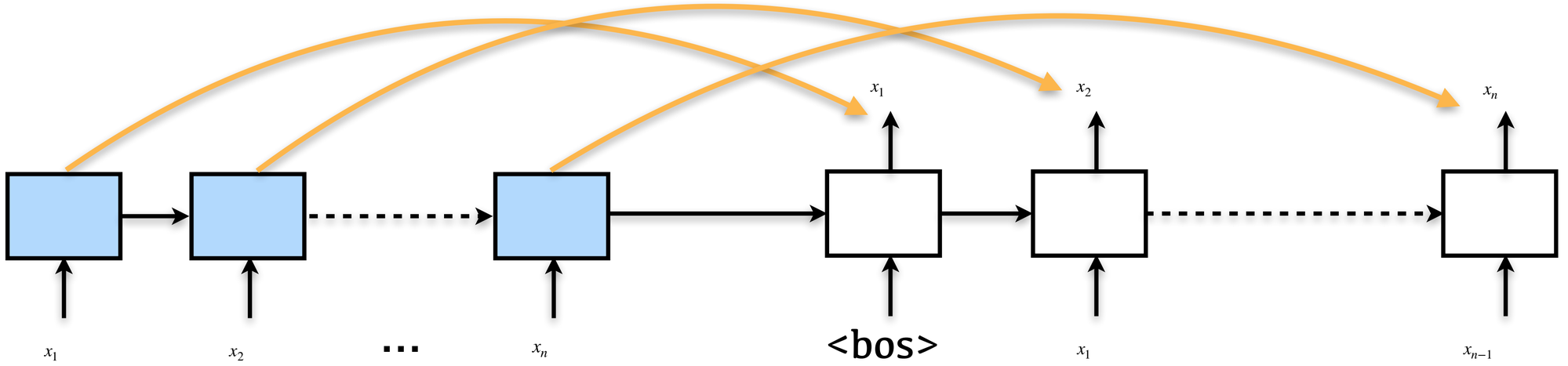
- Each generated token might be related to some specific source tokens.



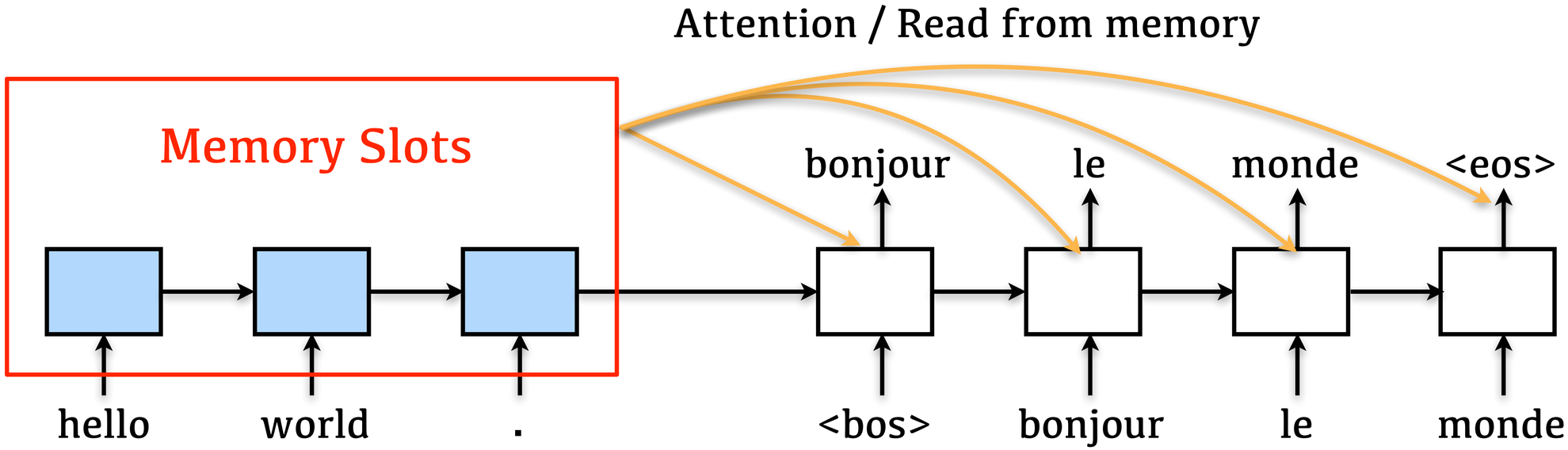
- Instead of just using the last state, all states are used in the decoding process. Search for related states output by encoder.

