

Machine Translation



Machine Translation (MT) is the task of translating a sentence x from one language (the **source language**) to a sentence y in another language (the **target language**).

x : *L'homme est né libre, et partout il est dans les fers*



y : *Man is born free, but everywhere he is in chains*

What is Neural Machine Translation?

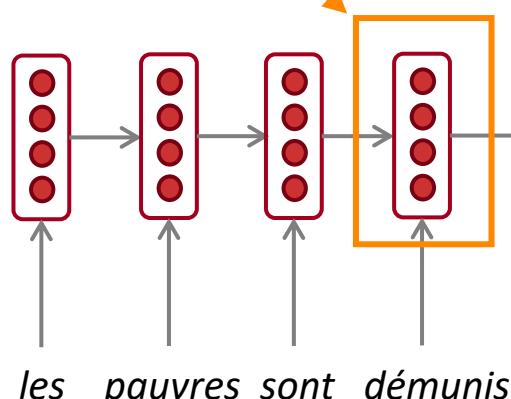
- Neural Machine Translation (NMT) is a way to do Machine Translation with a *single neural network*
- The neural network architecture is called sequence-to-sequence (aka seq2seq) and it involves *two RNNs*.

Neural Machine Translation (NMT)

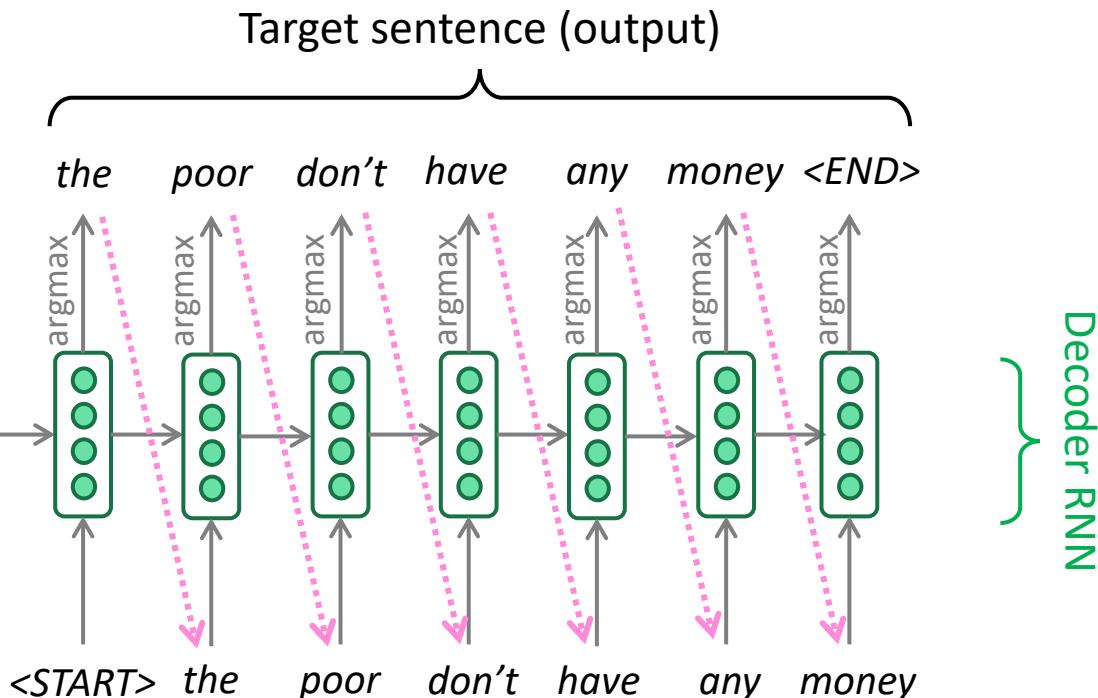
The sequence-to-sequence model

Encoding of the source sentence.

Provides initial hidden state
for Decoder RNN.



Encoder RNN produces
an encoding of the
source sentence.



Decoder RNN is a Language Model that generates target sentence conditioned on encoding.

Note: This diagram shows test time behavior:
decoder output is fed in as next step's input

Neural Machine Translation (NMT)

- The sequence-to-sequence model is an example of a **Conditional Language Model**.
 - **Language Model** because the decoder is predicting the next word of the target sentence y
 - **Conditional** because its predictions are *also* conditioned on the source sentence x

- NMT directly calculates $P(y|x)$:

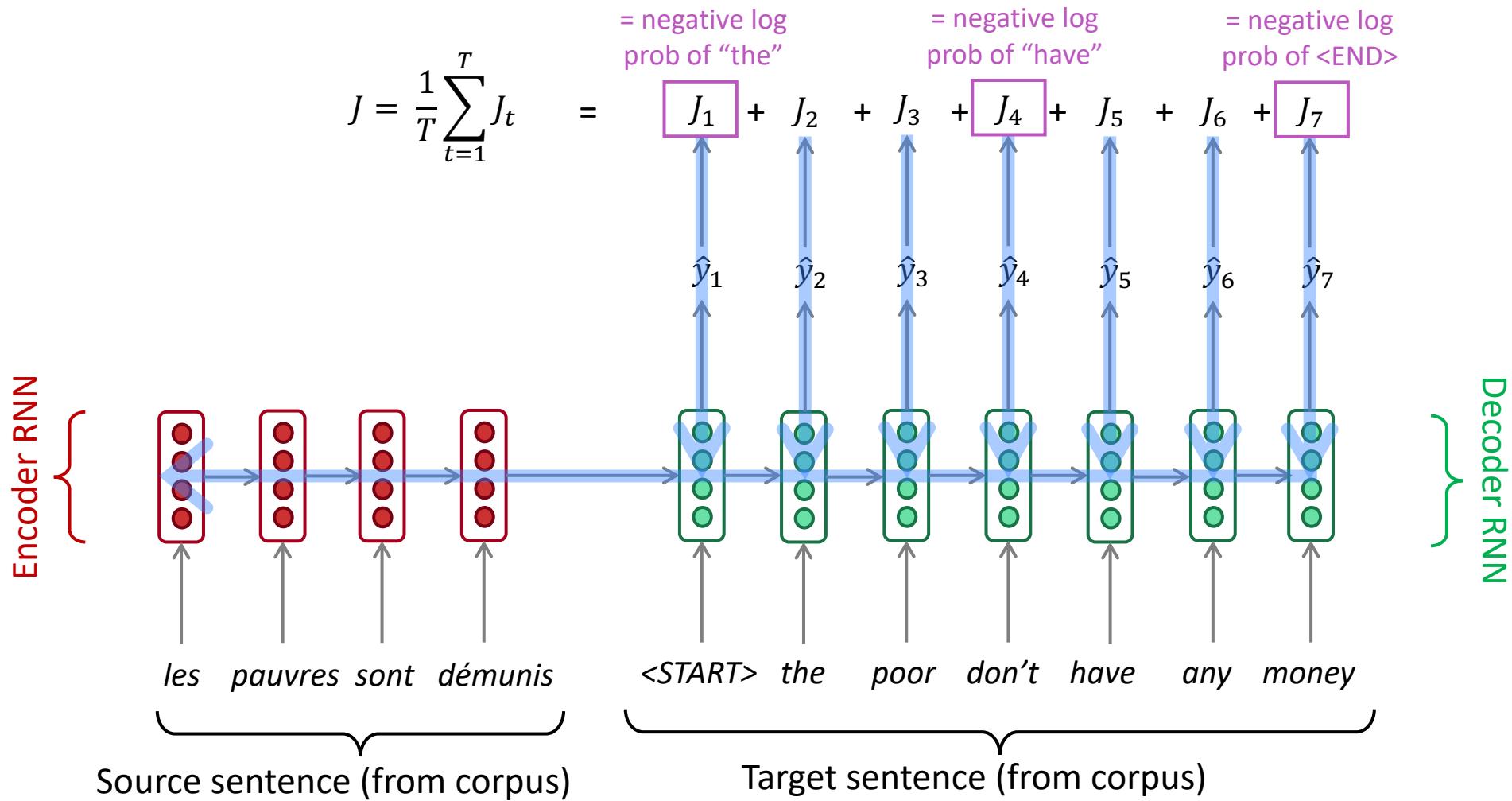
$$P(y|x) = P(y_1|x) P(y_2|y_1, x) P(y_3|y_1, y_2, x) \dots, P(y_T|y_1, \dots, y_{T-1}, x)$$



Probability of next target word, given target words so far and source sentence x

- **Question:** How to **train** a NMT system?
- **Answer:** Get a big parallel corpus...

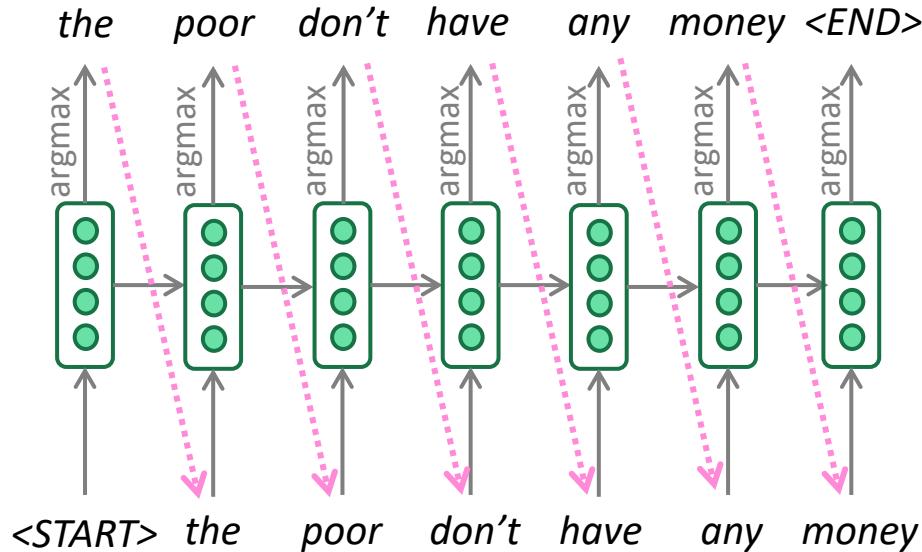
Training a Neural Machine Translation system



Seq2seq is optimized as a single system.
Backpropagation operates “end to end”.

Better-than-greedy decoding?

- We showed how to generate (or “decode”) the target sentence by taking argmax on each step of the decoder



- This is **greedy decoding** (take most probable word on each step)
- Problems?**

Better-than-greedy decoding?

