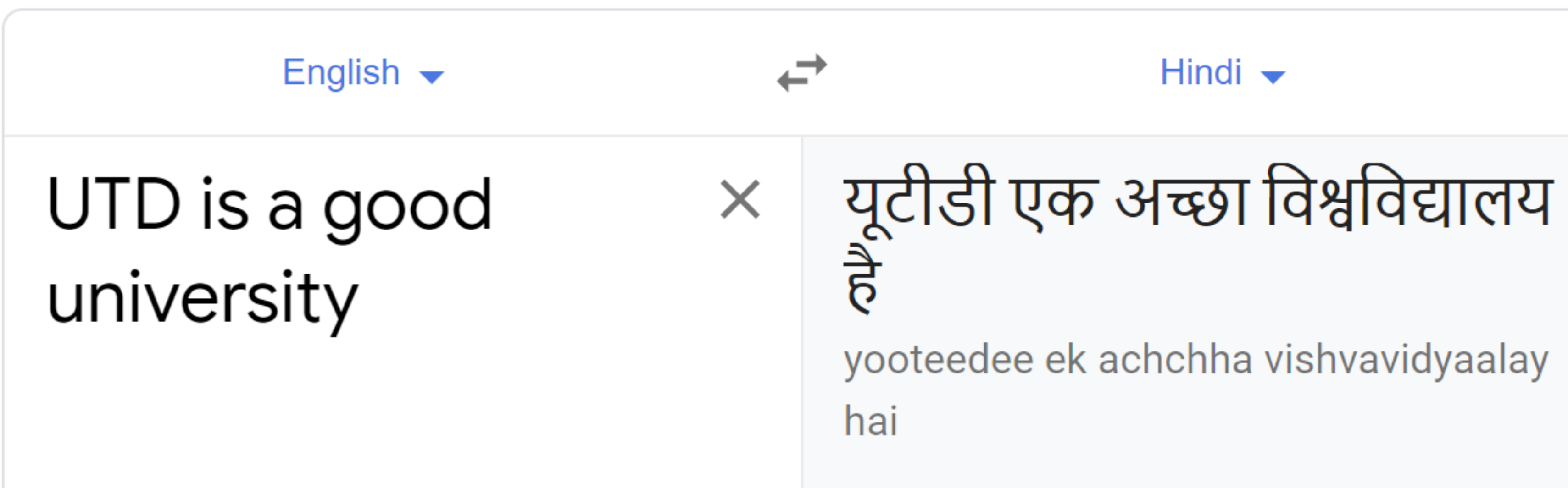
Encoder Decoder model (Sequence to Sequence Model)

Machine Translation

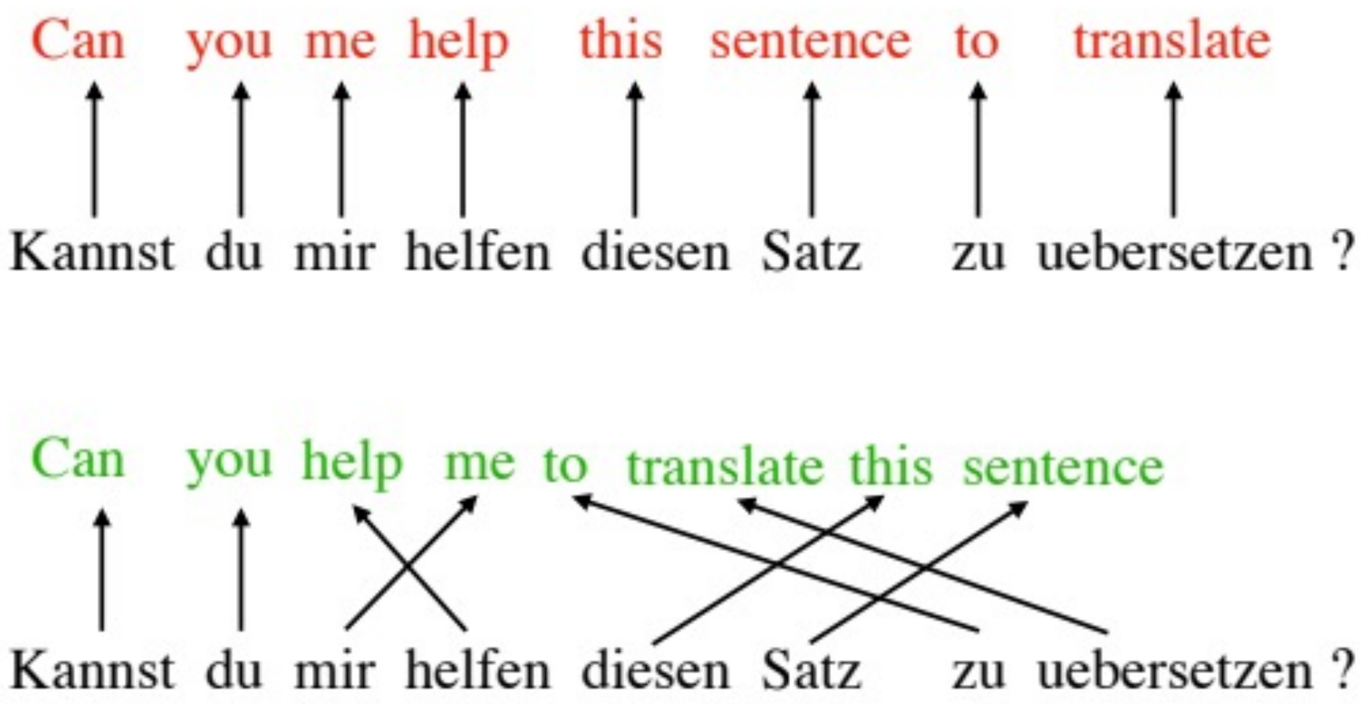
- Given a sentence in a source language, translate into a target language
- These two sequences may have different lengths



Machine Translation

- Given a sentence in a source language, translate into a target language
- These two sequences may have different lengths

We cannot translate sentences word by word



Source
: https://lightning.ai/courses/deep-learning-fundamentals/unit-8.0-natural-language-processing-and-large-language-models/8.4-from-rnns-to-the-transformer-architecture/

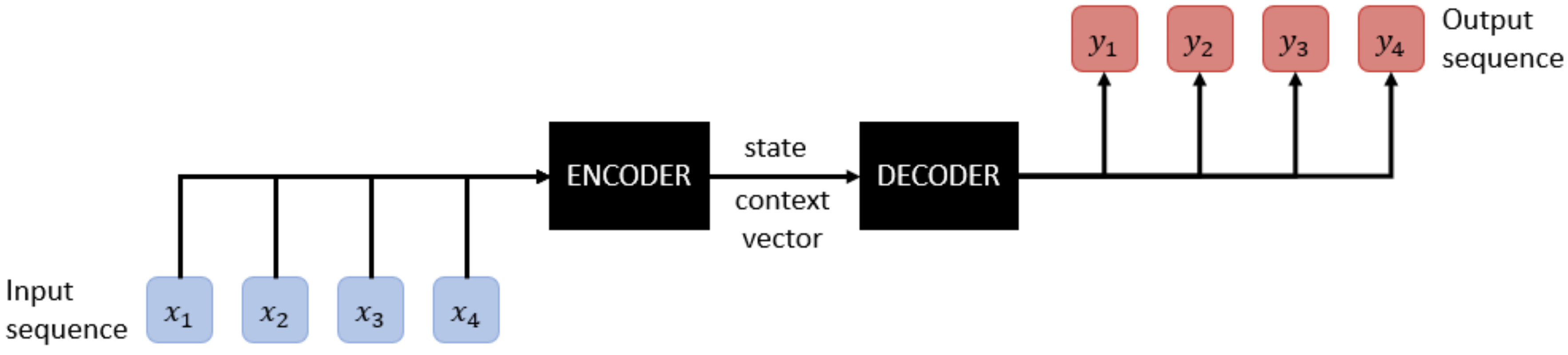
Neural Machine Translation

Neural machine Translation (NMT) – Machine translation with a single neural network

The neural network architecture typically involves two RNNs – this is called sequence-to-sequence

The Encoder-decoder Architecture

- A model is partitioned into two parts
  - The encoder process inputs
  - The decoder generates outputs