Sequence Copy by Attention

- Sequence Copy is possible with attention
  - Why? All states in the encoder are kept — Memory
- $x_1, x_2, \ldots, x_n \rightarrow h_1, h_2, \ldots, h_n \rightarrow x_1, x_2, \ldots, x_n$



Attention & Memory



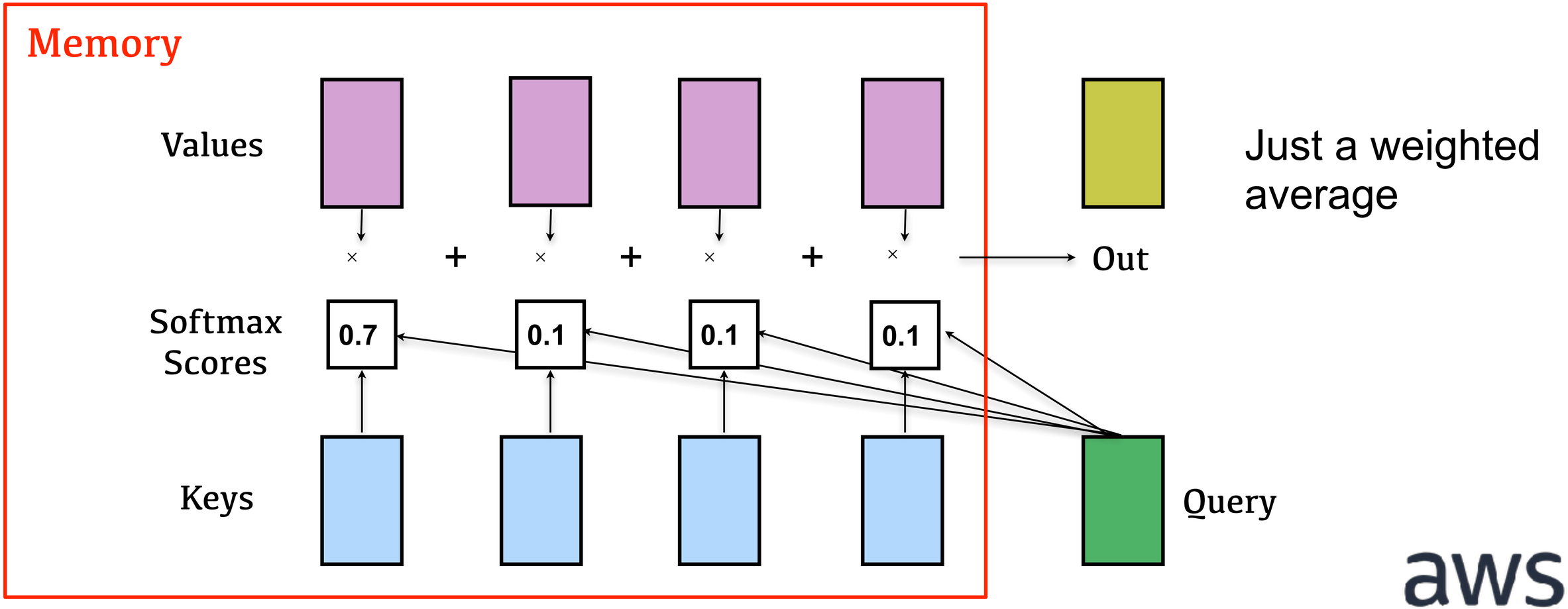
Treat each state in the RNN as memory slots

In each decoding step, the network will search the states that are related to the query.