- Greedy decoding has no way to undo decisions!
 - *les pauvres sont démunis* (*the poor don't have any money*)
 - → *the* _____
 - → *the poor* _____
 - → *the poor are* _____
- Better option: use beam search (a search algorithm) to explore several hypotheses and select the best one

Beam search decoding

- Ideally we want to find y that maximizes
$$P(y|x) = P(y_1|x) P(y_2|y_1, x) P(y_3|y_1, y_2, x) \dots, P(y_T|y_1, \dots, y_{T-1}, x)$$
- We could try enumerating all $y \rightarrow$ too expensive!
 - Complexity $O(V^T)$ where V is vocab size and T is target sequence length
- Beam search: On each step of decoder, keep track of the k most probable partial translations
 - k is the beam size (in practice around 5 to 10)
 - Not guaranteed to find optimal solution
 - But much more efficient!

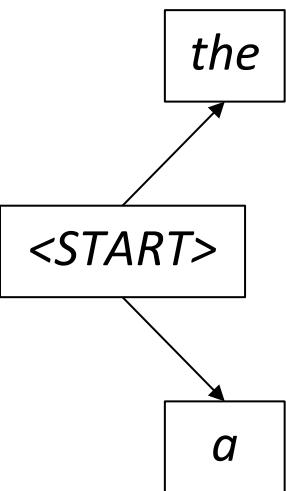
Beam search decoding: example

Beam size = 2

<START>

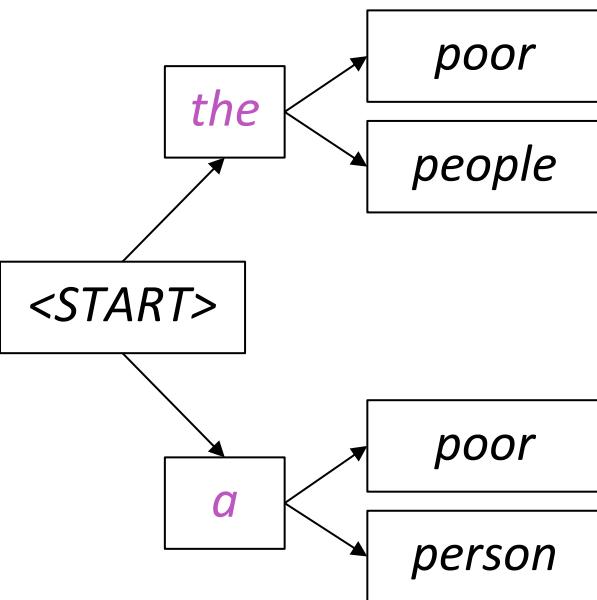
Beam search decoding: example

Beam size = 2



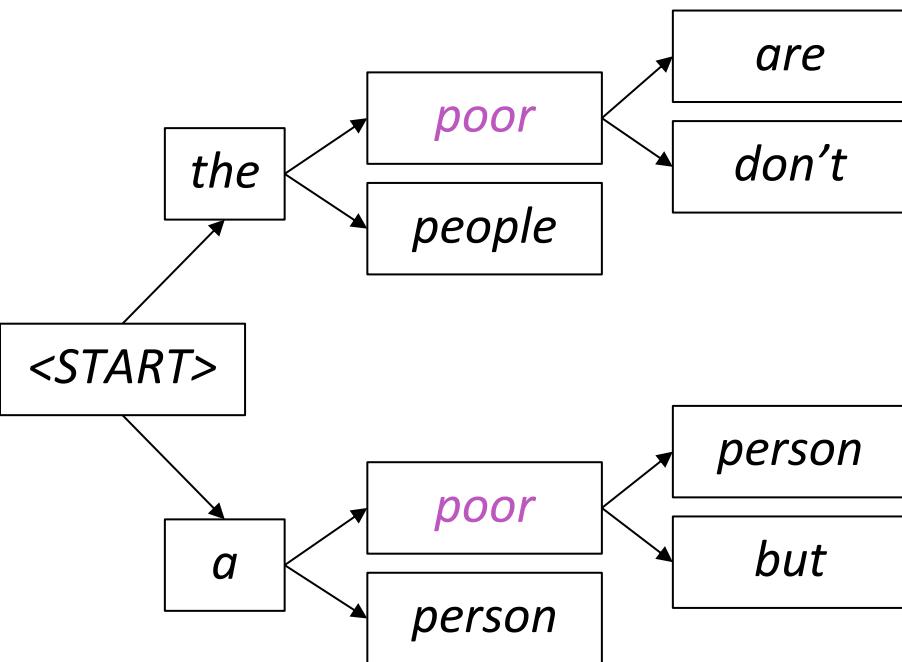