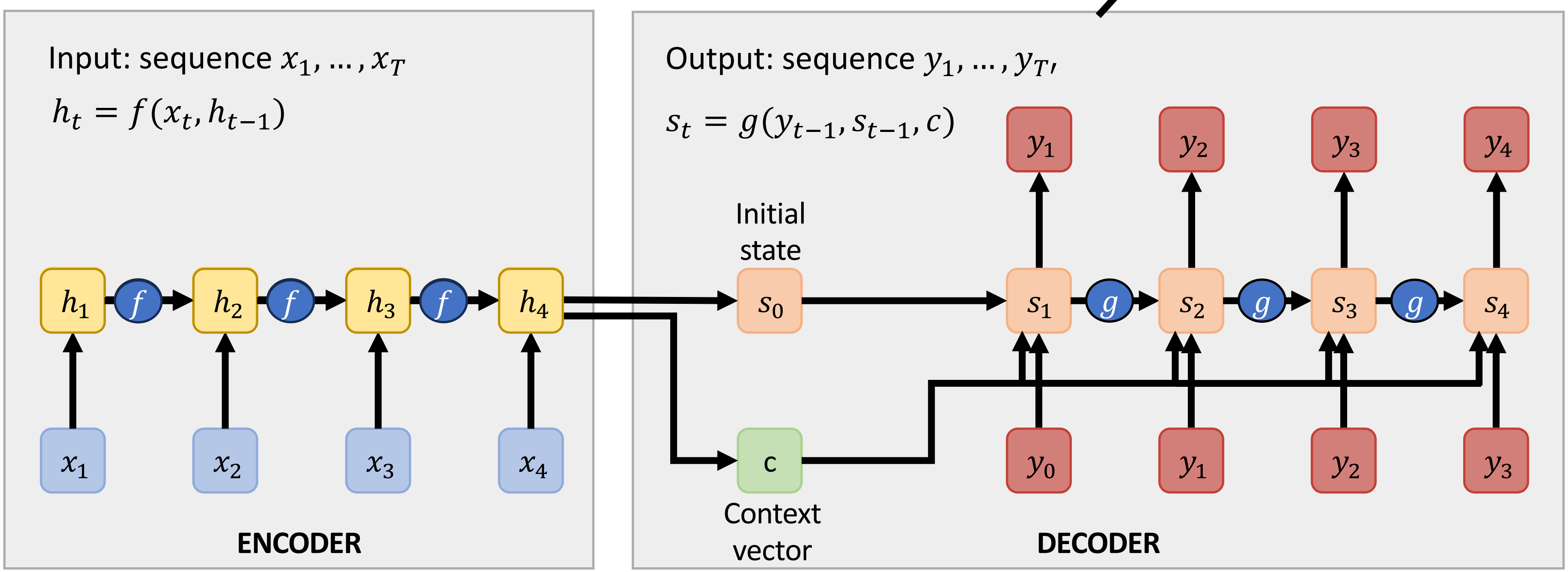


Slide Credit : https://cs.uwaterloo.ca/~wenhuche/teaching/cs886/

Sequence to Sequence with RNNs

- Encoder (LSTM) and decoder (LSTM)
- Fixed-length context vector

Language Model

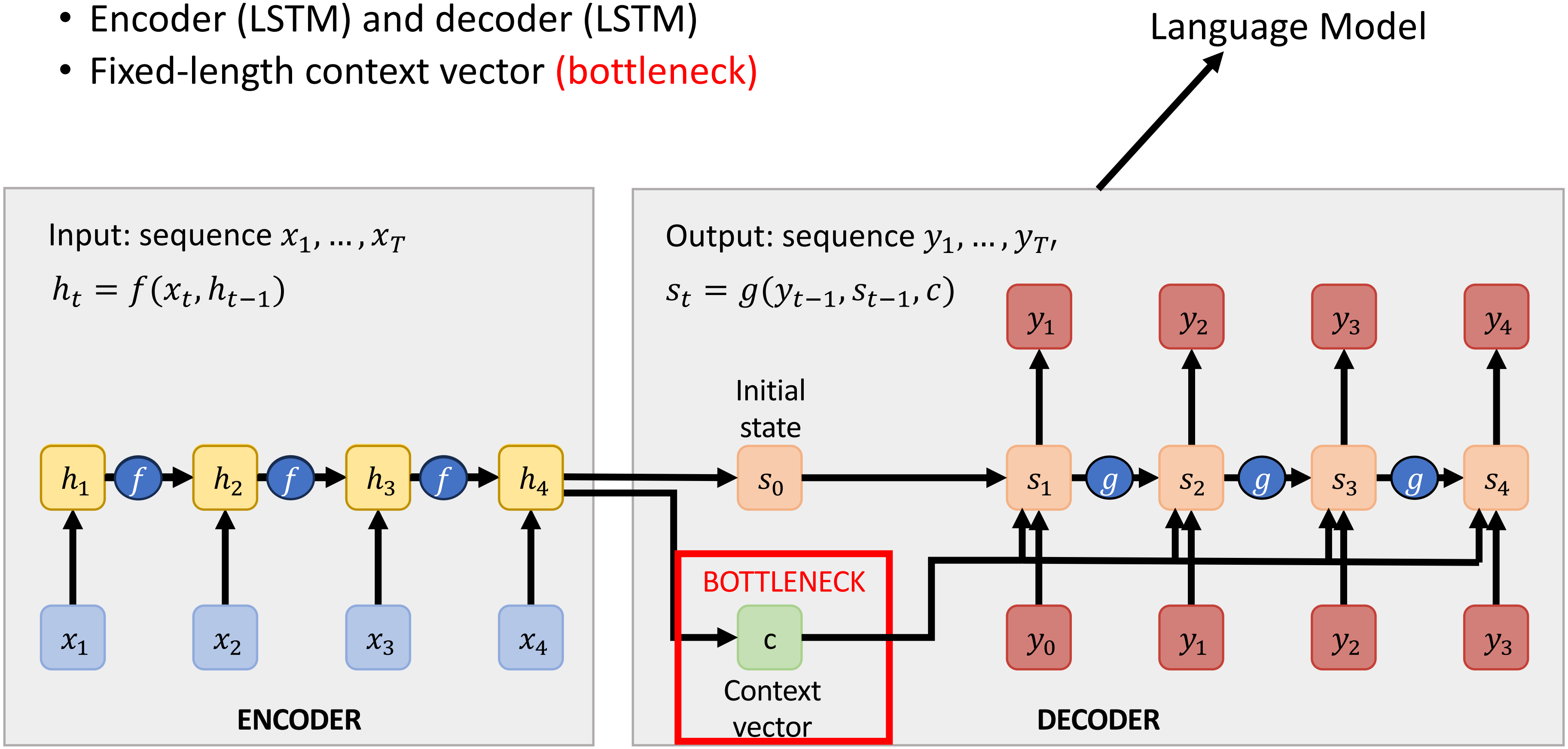


I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS)*, 2014, pp. 3104–3112.

Slide Credit : https://cs.uwaterloo.ca/~wenhuche/teaching/cs886/

Sequence to Sequence with RNNs

- Encoder (LSTM) and decoder (LSTM)
- Fixed-length context vector (bottleneck)



I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS)*, 2014, pp. 3104–3112.

Slide Credit : https://cs.uwaterloo.ca/~wenhuche/teaching/cs886/

sequence-to-sequence – bottleneck problem

- Fixed source representation is suboptimal for the encoder, it is hard to compress the sentence
- For the decoder, different information may be relevant at different steps

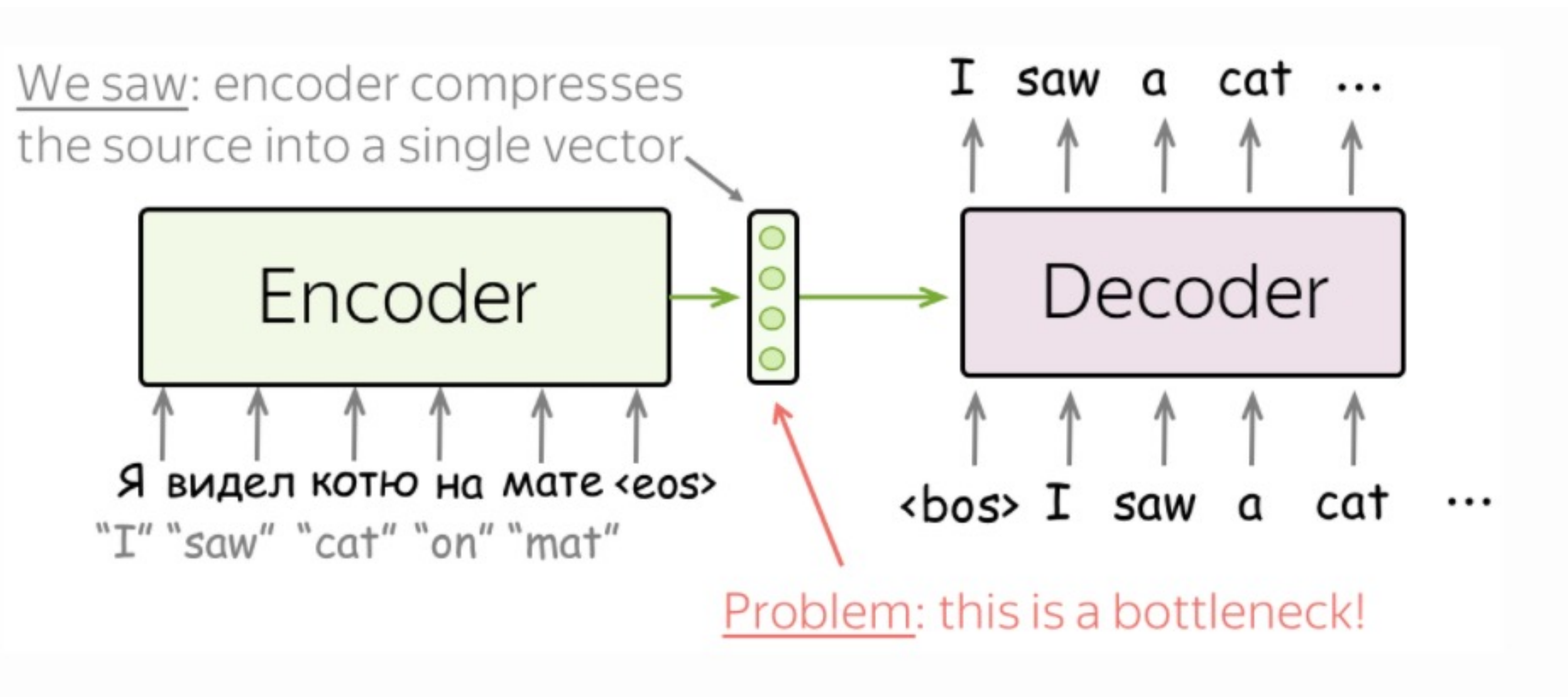
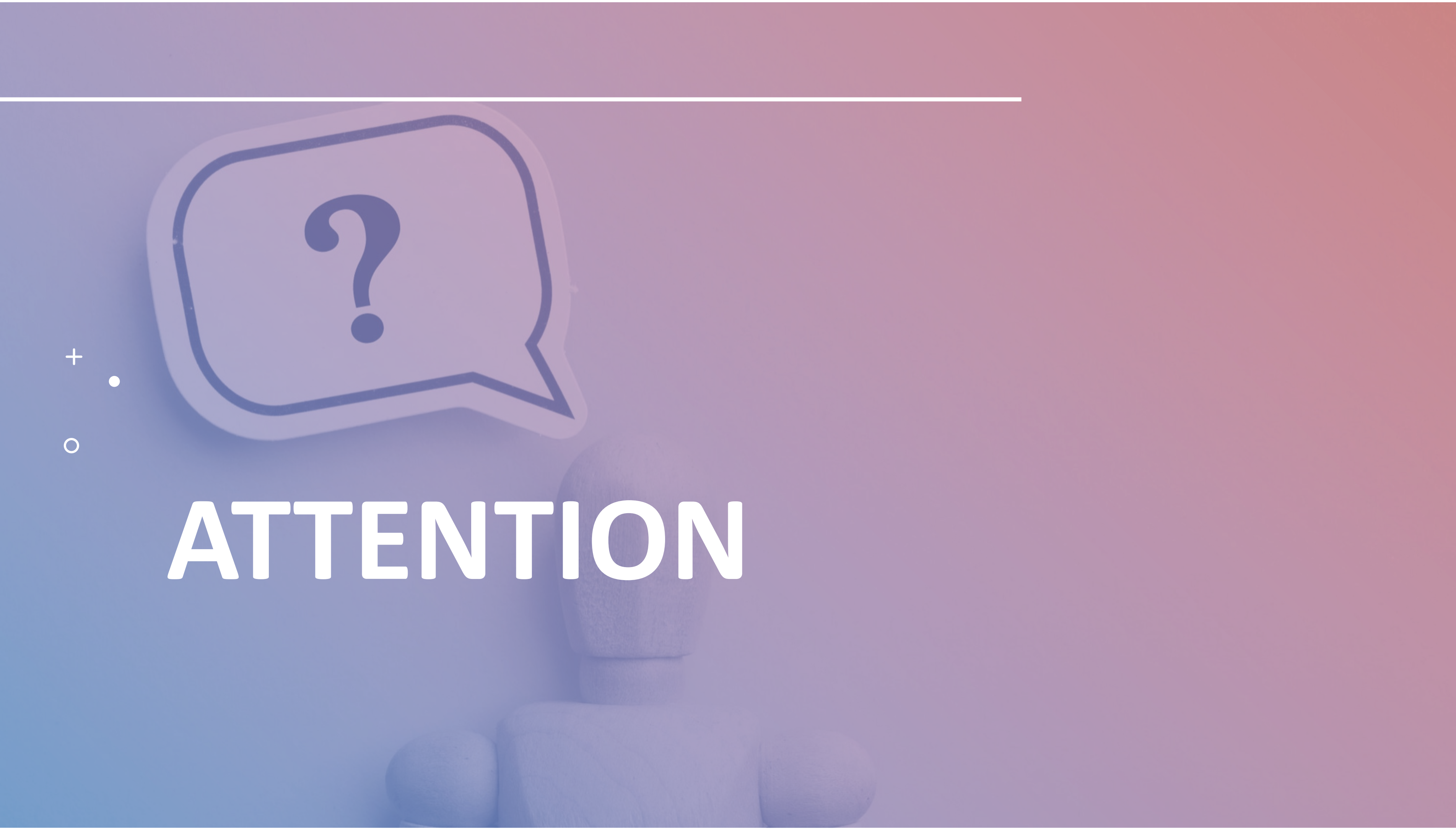