Interpretation — Key Value Memory Network

Out = Attention(Query, Key, Value)

Out = Attention(h_i^{dec}, $f(H^{enc})$, $g(H^{enc})$)



[Sukhbaatar et al. 2015] End-to-end Memory Networks

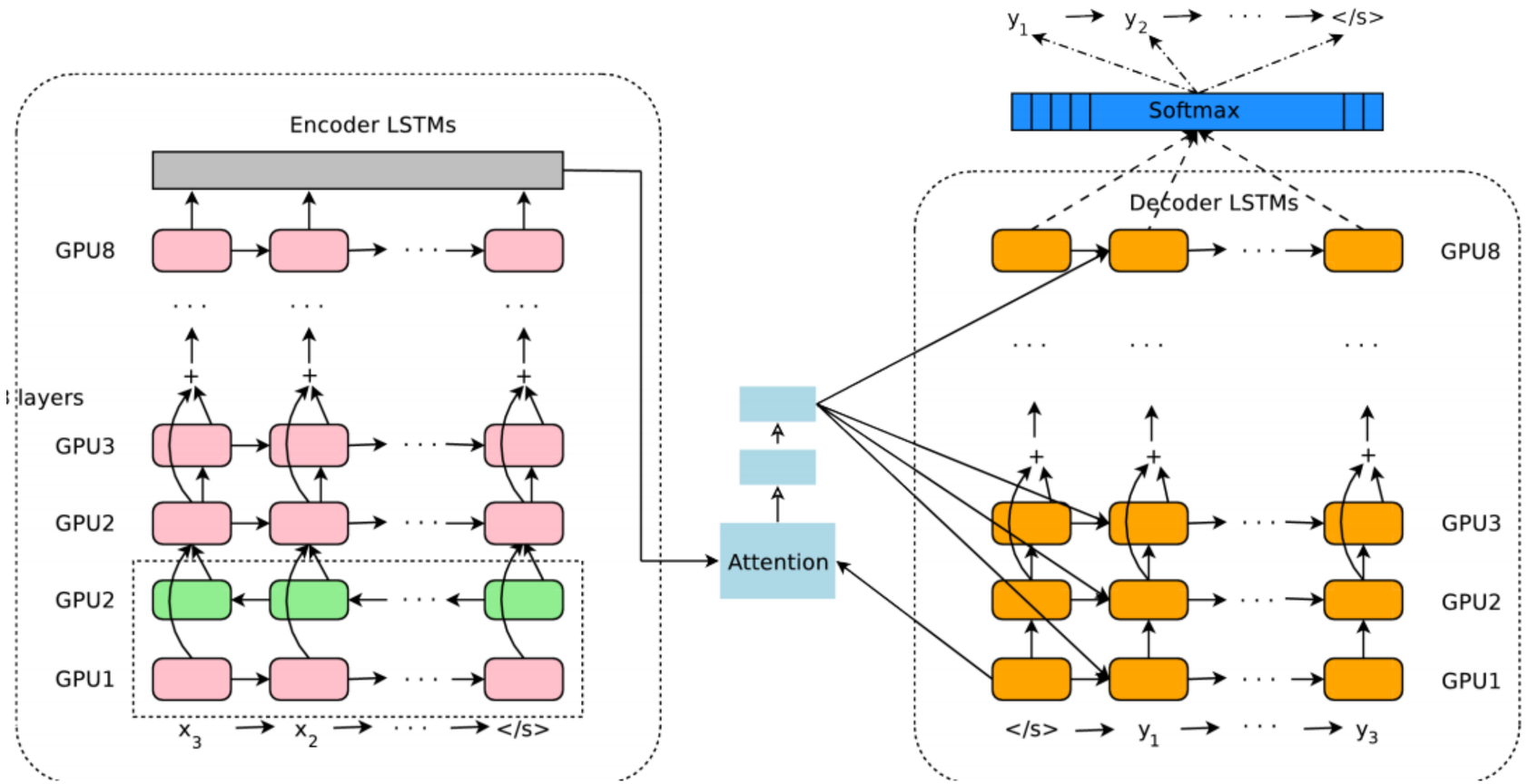
Implement Different Attention Scores in GluonNLP

- Query: q, Keys: $k_{1:n}$

gluonnlp.model.attention_cell

Attention Score Before Softmax	GluonNLP
$s_i = \mathbf{q}^T \mathbf{k}$	DotProductAttentionCell
$s_i = W_o \tanh(W_i[\mathbf{q} \parallel \mathbf{k}] + b)$	MLPAttentionCell

Google Neural Machine Translation



Residual: X + f(X)

Implementation available in GluonNLP

[scripts/machine_translation]