Beam search decoding: example

Beam size = 2



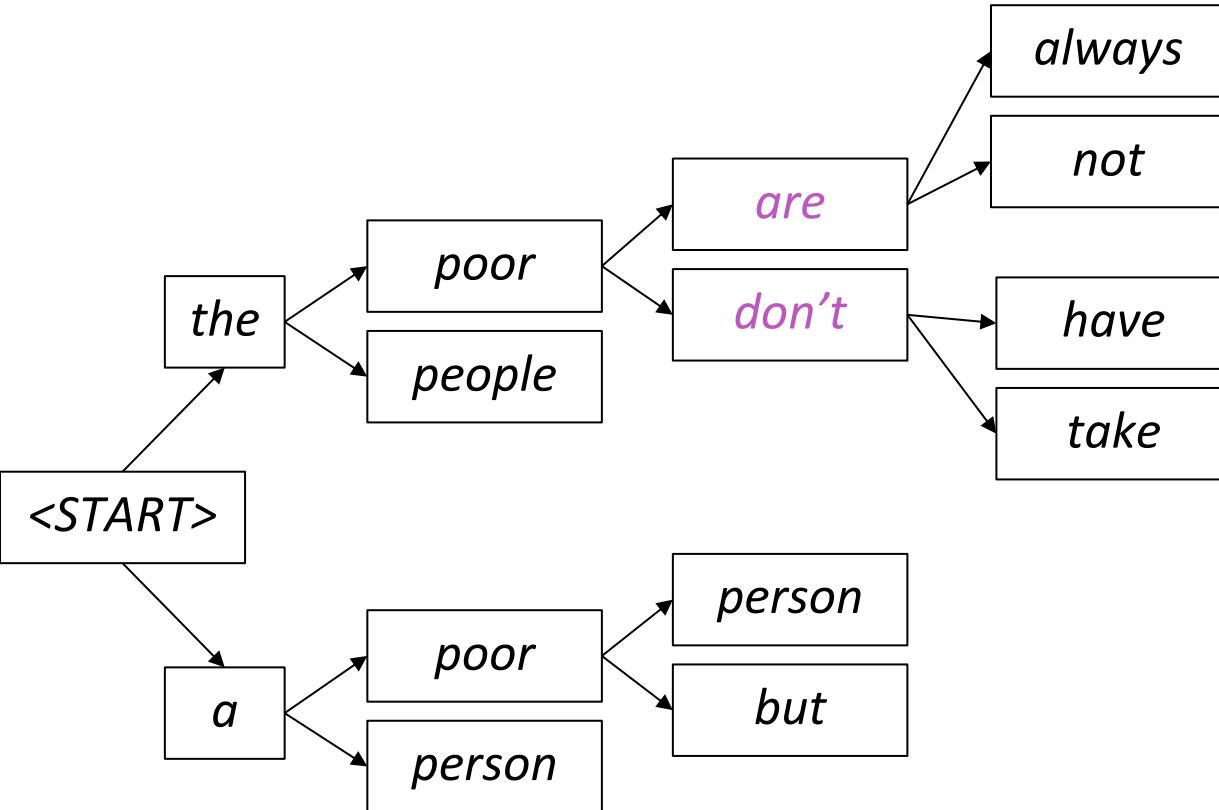
Beam search decoding: example

Beam size = 2



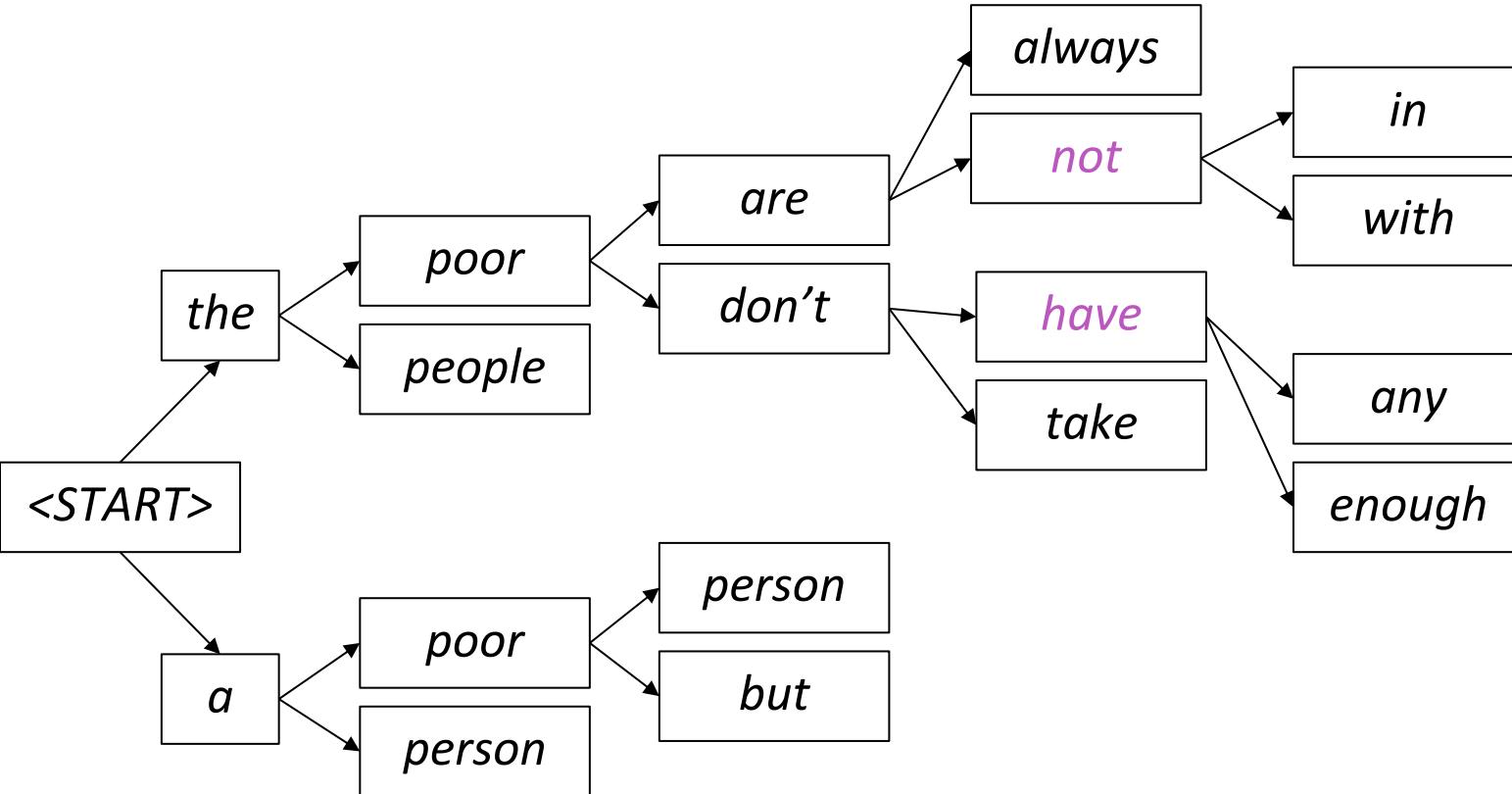
Beam search decoding: example

Beam size = 2



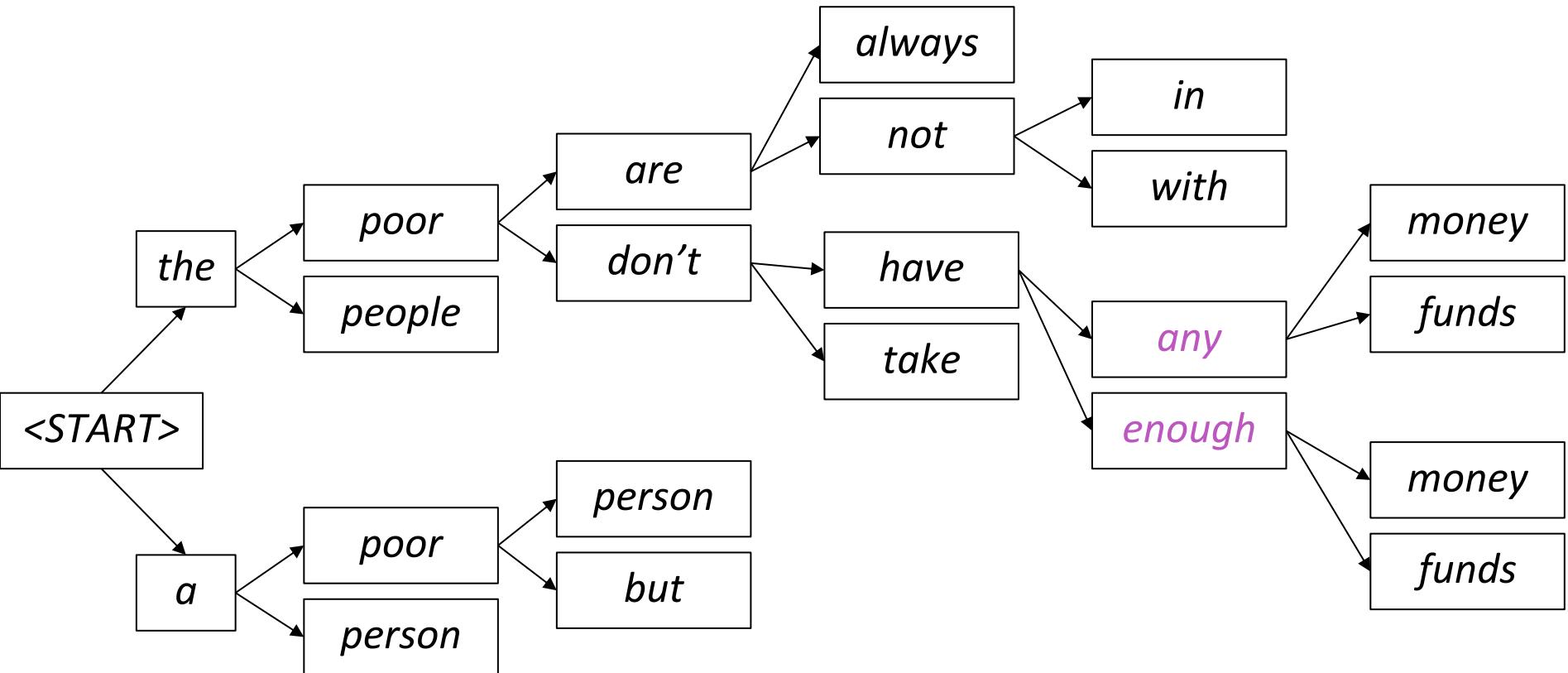
Beam search decoding: example

Beam size = 2



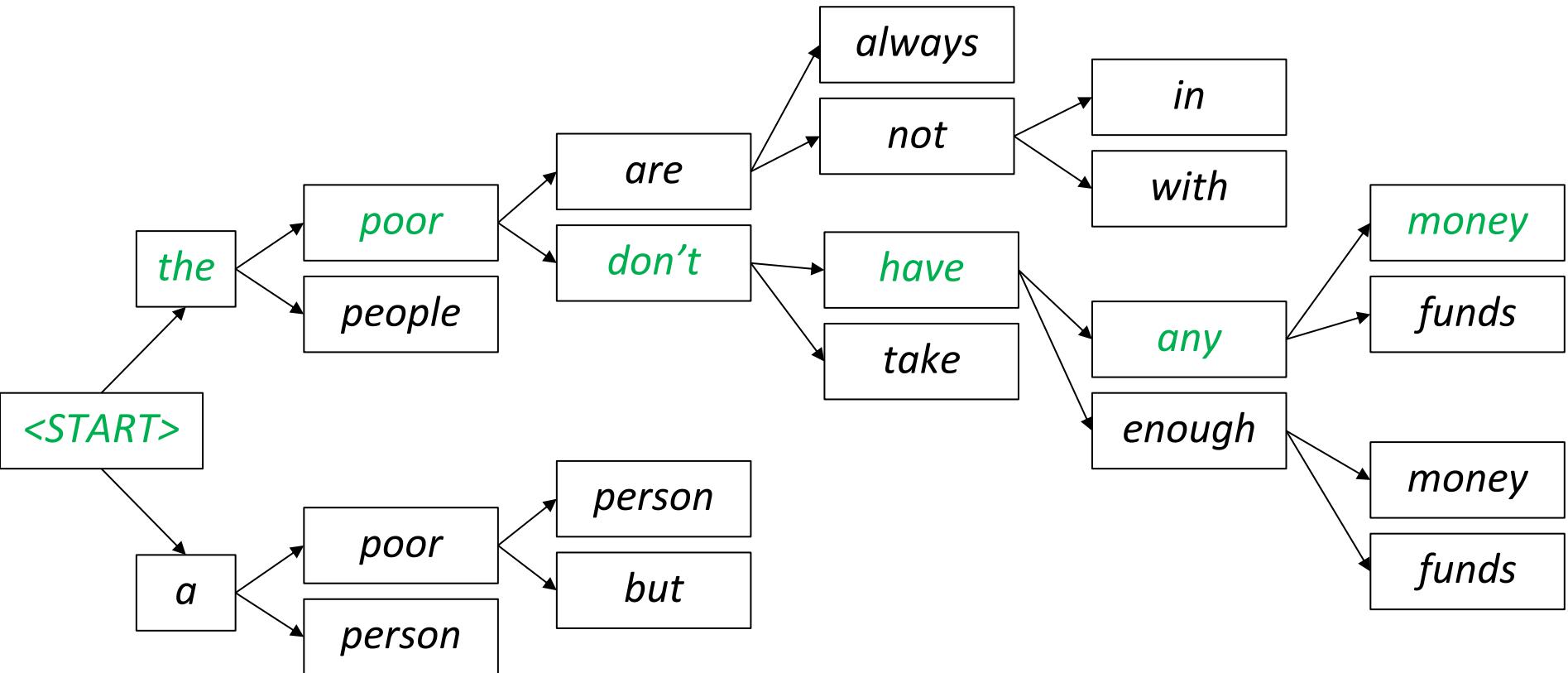
Beam search decoding: example

Beam size = 2



Beam search decoding: example

Beam size = 2



Advantages of NMT

Compared to SMT, NMT has many **advantages**:

- Better performance
 - More fluent
 - Better use of context
 - Better use of phrase similarities
- A single neural network to be optimized end-to-end
 - No subcomponents to be individually optimized
- Requires much less human engineering effort
 - No feature engineering
 - Same method for all language pairs

Disadvantages of NMT?

Compared to SMT:

- NMT is **less interpretable**
 - Hard to debug
- NMT is **difficult to control**
 - For example, can't easily specify rules or guidelines for translation
 - Safety concerns!