Image source https://lena-voita.github.io/nlp_course/seq2seq_and_attention.html



ATTENTION

Attention

- **Attention** provides a solution to the bottleneck problem.

- Core idea: on each step of the decoder, use *direct connection to the encoder* to *focus on a particular part* of the source sequence

Sequence to Sequence with RNNs + Attention

Idea! Use a different context vector for each timestep in the decoder

$$s_t = g(y_{t-1}, s_{t-1}, \boldsymbol{c_t})$$

No more bottleneck through a single vector

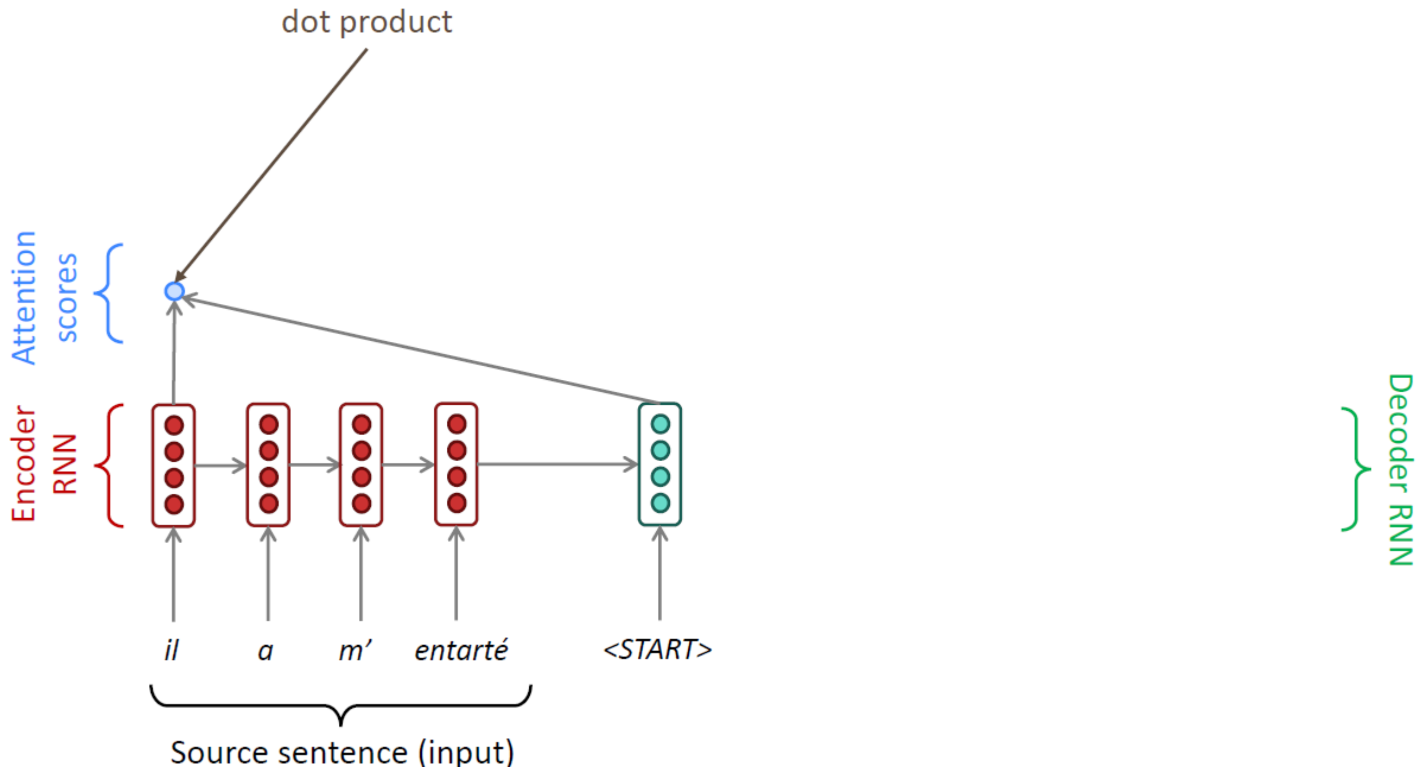
Craft the context vector so that it "looks at" different parts of the input sequence for each decoder timestep

D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," in *3rd International Conference on Learning Representations (ICLR),* 2015.

Slide Credit : https://cs.uwaterloo.ca/~wenhuche/teaching/cs886/

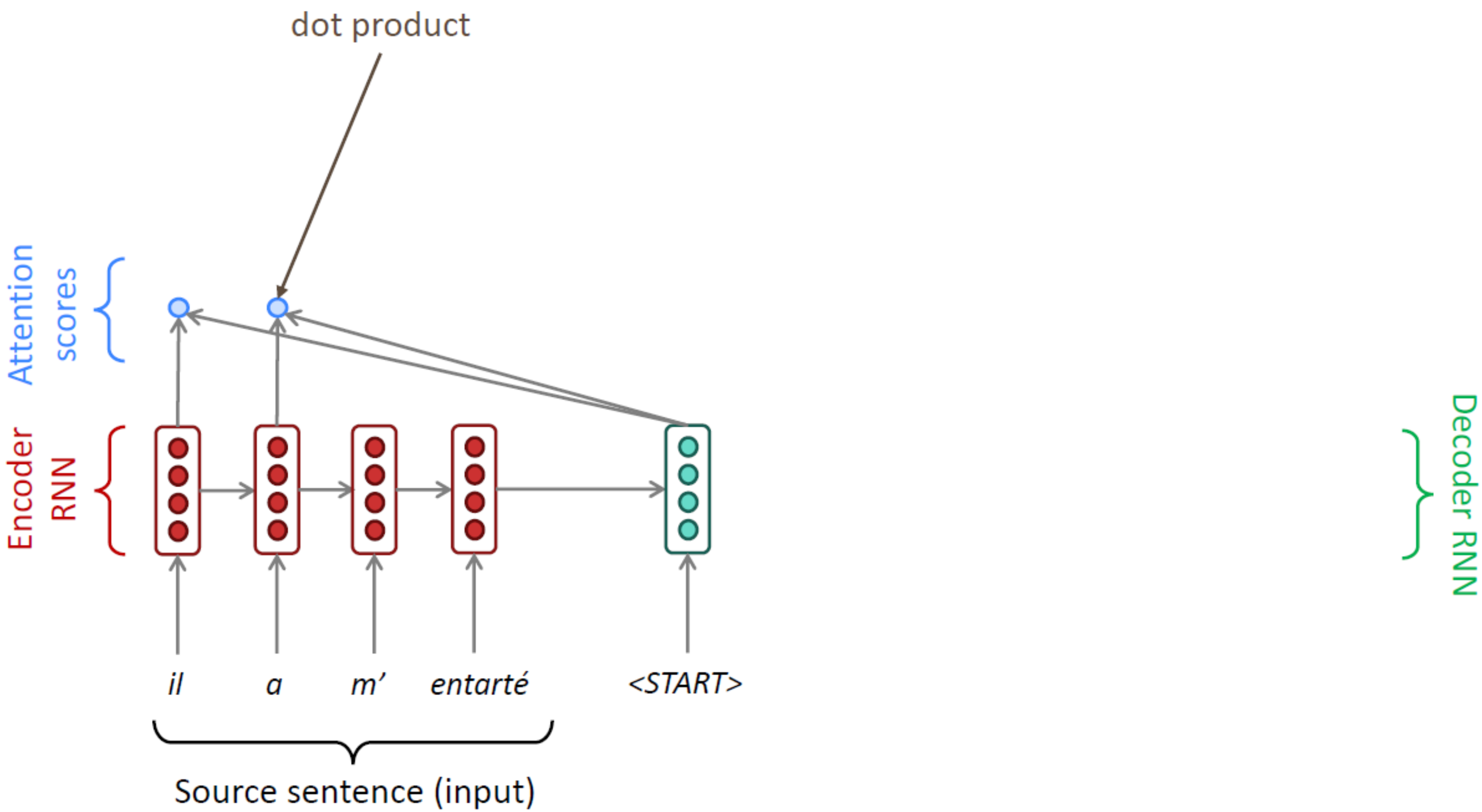
Sequence to Sequence with RNNs + Attention

Core idea: on each step of the decoder, *use direct connection to the encoder* to *focus on a particular part* of the source sequence