Encoder: Bidirectional LSTM + Residual

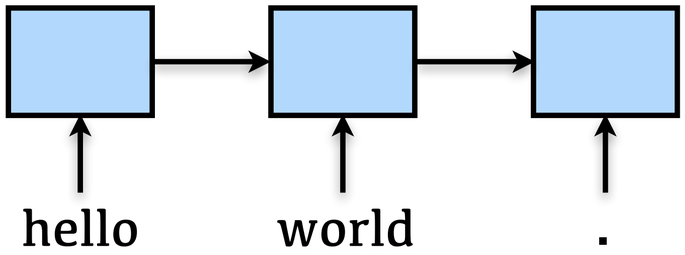
Decoder: LSTM + Residual + Attention

[Wu et al., 2016] Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation

Transformer

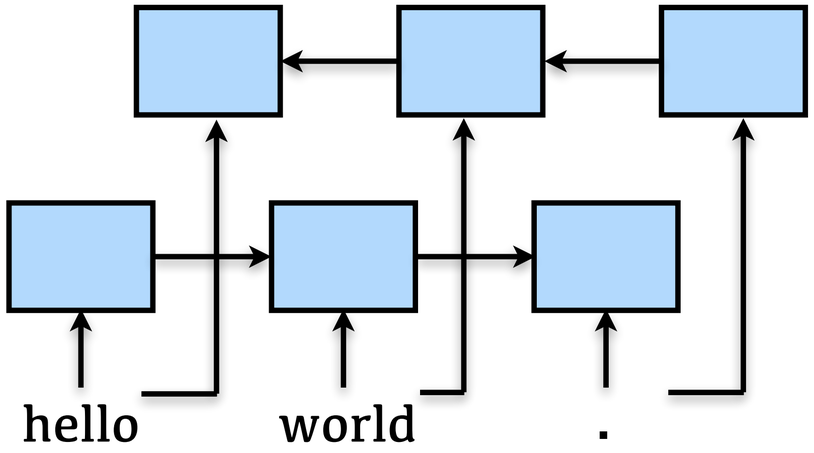
RNN is slow in the encoding phase

RNN



Sequential, Hard to make parallel