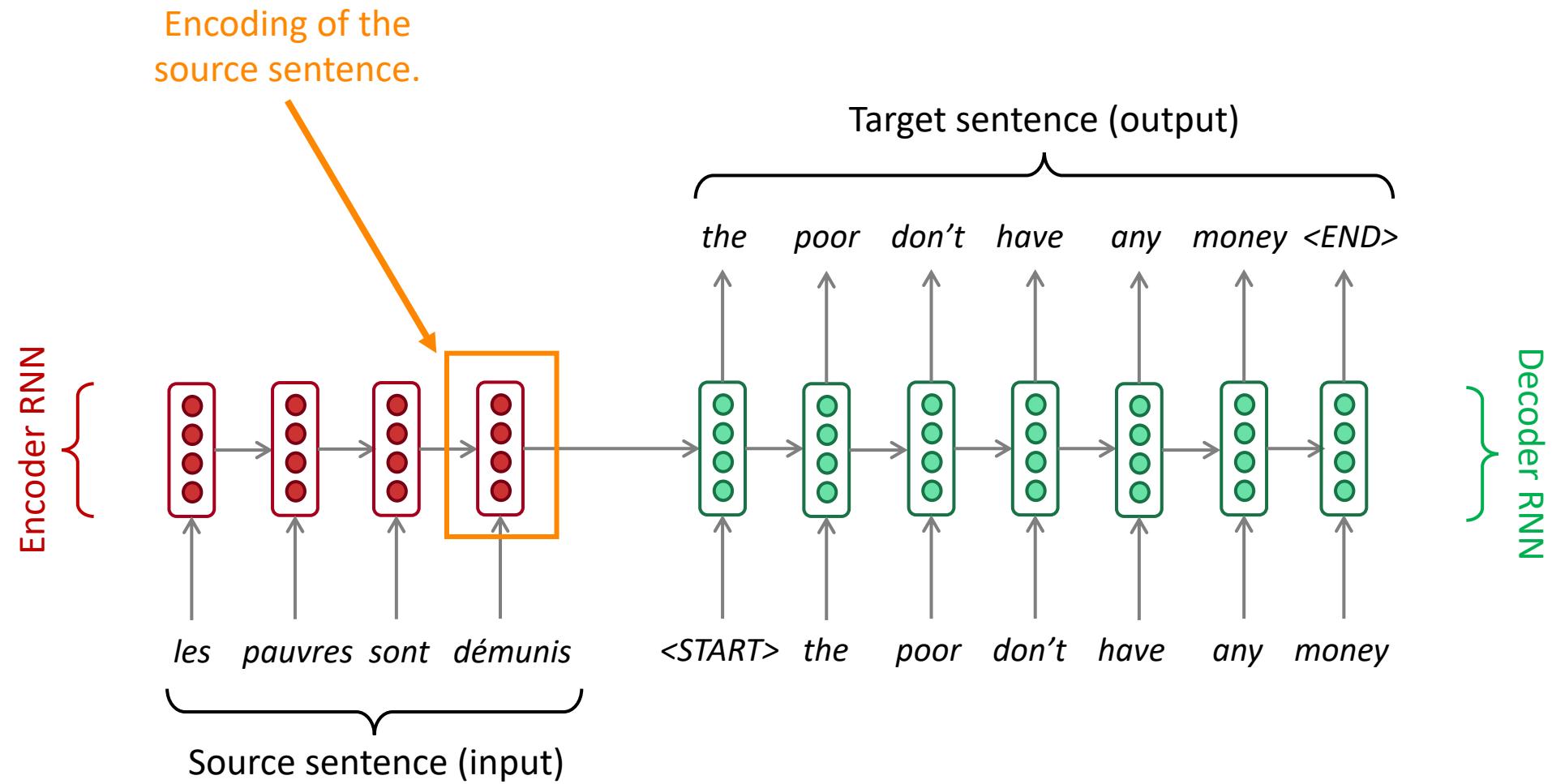
NMT research continues

NMT is the **flagship task** for NLP Deep Learning

- NMT research has **pioneered** many of the recent **innovations** of NLP Deep Learning
- In **2018**: NMT research continues to **thrive**
 - Researchers have found **many, many improvements** to the “vanilla” seq2seq NMT system we’ve presented today
 - But **one improvement** is so integral that it is the new vanilla...

ATTENTION

Sequence-to-sequence: the bottleneck problem



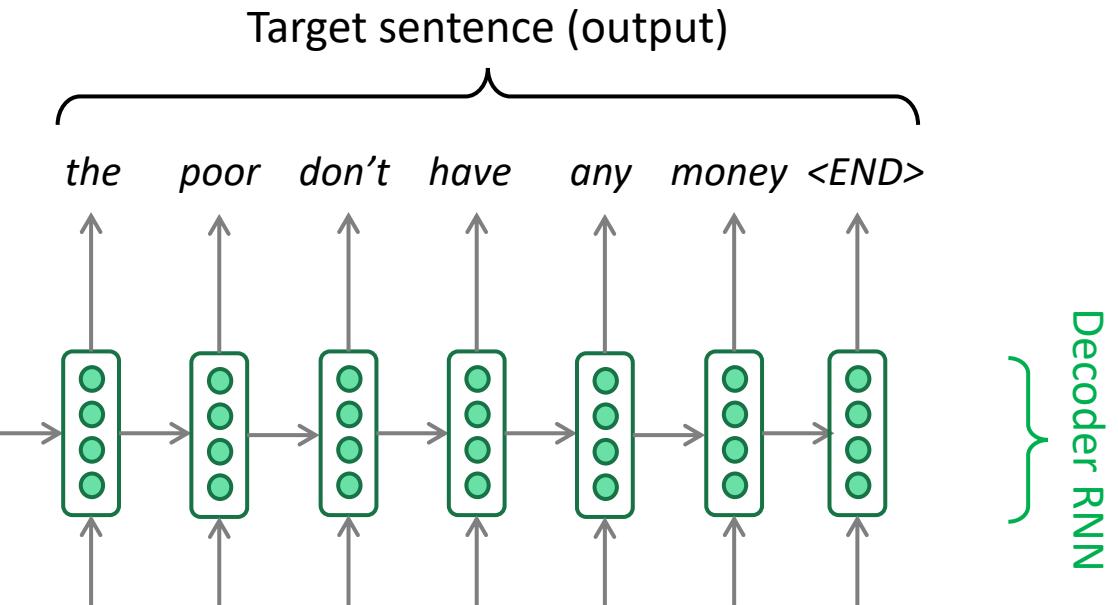
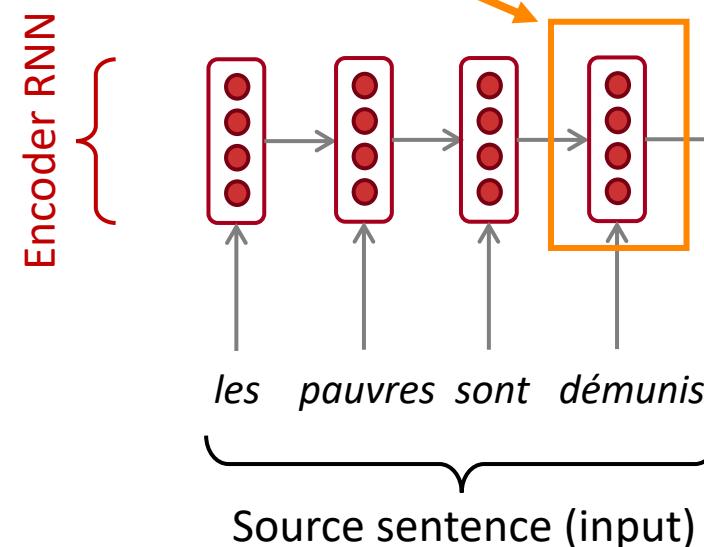
Problems with this architecture?

Sequence-to-sequence: the bottleneck problem

Encoding of the source sentence.

This needs to capture *all information* about the source sentence.

Information bottleneck!



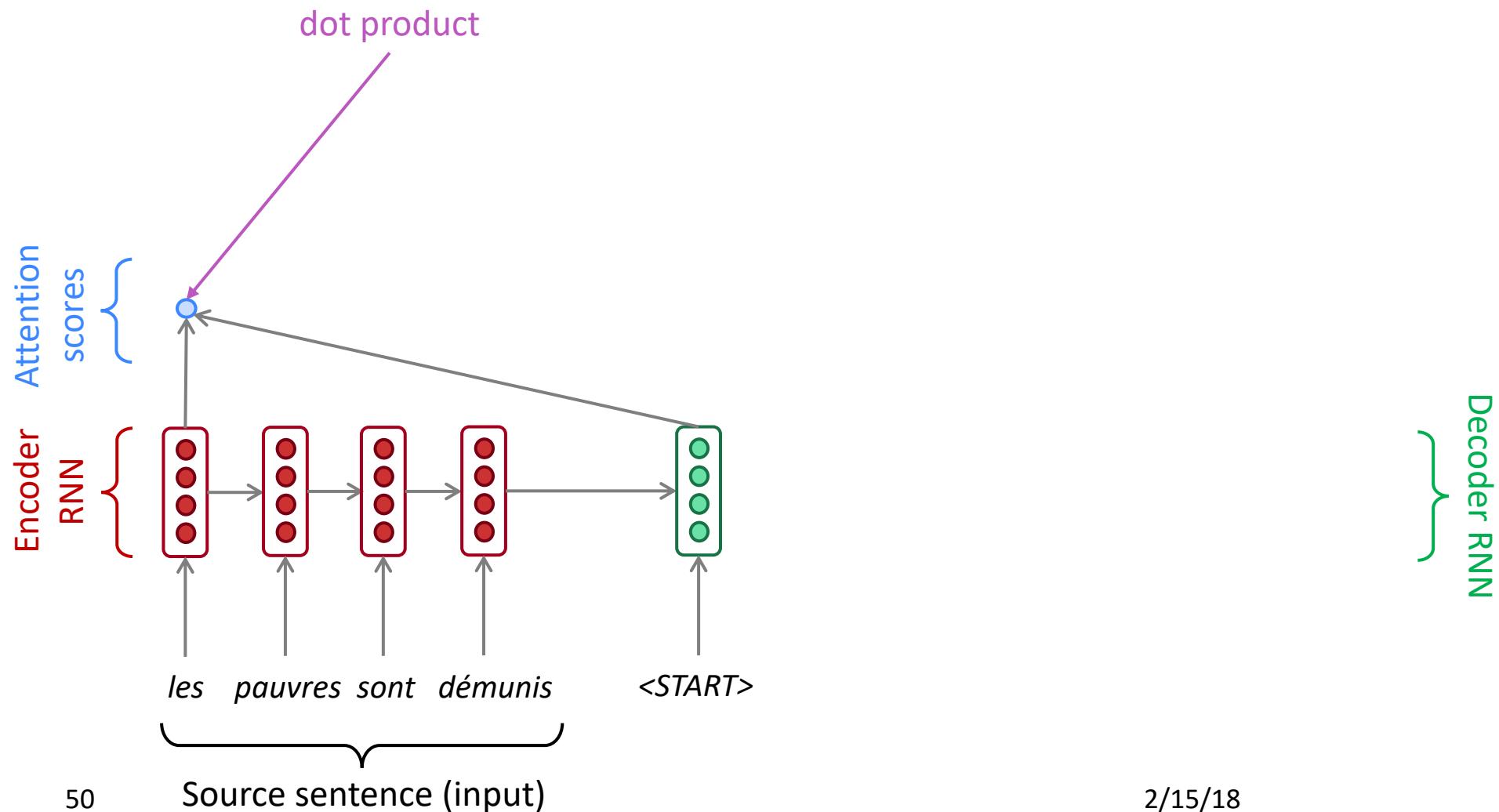
Attention

- **Attention** provides a solution to the bottleneck problem.
- Core idea: on each step of the decoder, *focus on a particular part* of the source sequence