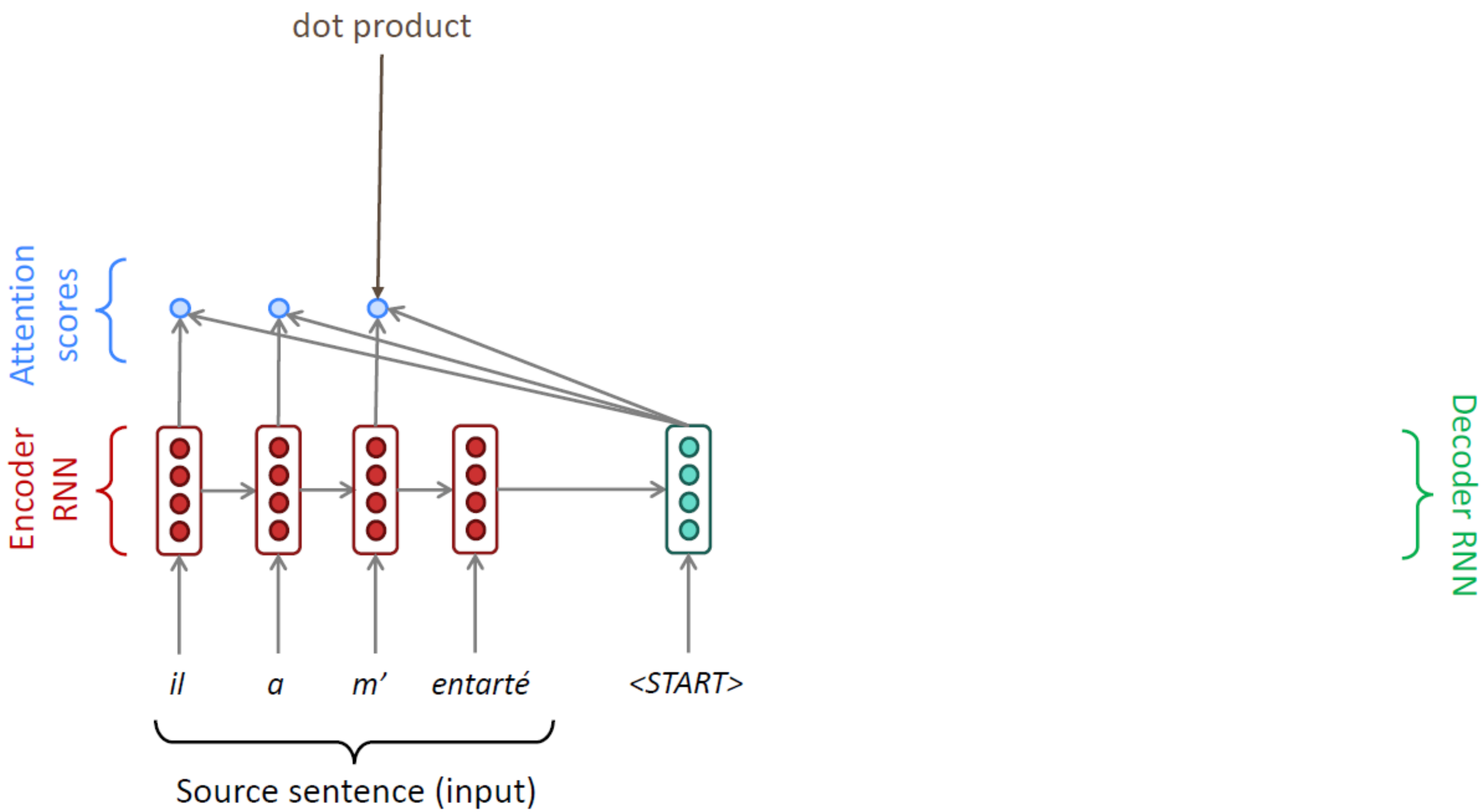
Slide Source : http://web.stanford.edu/class/cs224n/slides/cs224n-2020-lecture08-nmt.pdf

Sequence to Sequence with RNNs + Attention



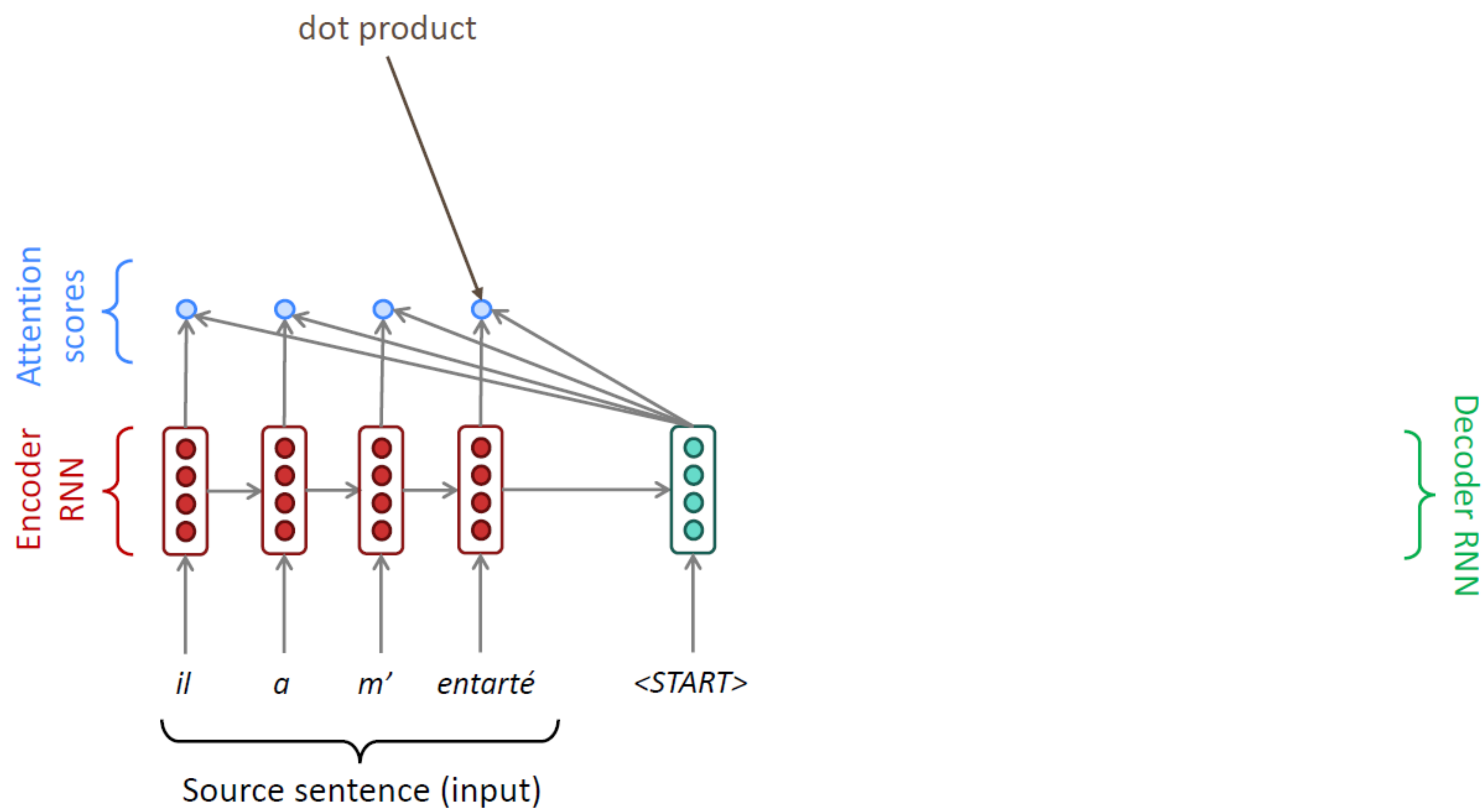
Slide Source : http://web.stanford.edu/class/cs224n/slides/cs224n-2020-lecture08-nmt.pdf

Sequence to Sequence with RNNs + Attention



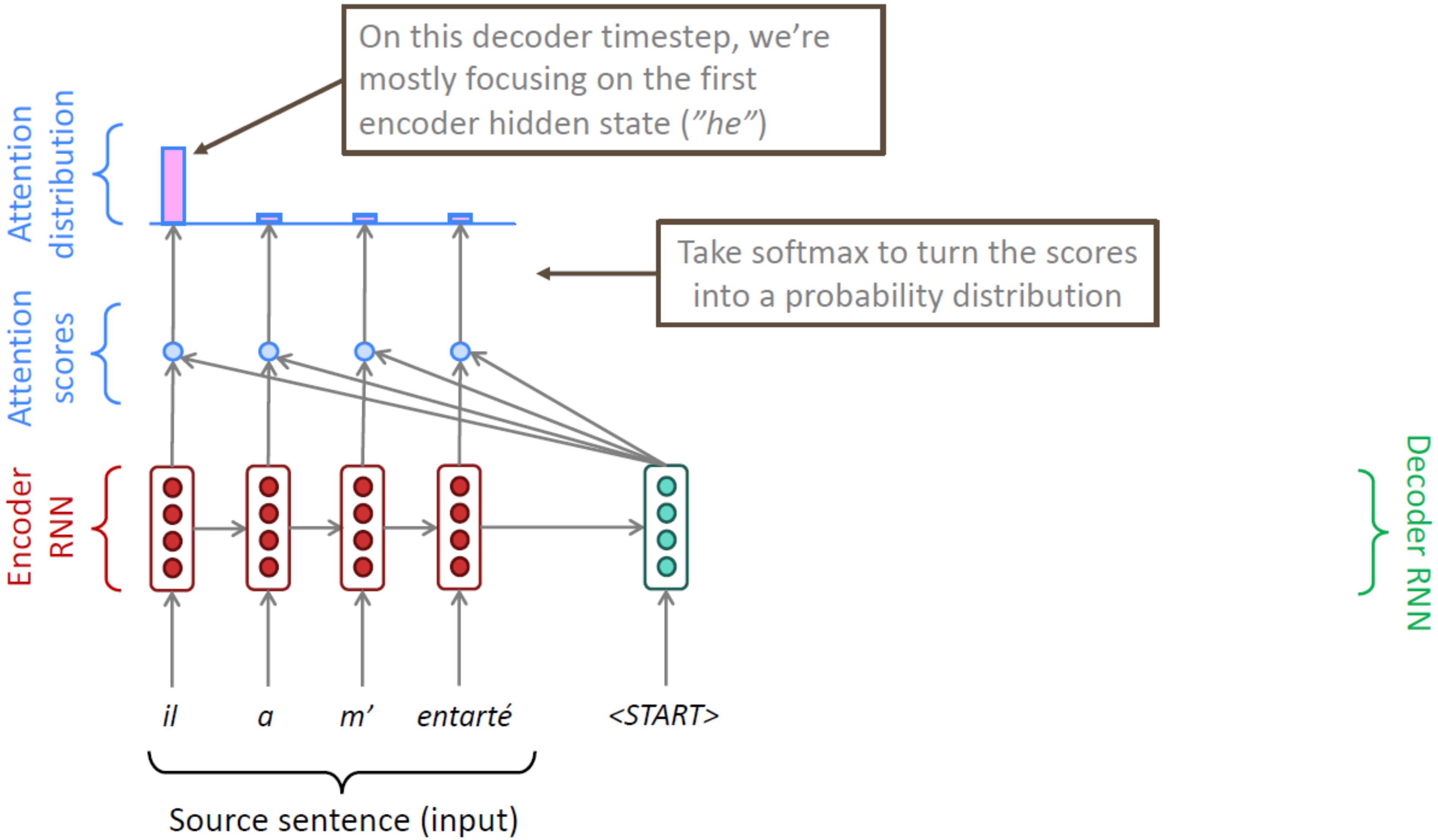
Slide Source : http://web.stanford.edu/class/cs224n/slides/cs224n-2020-lecture08-nmt.pdf