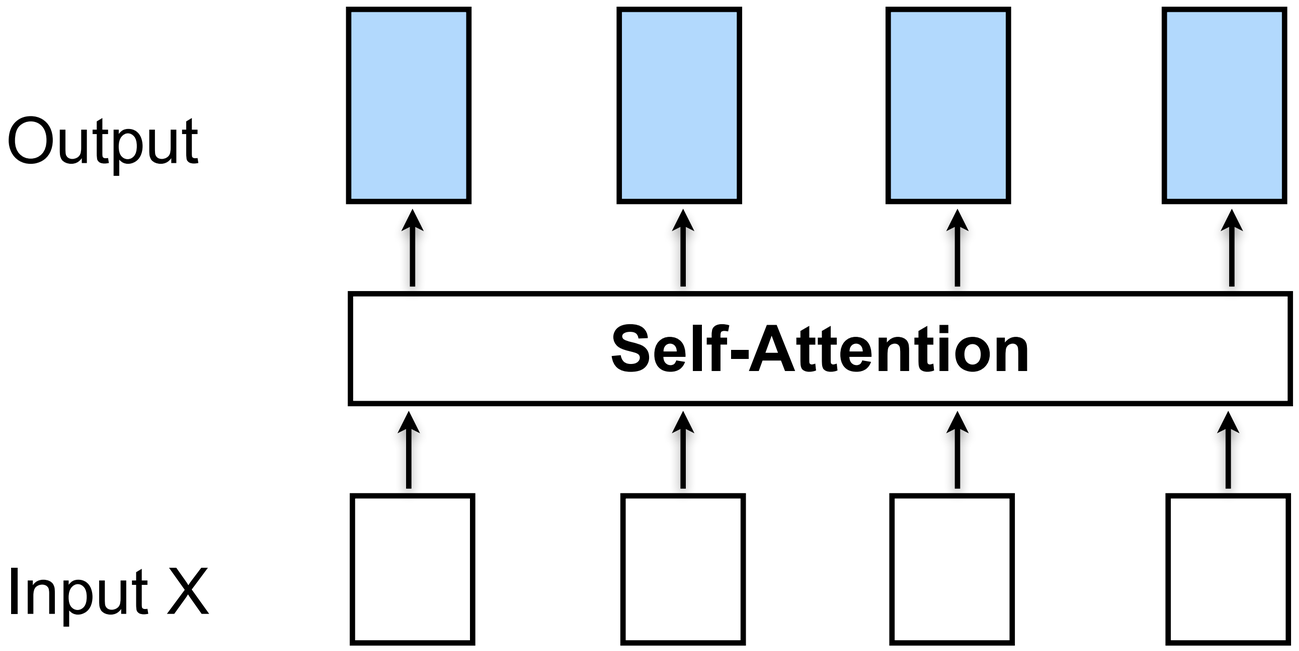
Bidirectional RNN



Self-Attention

$$\text{Out} = \text{Attention}(\text{Query}, \text{Key}, \text{Value})$$

$$\text{Self-Attention: Attention}(W_q X, W_k X, W_v X)$$



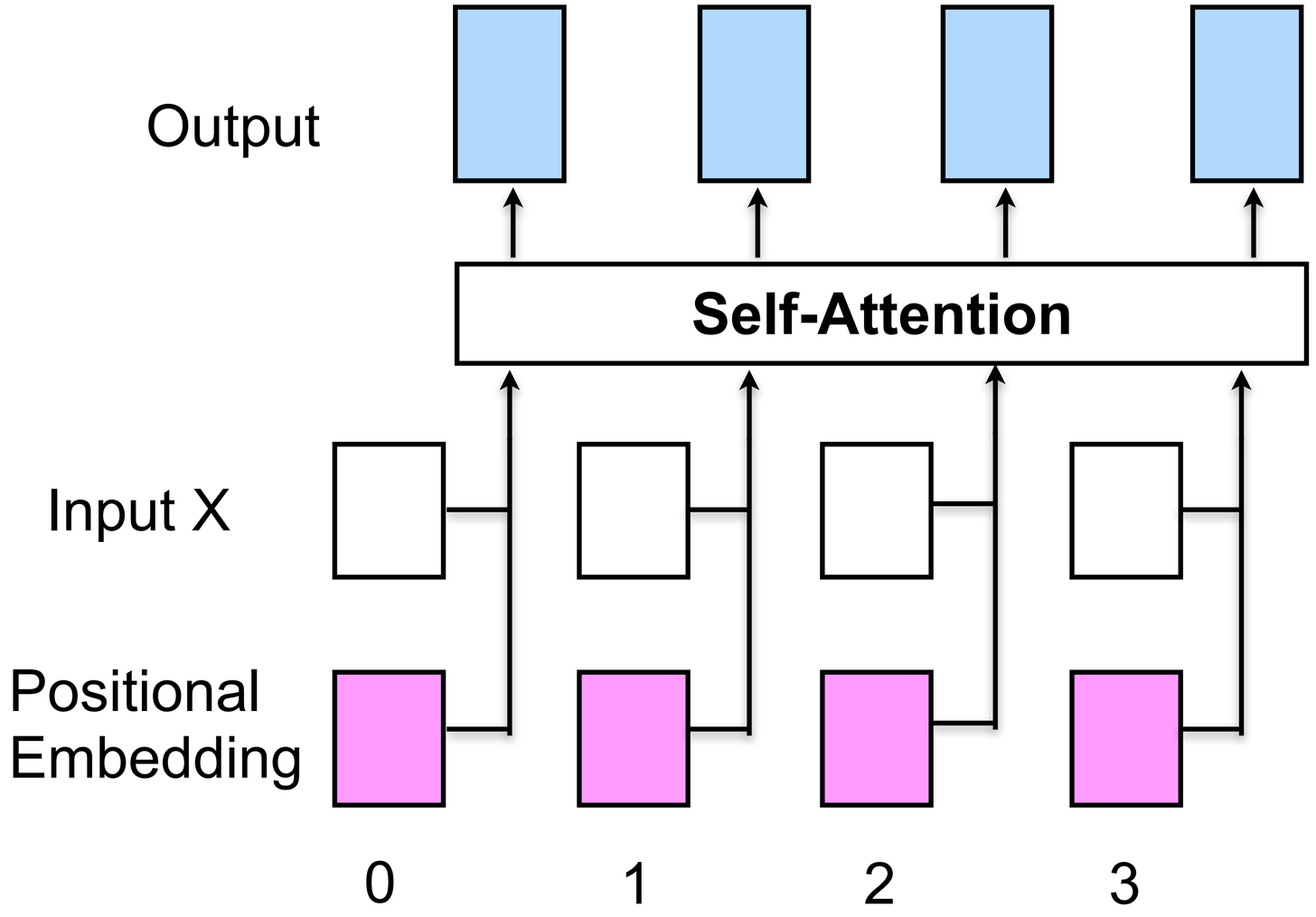
Attention in the encoder.
Run in parallel.