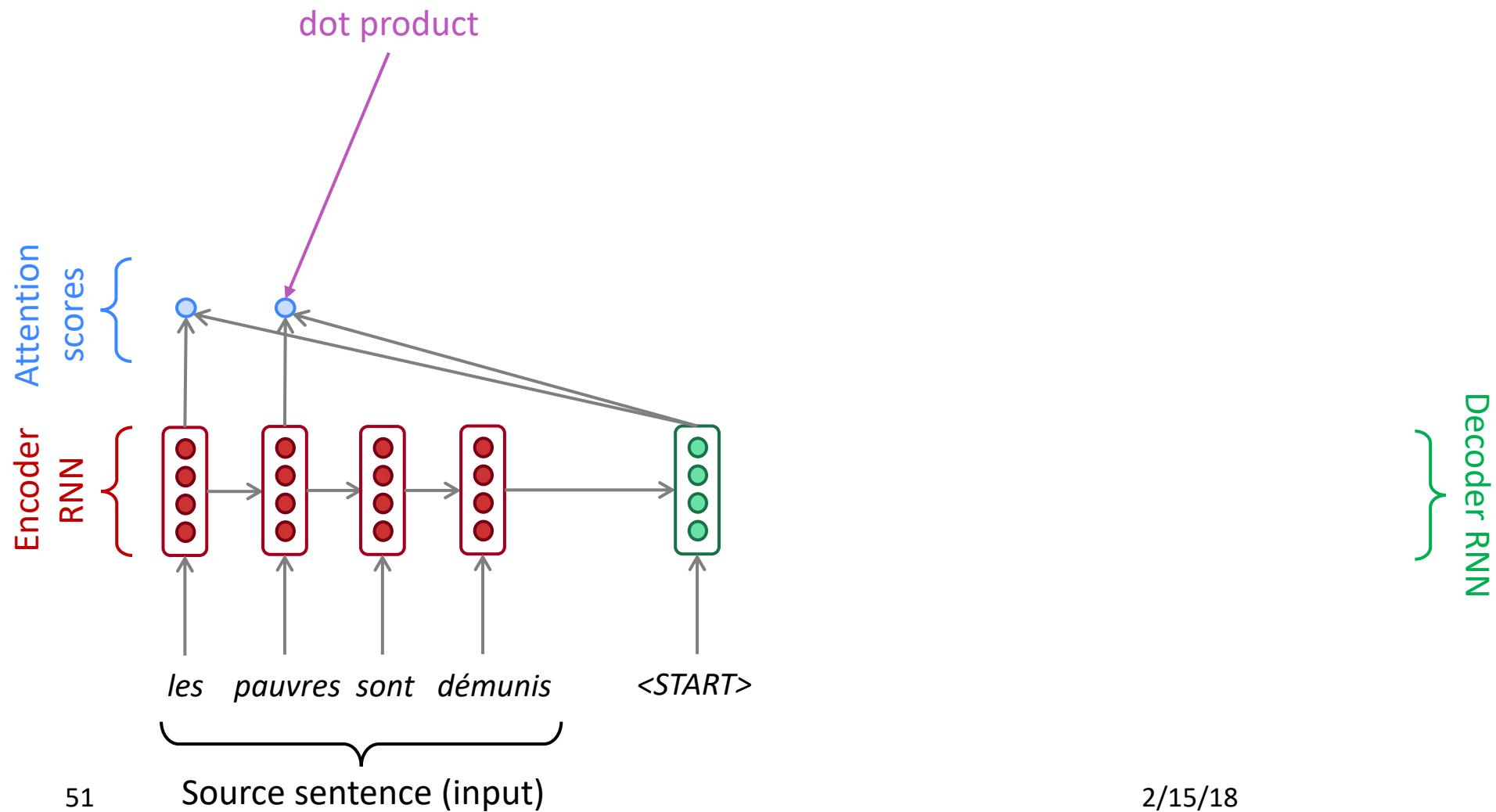


- First we will show via diagram (no equations), then we will show with equations