Sequence to Sequence with RNNs + Attention



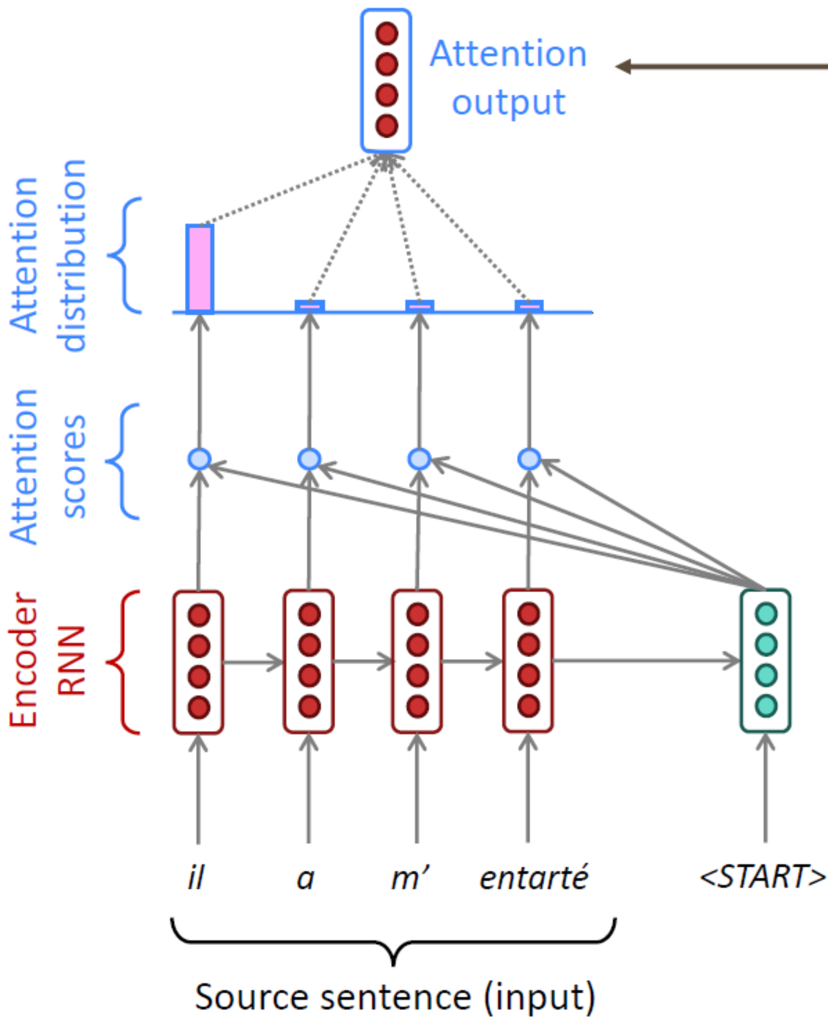
Slide Source : http://web.stanford.edu/class/cs224n/slides/cs224n-2020-lecture08-nmt.pdf

Sequence to Sequence with RNNs + Attention



Slide Source : http://web.stanford.edu/class/cs224n/slides/cs224n-2020-lecture08-nmt.pdf

Sequence to Sequence with RNNs + Attention

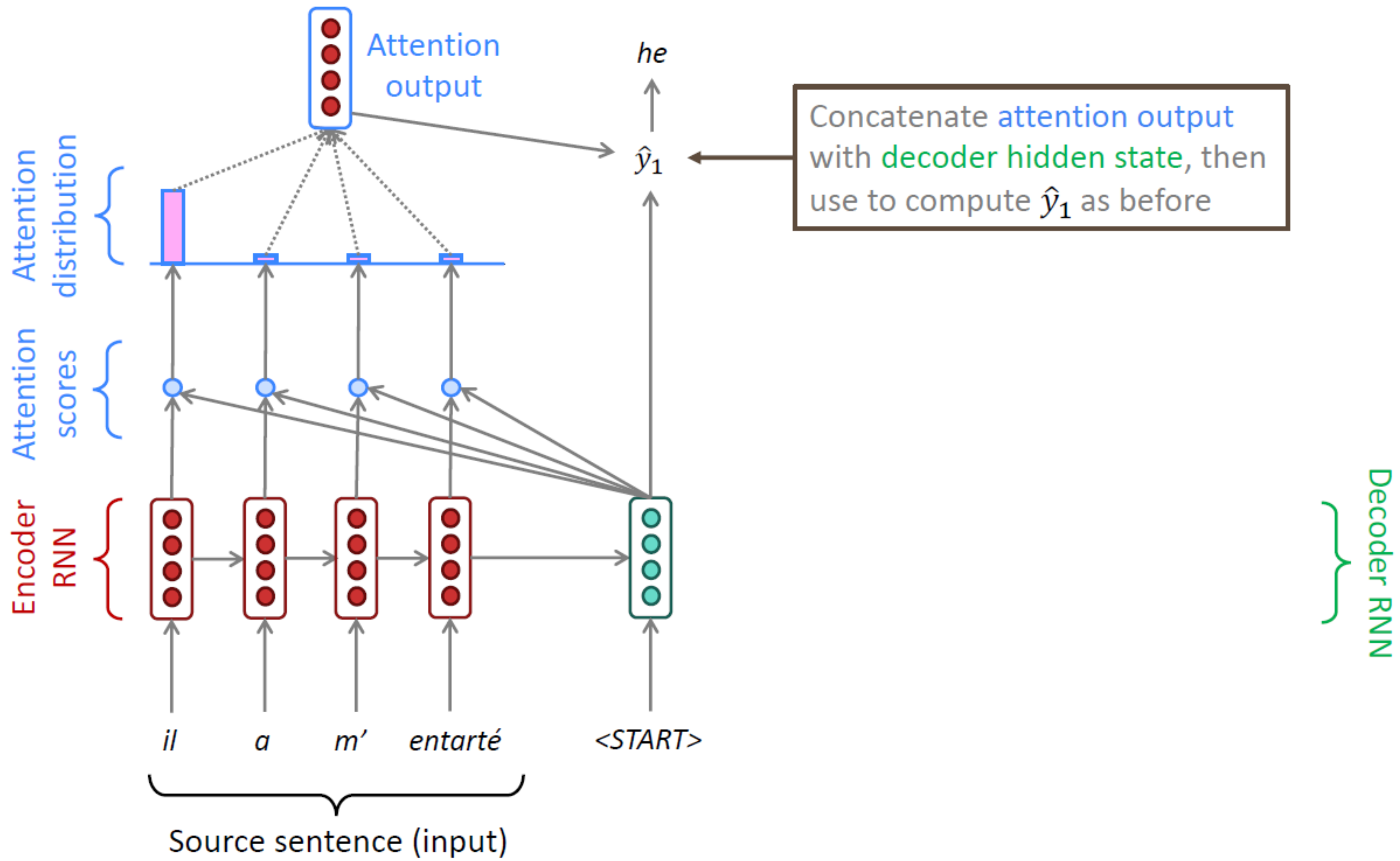


Use the attention distribution to take a **weighted sum** of the encoder hidden states.

The attention output mostly contains information from the hidden states that received high attention.

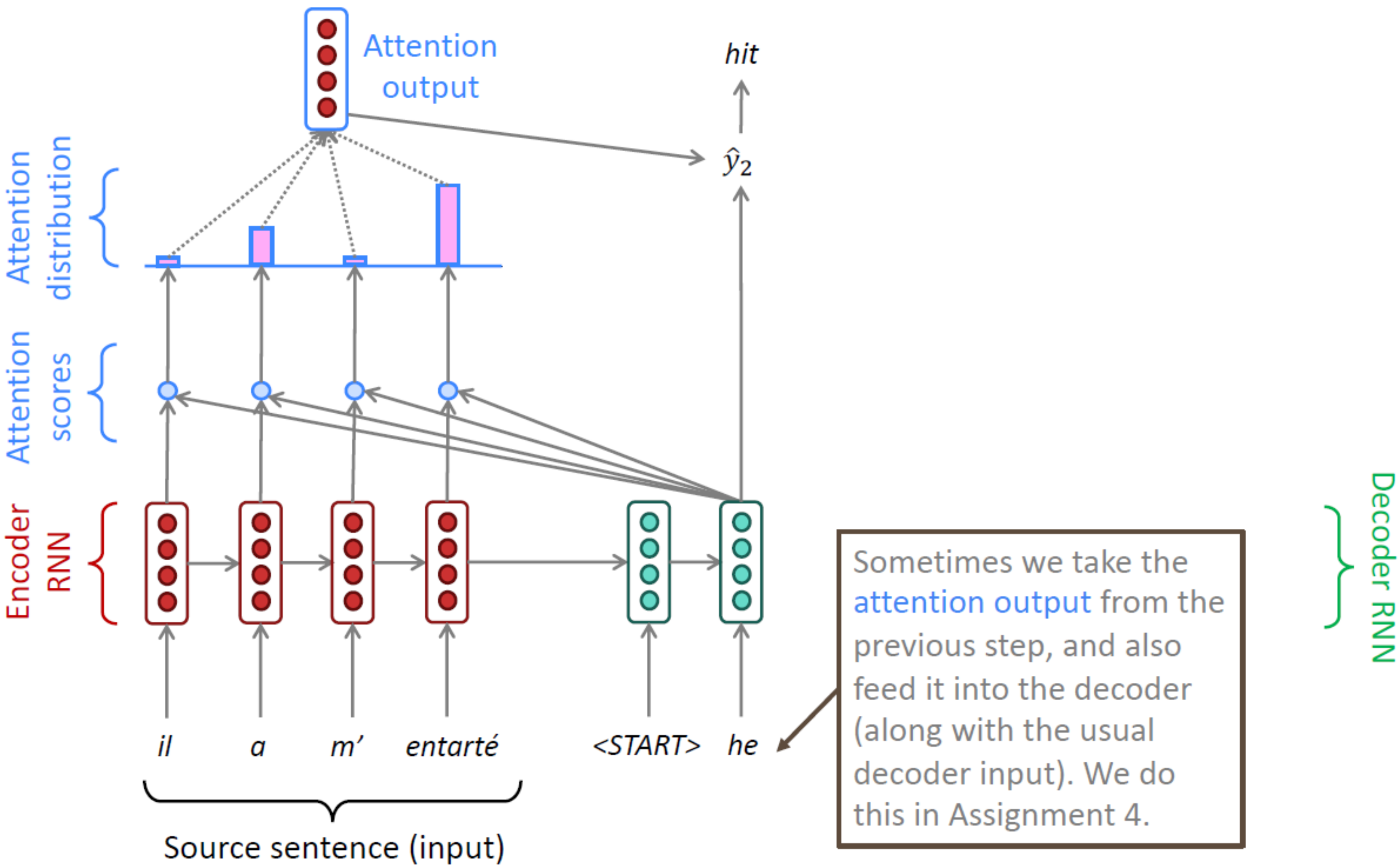
Slide Source : http://web.stanford.edu/class/cs224n/slides/cs224n-2020-lecture08-nmt.pdf