Permutational invariant?

Positional encoding

[Vaswani et al., 2017] Attention is all you need

Positional Encoding



Many options:

- Learn the positional embeddings

- Interpolating a Sinusoidal function

Multi-head Attention

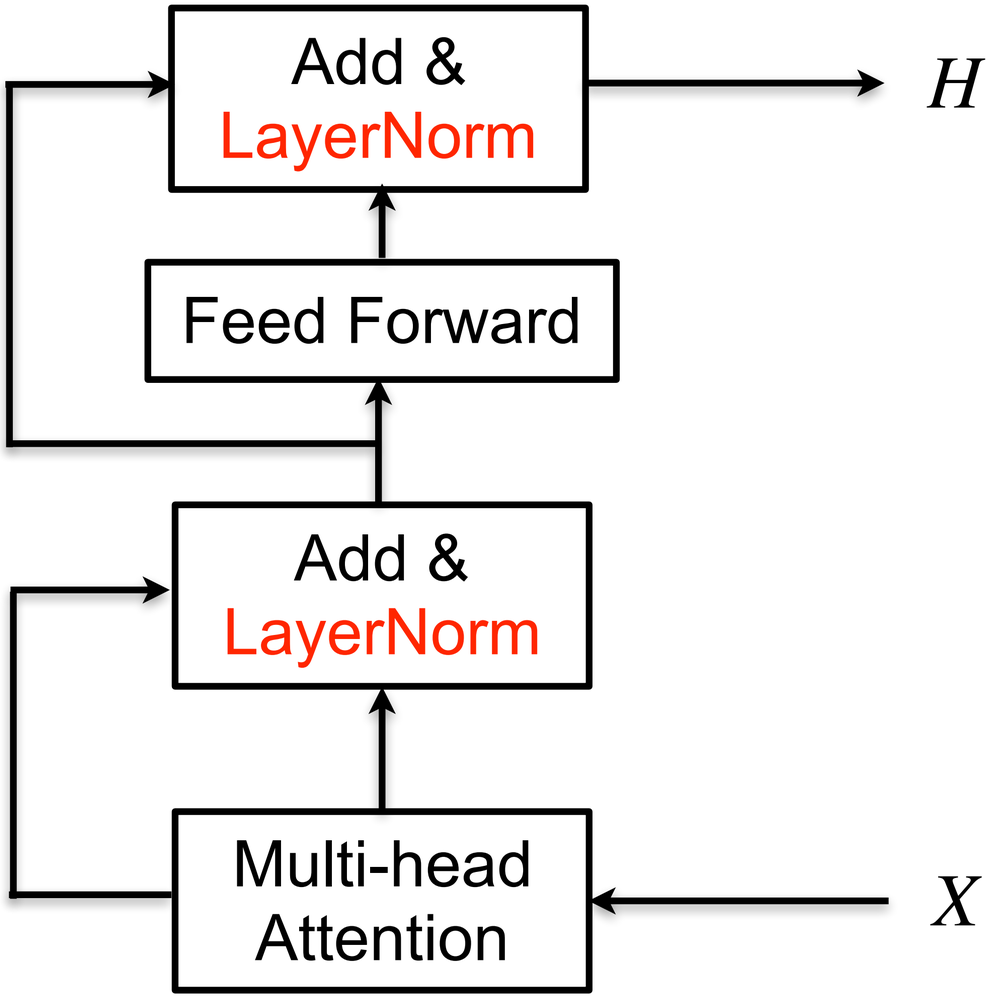
How to increase the representational power?