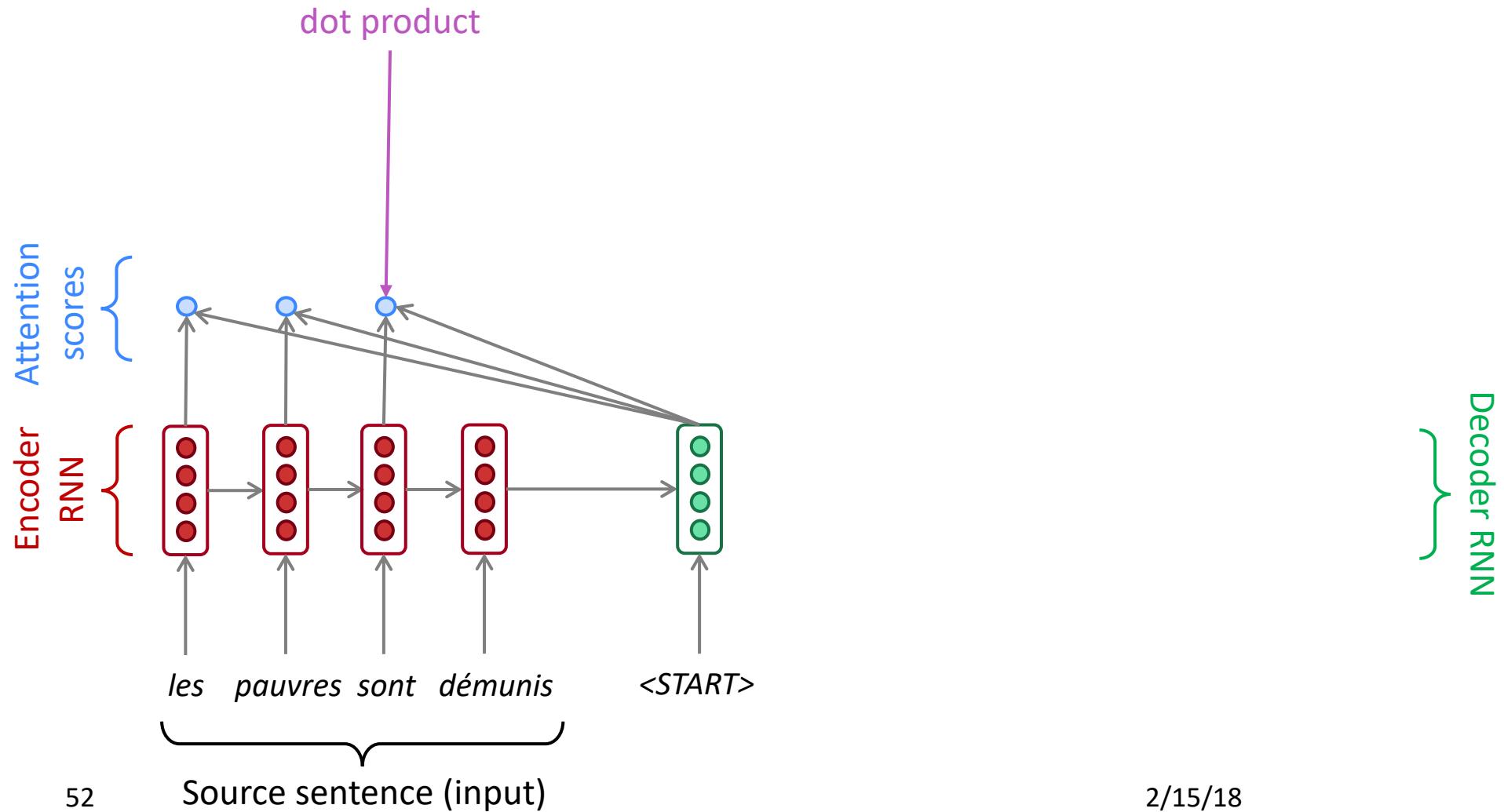
Sequence-to-sequence with attention



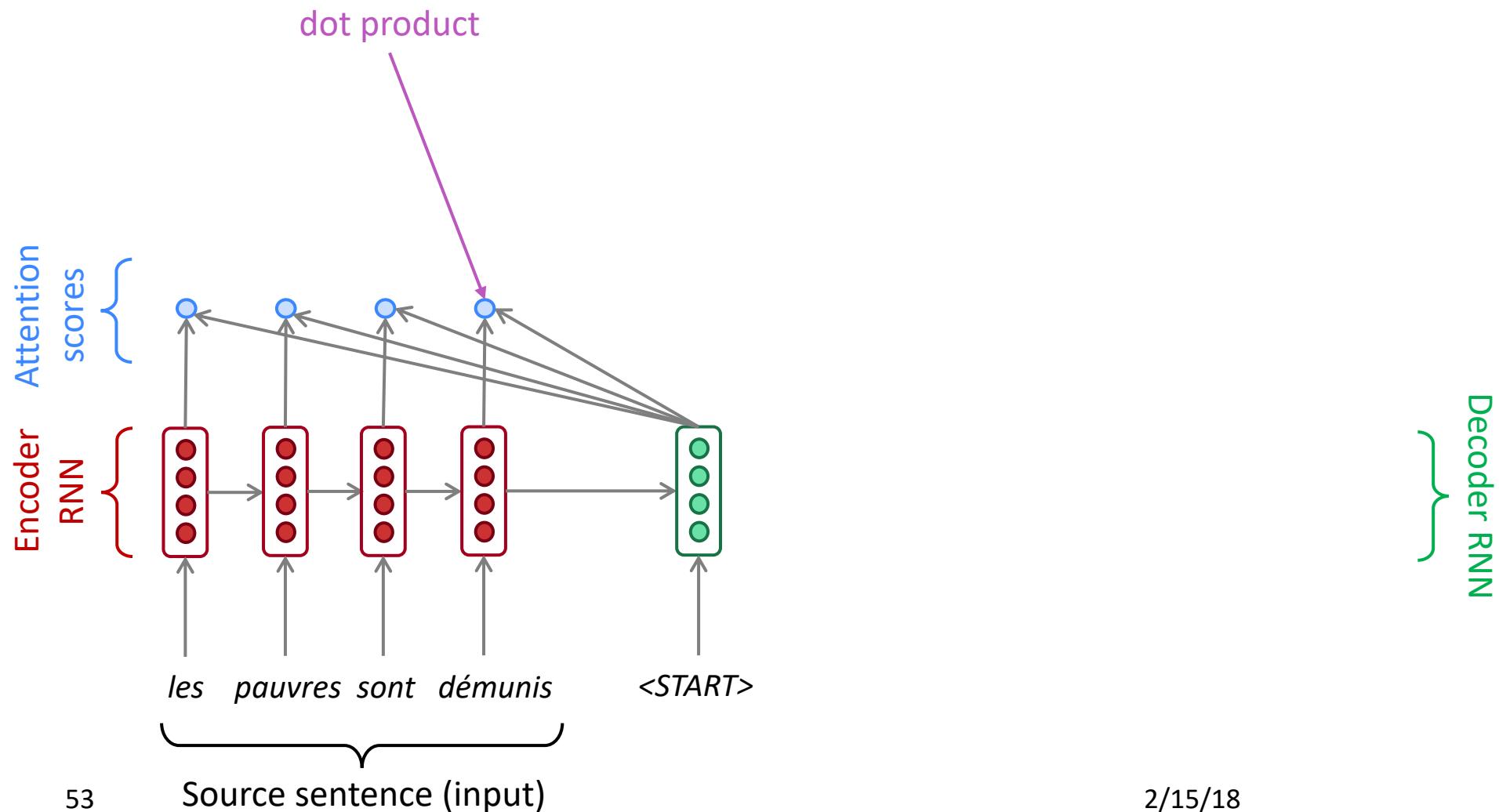
Sequence-to-sequence with attention



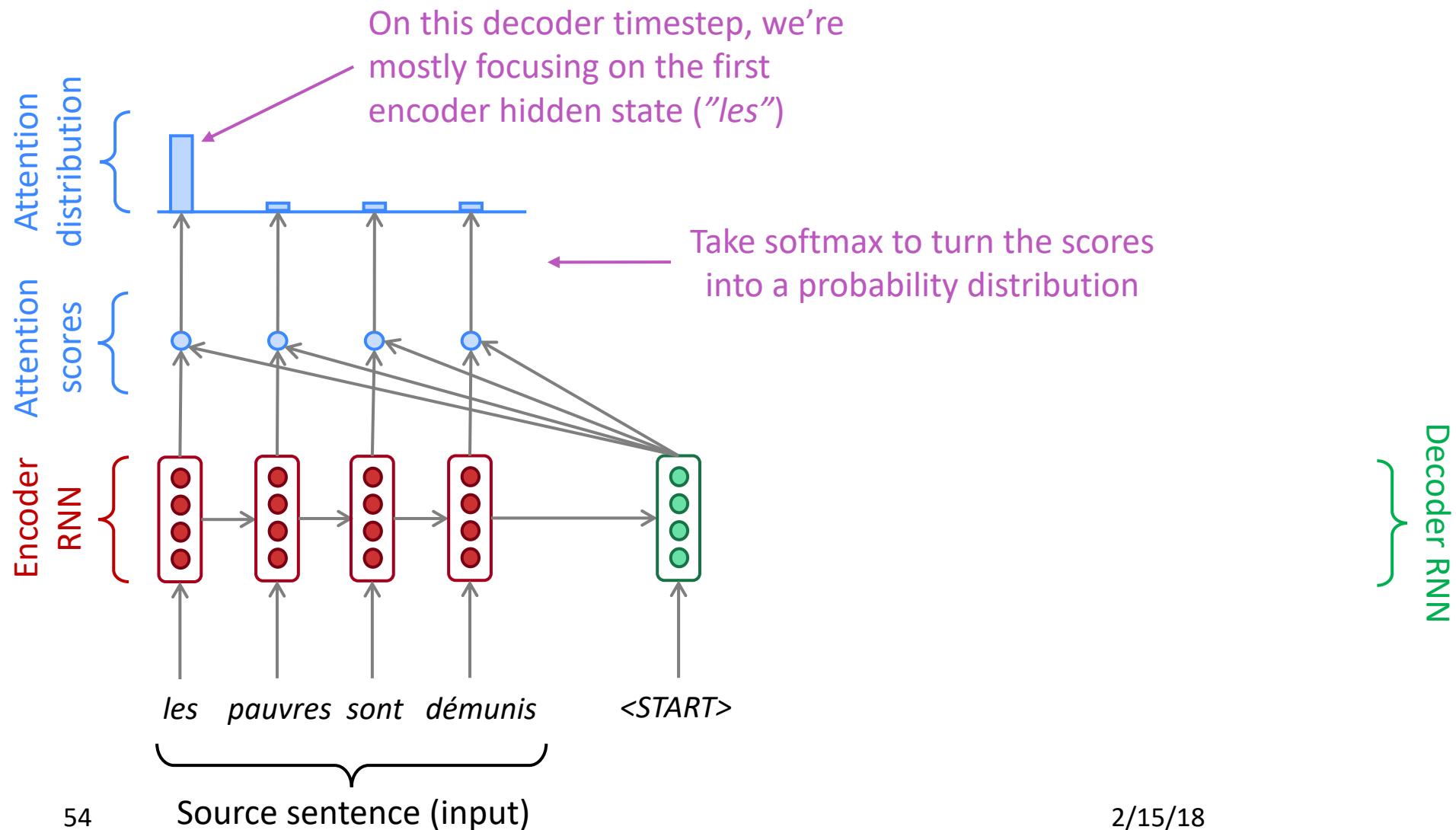
Sequence-to-sequence with attention



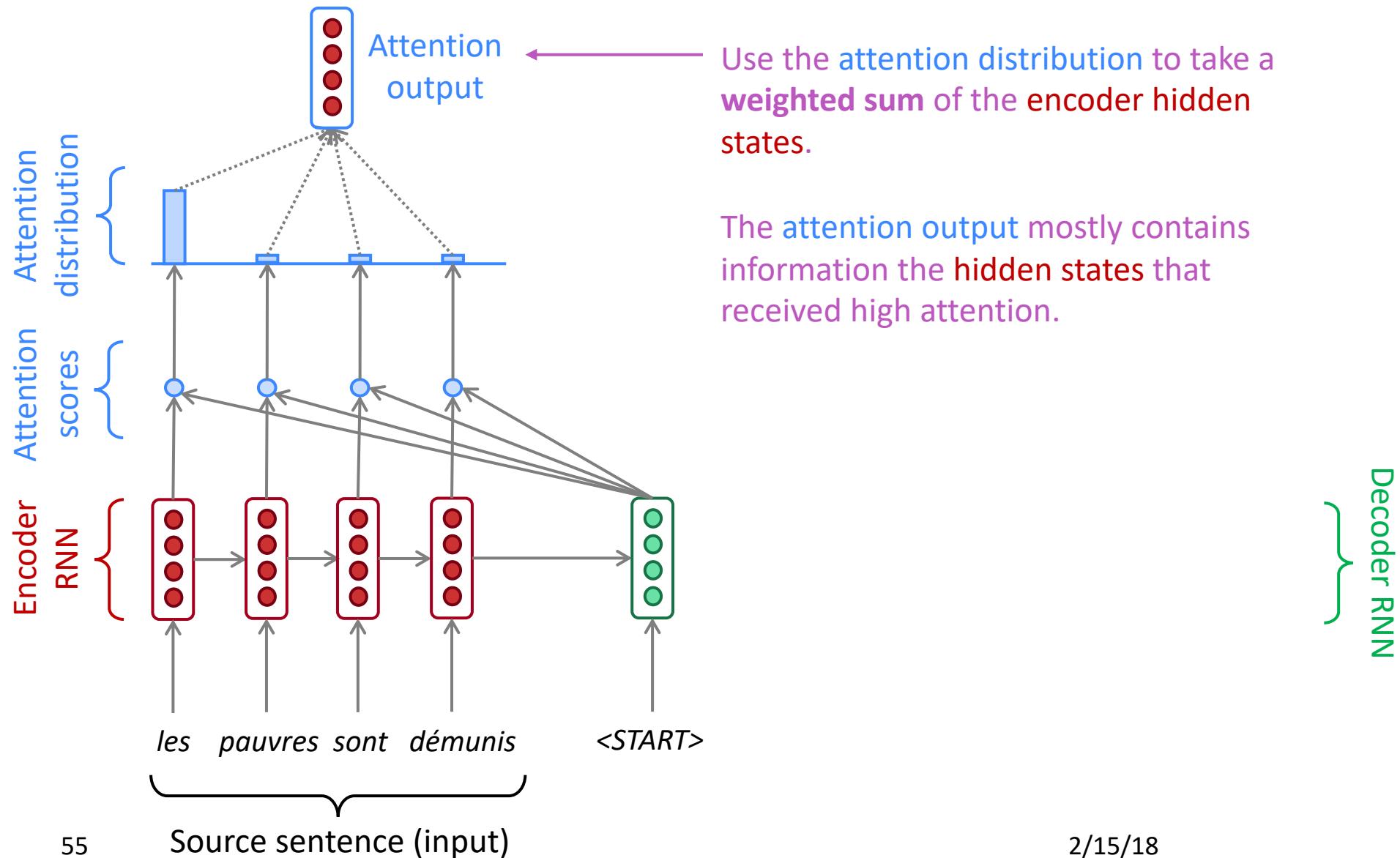
Sequence-to-sequence with attention



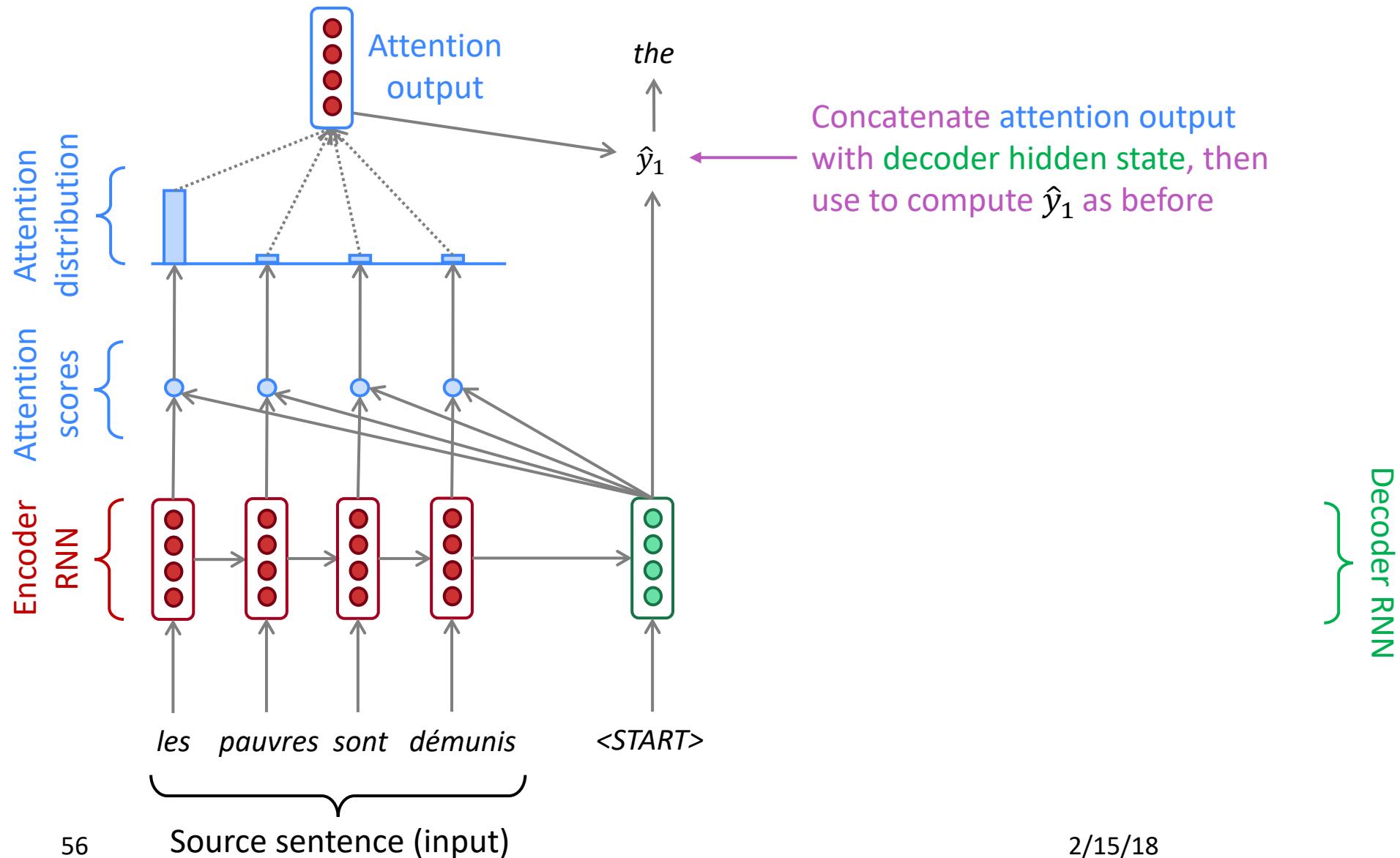
Sequence-to-sequence with attention



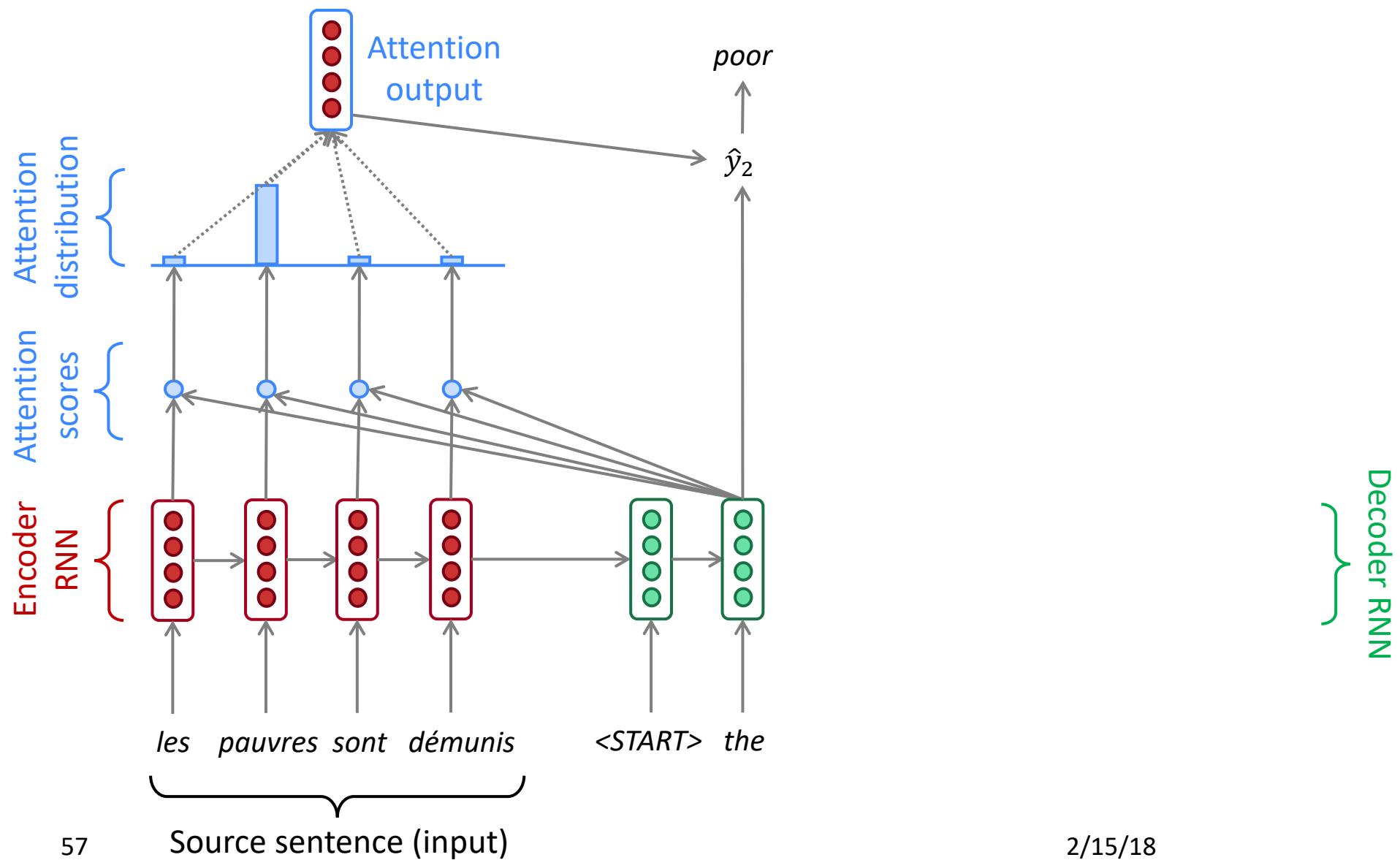
Sequence-to-sequence with attention



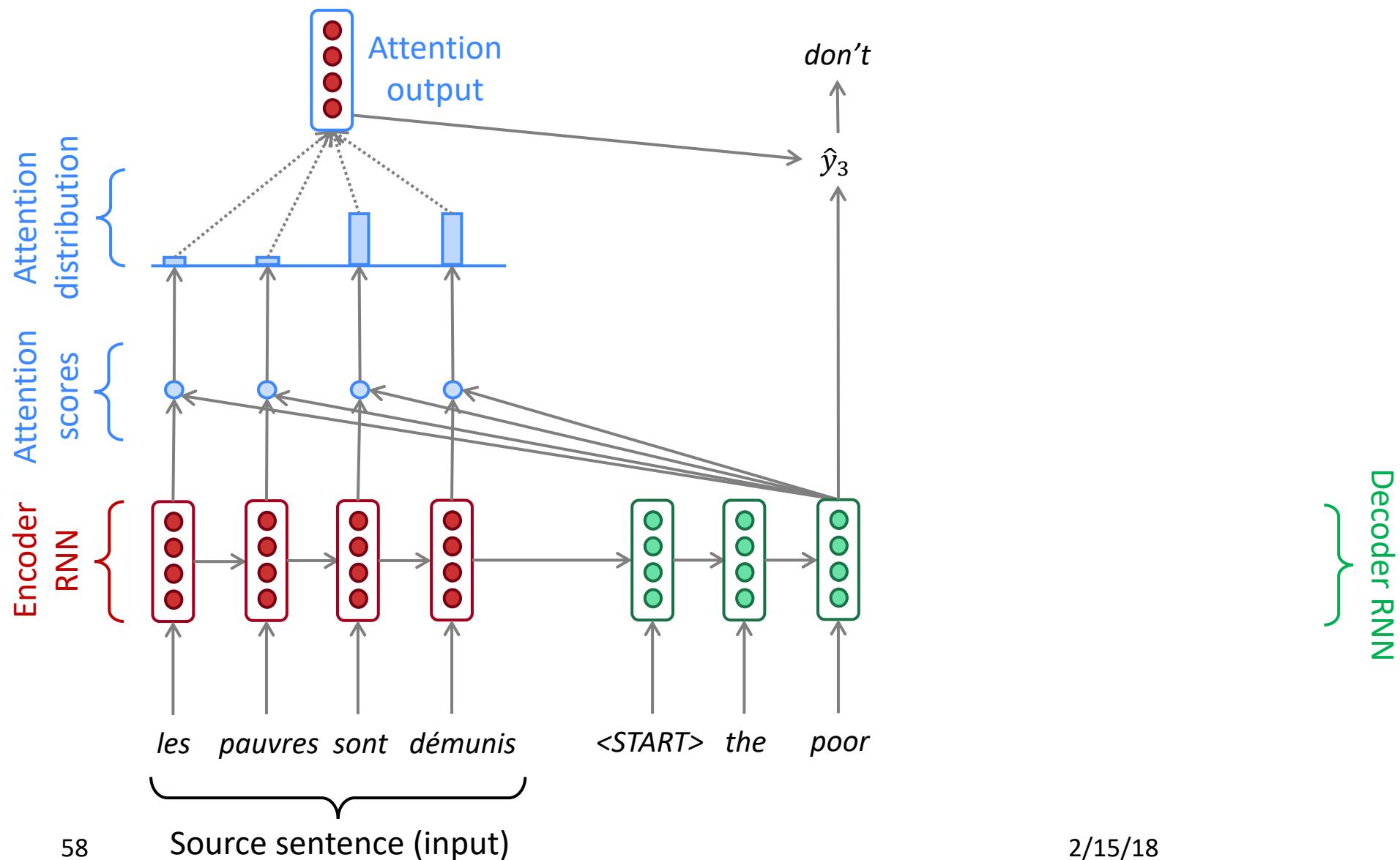
Sequence-to-sequence with attention