Sequence to Sequence with RNNs + Attention



Slide Source : http://web.stanford.edu/class/cs224n/slides/cs224n-2020-lecture08-nmt.pdf

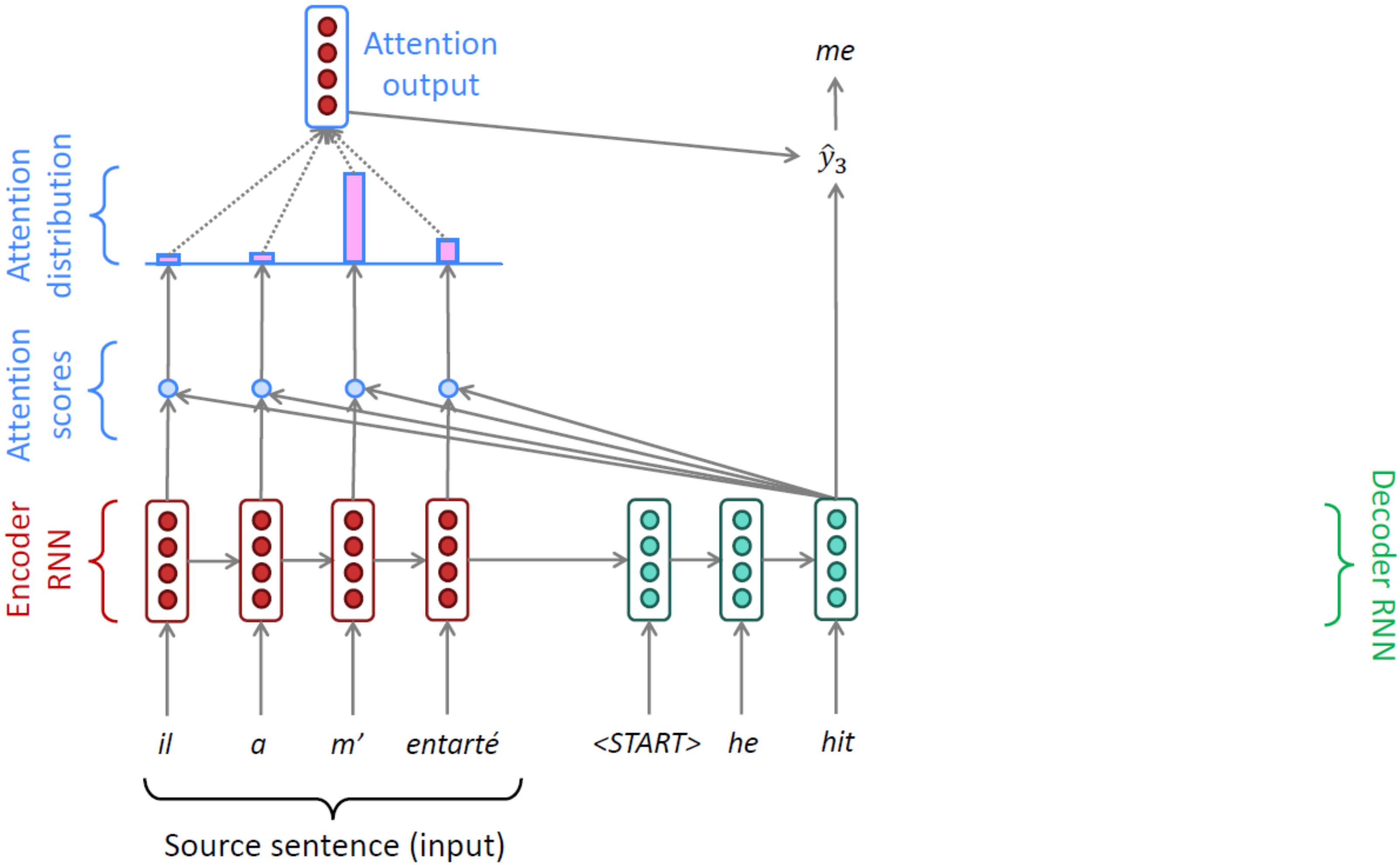
Sequence to Sequence with RNNs + Attention



Sometimes we take the attention output from the previous step, and also feed it into the decoder (along with the usual decoder input). We do this in Assignment 4.

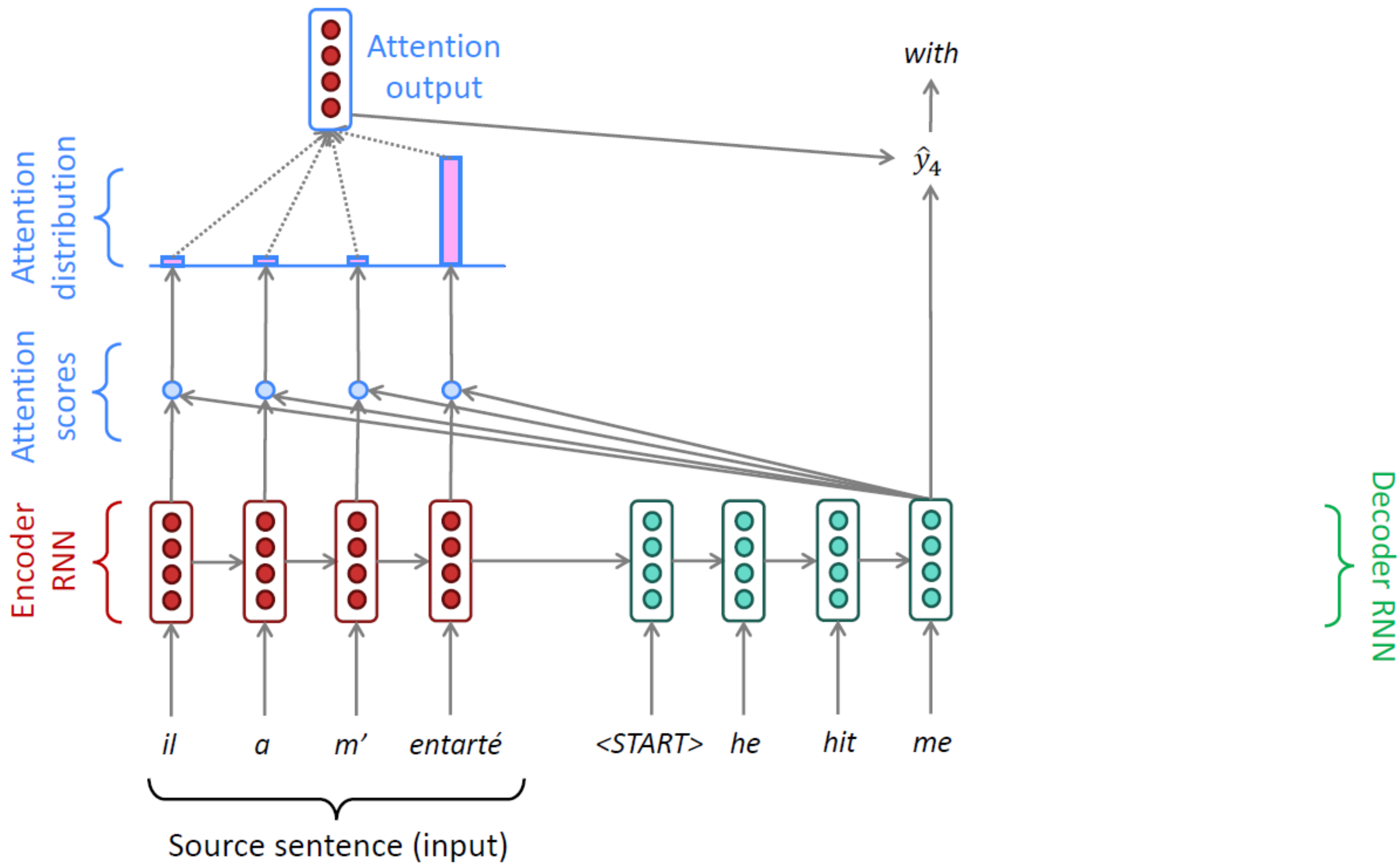
Slide Source : http://web.stanford.edu/class/cs224n/slides/cs224n-2020-lecture08-nmt.pdf