One-head: Similarity measurement in one subspace

Multi-head: Multiple subspaces!

$$\text{Multi-head Attention:} \quad W_o \begin{bmatrix} \text{Attention}(W_q^1 X, W_k^1 X, W_v^1 X) \\ \text{Attention}(W_q^2 X, W_k^2 X, W_v^2 X) \\ \dots \\ \text{Attention}(W_q^K X, W_k^K X, W_v^K X) \end{bmatrix}$$

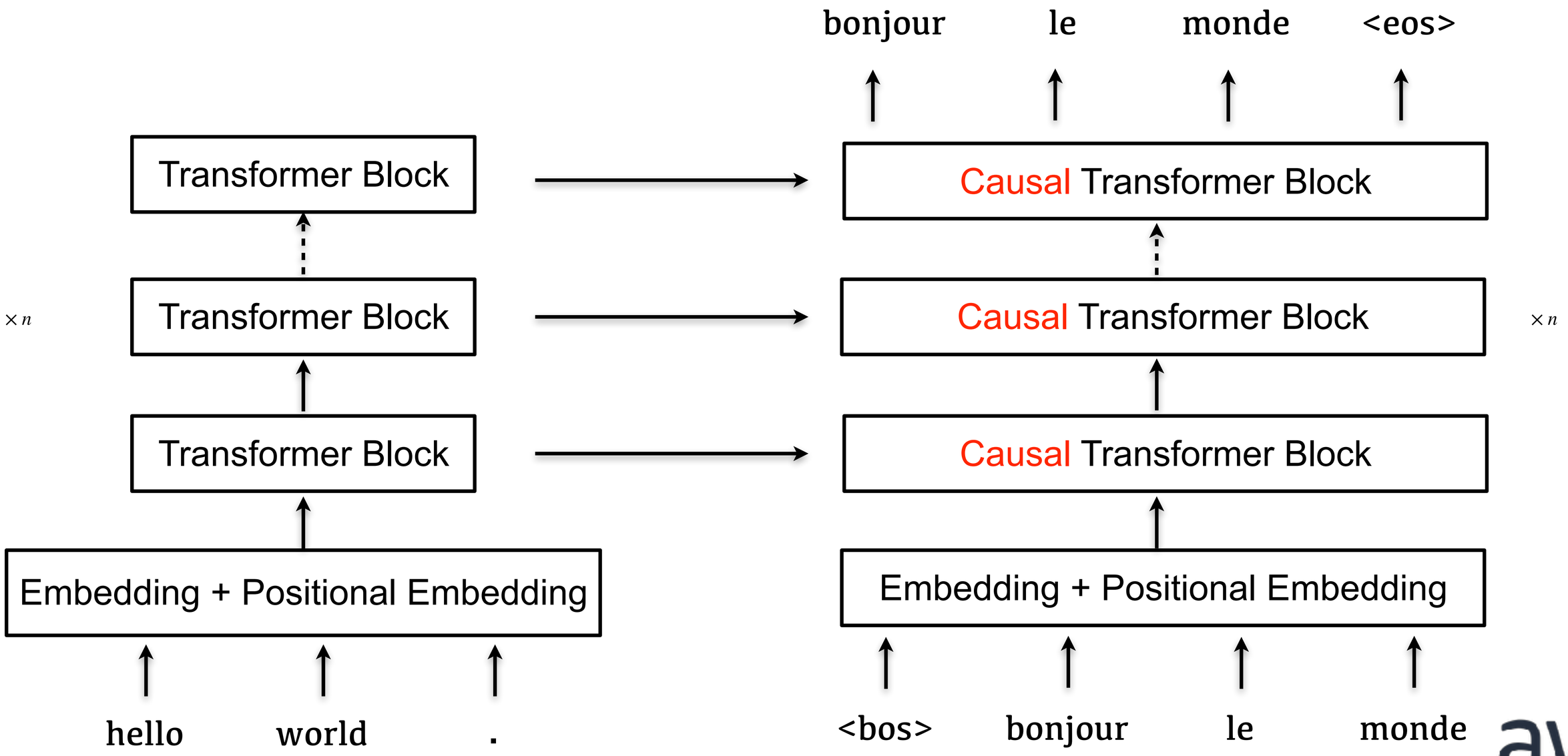
Transformer Block = Multi-head Attention + Residual + Layer Normalization



X.shape = (#Batch, #Channels)

LayerNorm: $\frac{X - X.mean(axis=-1)}{X.std(axis=-1)}$

Transformer Architecture for Machine Translation



gluon-nlp.mxnet.io

Causal Attention

- Never attend to the future. Avoid information leak.