Sequence to Sequence with RNNs + Attention



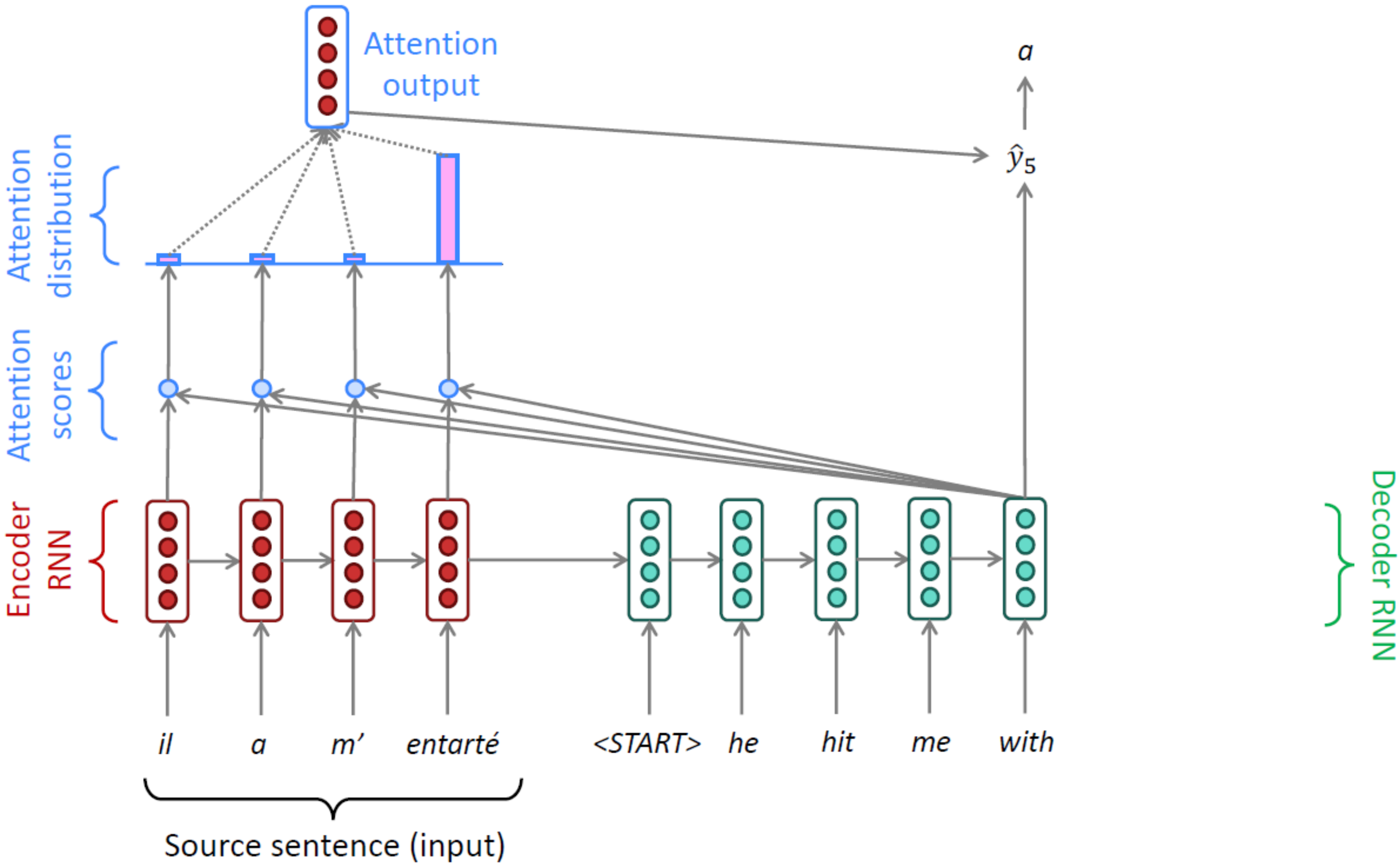
Slide Source : http://web.stanford.edu/class/cs224n/slides/cs224n-2020-lecture08-nmt.pdf

Sequence to Sequence with RNNs + Attention



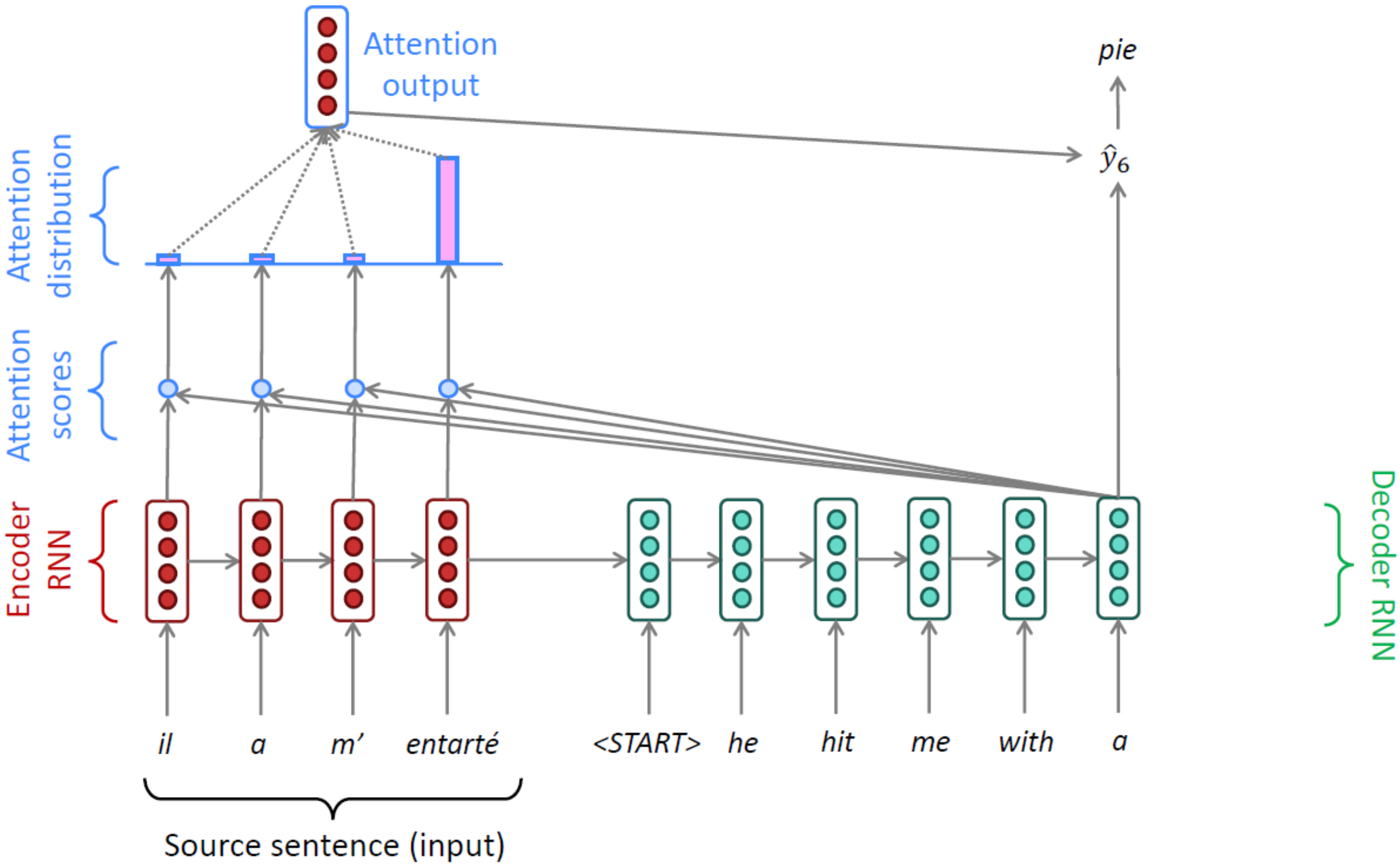
Slide Source : http://web.stanford.edu/class/cs224n/slides/cs224n-2020-lecture08-nmt.pdf

Sequence to Sequence with RNNs + Attention



Slide Source : http://web.stanford.edu/class/cs224n/slides/cs224n-2020-lecture08-nmt.pdf

Sequence to Sequence with RNNs + Attention



Slide Source : http://web.stanford.edu/class/cs224n/slides/cs224n-2020-lecture08-nmt.pdf

Attention

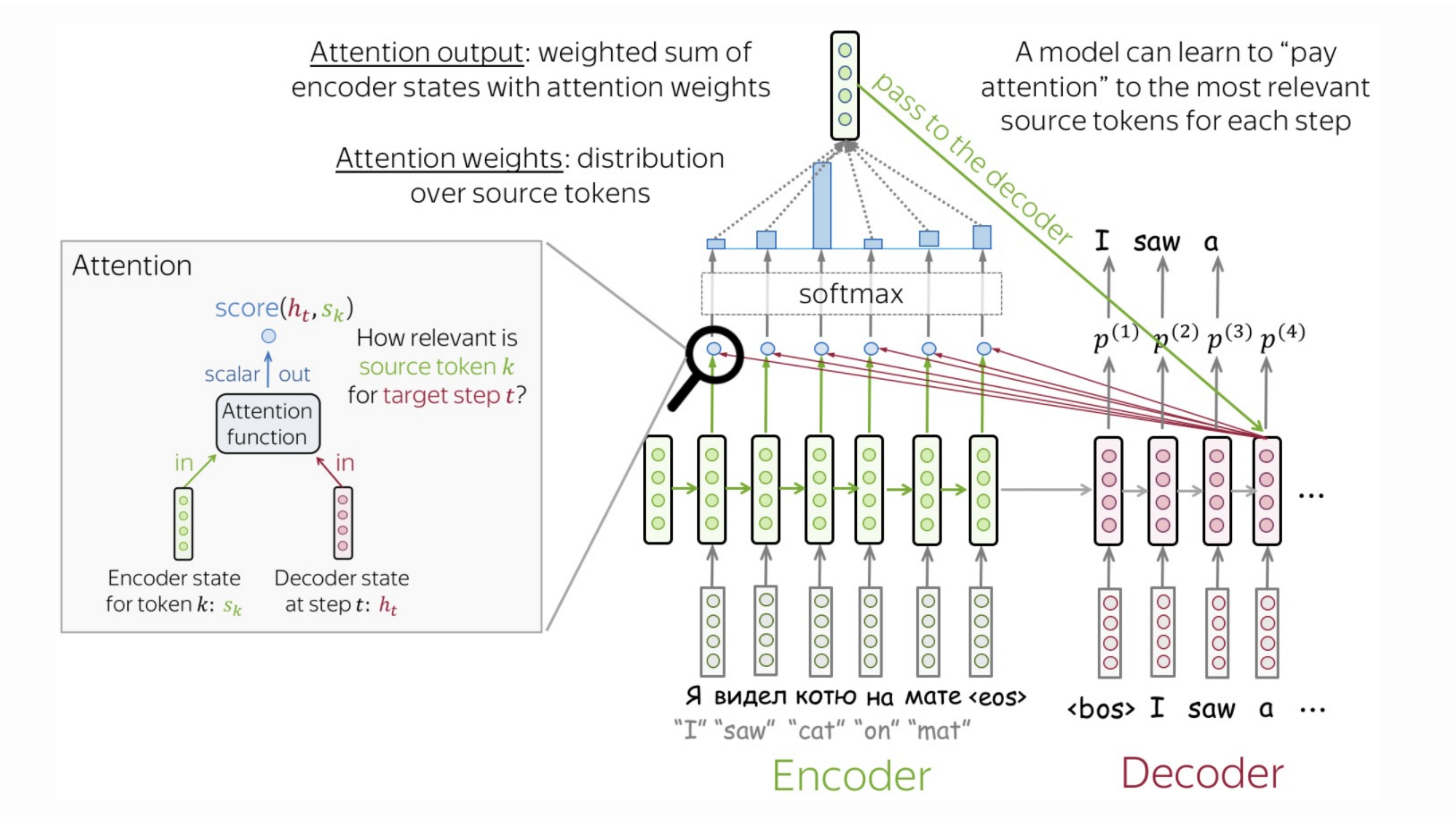


Image source https://lena-voita.github.io/nlp_course/seq2seq_and_attention.html

Bahdanau Model

Attention

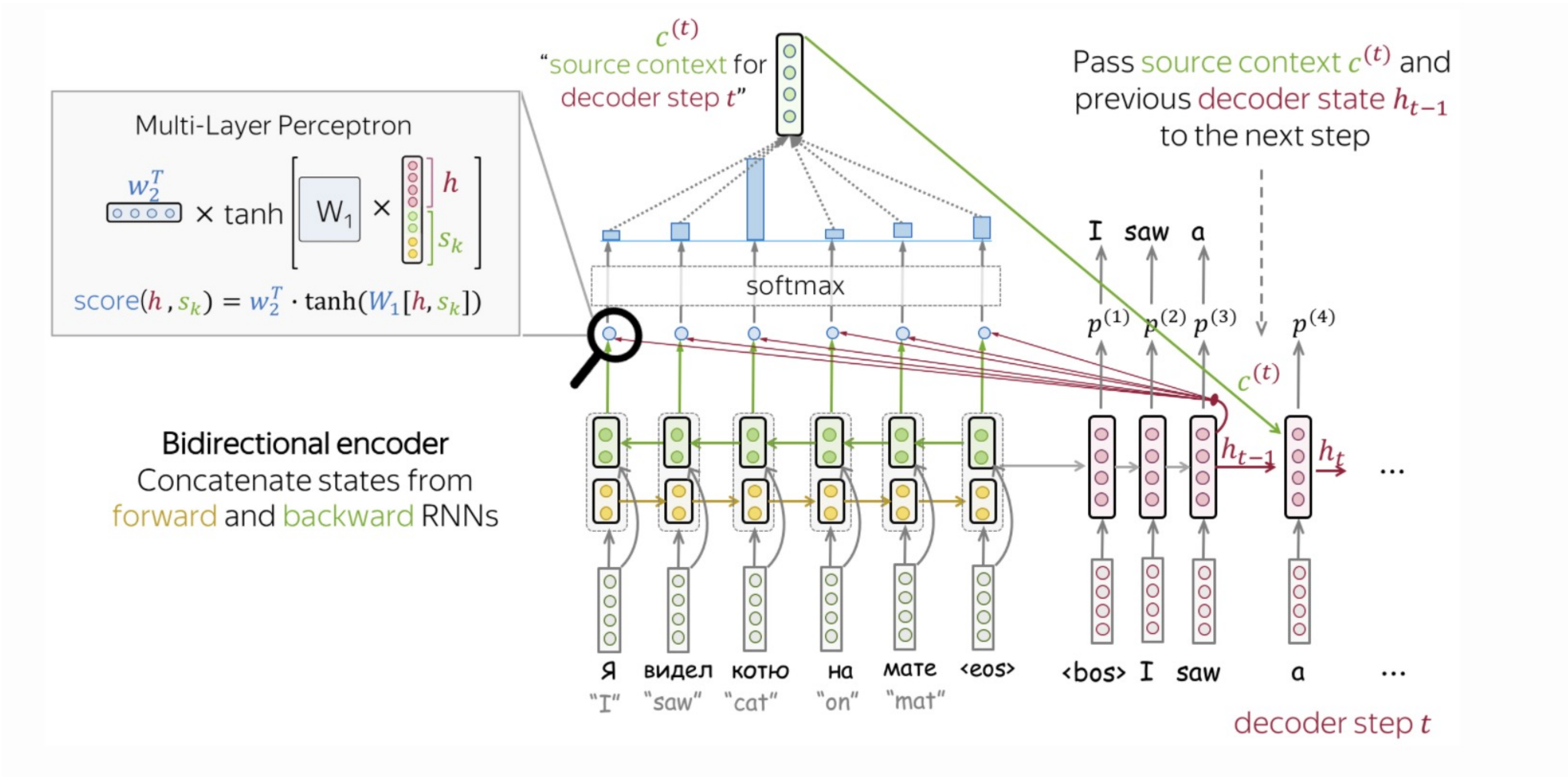


Image source https://lena-voita.github.io/nlp_course/seq2seq_and_attention.html

Attention Layer

Attention output

$$c^{(t)} = a_1^{(t)} s_1 + a_2^{(t)} s_2 + \cdots + a_m^{(t)} s_m = \sum_{k=1}^{m} a_k^{(t)} s_k$$

"source context for decoder step t"

(weighted sum)

Attention weights

$$a_k^{(t)} = \frac{\exp(\text{score}(h_t, s_k))}{\sum_{i=1}^{m} \exp(\text{score}(h_t, s_i))}, \mathrm{k} = 1..\mathrm{m}$$

"attention weight for source token k at decoder step t"

(softmax)

Attention scores

$$\text{score}(h_t, s_k), \mathrm{k} = 1..\mathrm{m}$$

"How relevant is source token k for target step t?"

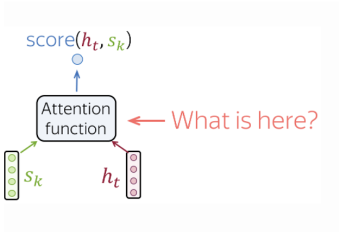
Attention input

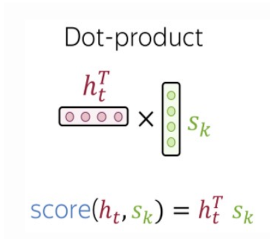
$s_1, s_2, \ldots, s_m$ — all encoder states

h_t — one decoder state

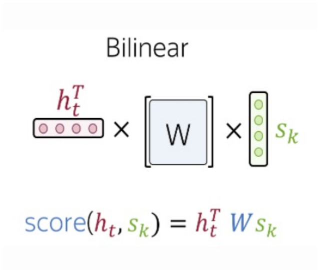
Image source https://lena-voita.github.io/nlp_course/seq2seq_and_attention.html

Ways to Calculate Attention scores

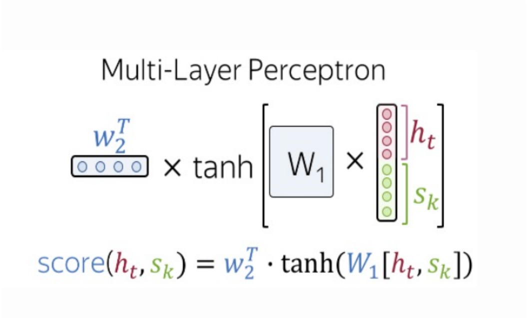




dot-product - the simplest method



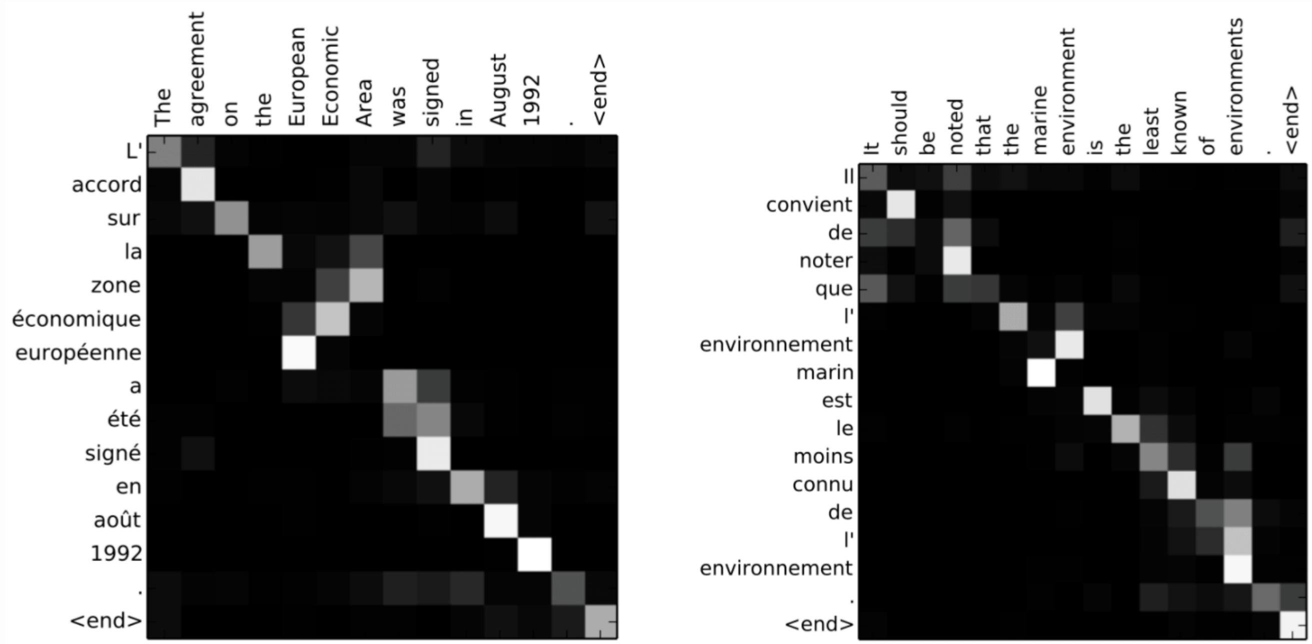
bilinear function (aka "Luong attention") - used in the paper [Effective Approaches to Attention-based Neural Machine Translation]



multi-layer perceptron (aka "Bahdanau attention") - the method proposed in the [original paper].

Image source https://lena-voita.github.io/nlp_course/seq2seq_and_attention.html

Attention Learns (Nearly) Alignment



The examples are from the paper Neural Machine Translation by Jointly Learning to Align and Translate.

Attention is great!

- Attention significantly improves NMT performance
  - It's very useful to allow decoder to focus on certain parts of the source
- Attention provides a more "human-like" model of the MT process
  - You can look back at the source sentence while translating, rather than needing to remember it all
- Attention solves the bottleneck problem
  - Attention allows decoder to look directly at source; bypass bottleneck
- Attention helps with the vanishing gradient problem
  - Provides shortcut to faraway states
- Attention provides some interpretability
  - By inspecting attention distribution, we see what the decoder was focusing on
  - We get (soft) alignment for free!
  - This is cool because we never explicitly trained an alignment system
  - The network just learned alignment by itself