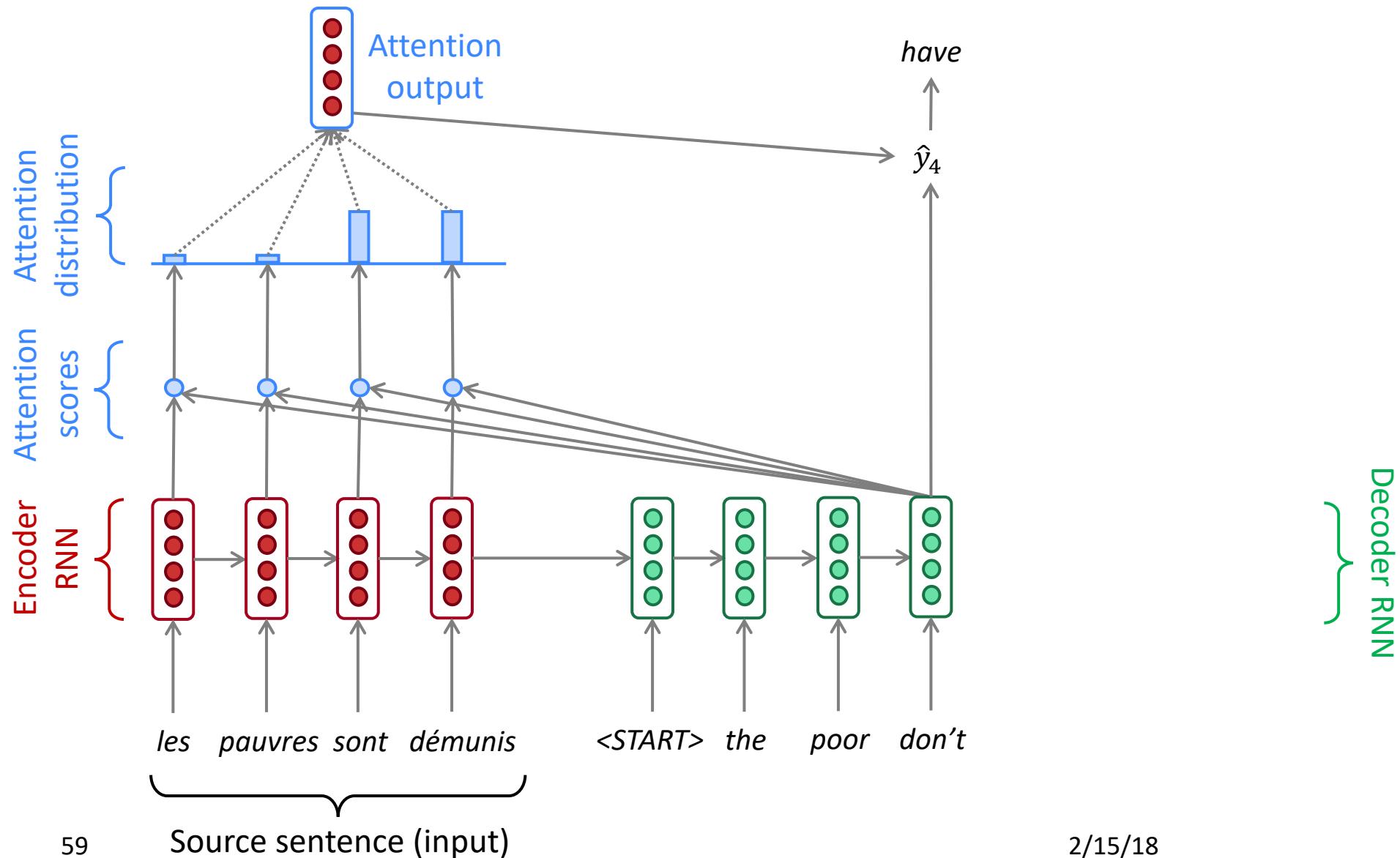
Sequence-to-sequence with attention



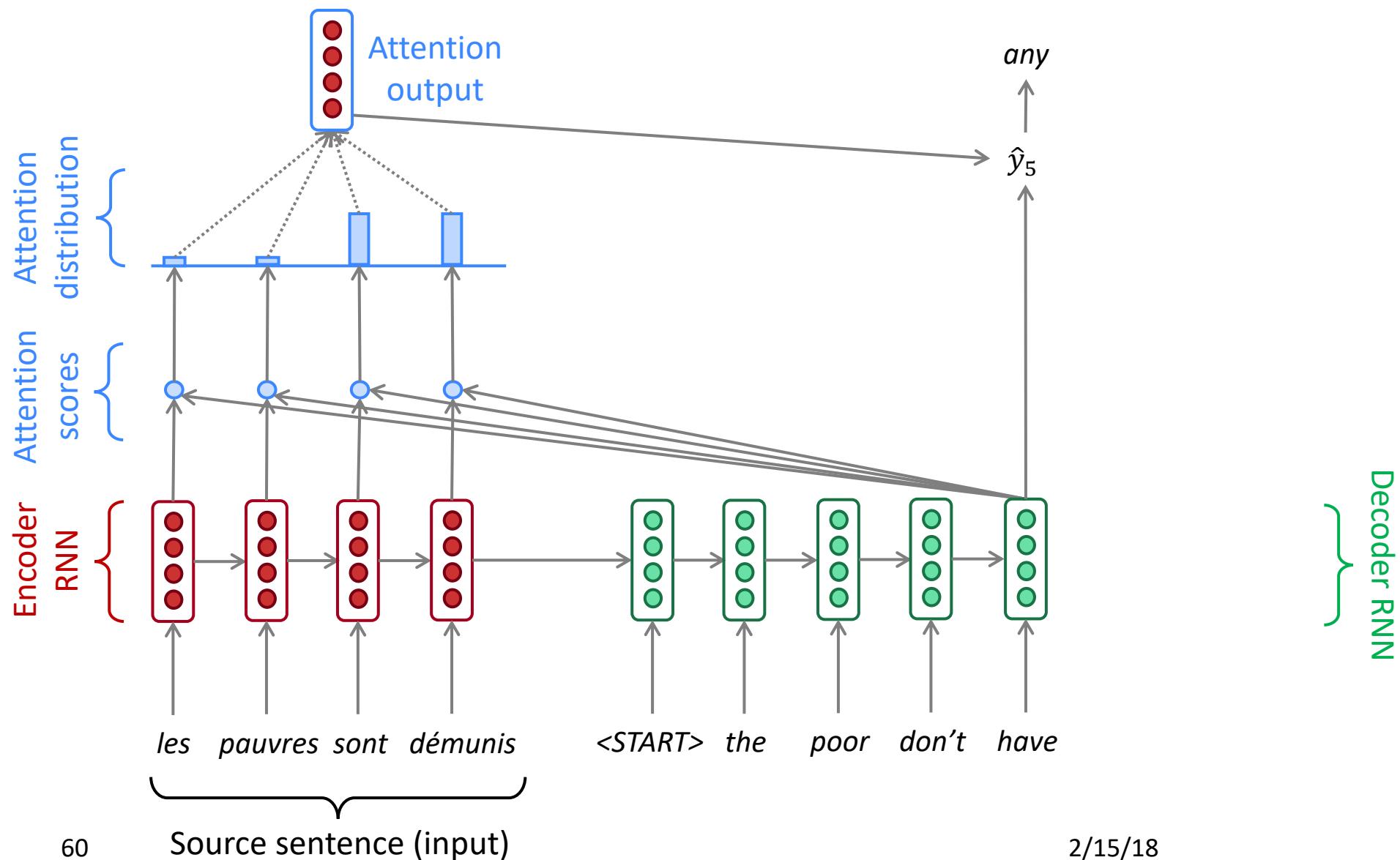
Sequence-to-sequence with attention



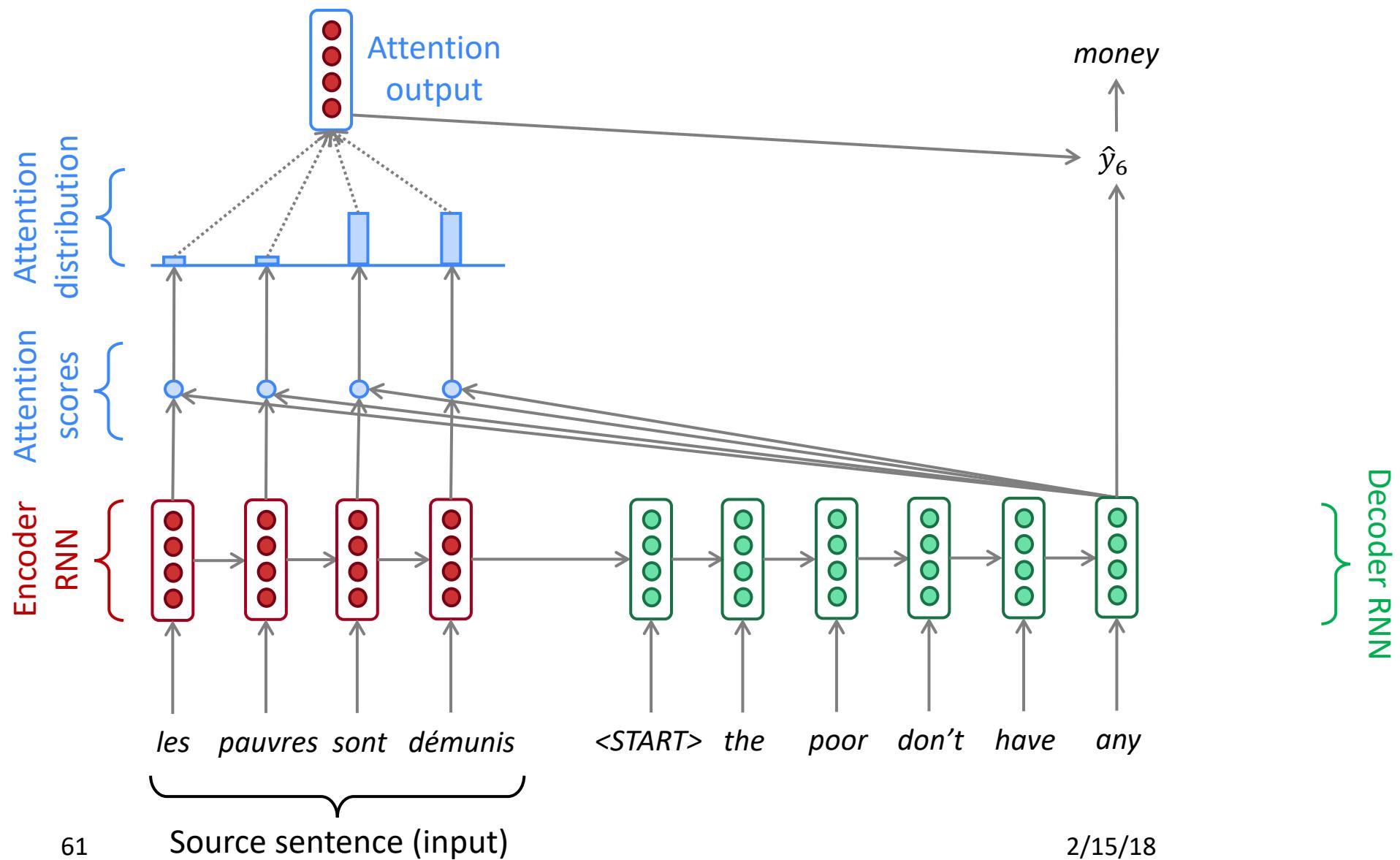
Sequence-to-sequence with attention



Sequence-to-sequence with attention



Sequence-to-sequence with attention



Attention: in equations

- We have encoder hidden states $h_1, \dots, h_N \in \mathbb{R}^h$
- On timestep t , we have decoder hidden state $s_t \in \mathbb{R}^h$
- We get the attention scores e^t for this step:

$$e^t = [s_t^T h_1, \dots, s_t^T h_N] \in \mathbb{R}^N$$

- We take softmax to get the attention distribution α^t for this step (this is a probability distribution and sums to 1)

$$\alpha^t = \text{softmax}(e^t) \in \mathbb{R}^N$$

- We use α^t to take a weighted sum of the encoder hidden states to get the attention output a_t

$$a_t = \sum_{i=1}^N \alpha_i^t h_i \in \mathbb{R}^h$$

- Finally we concatenate the attention output a_t with the decoder hidden state s_t and proceed as in the non-attention seq2seq model

$$[a_t; s_t] \in \mathbb{R}^{2h}$$

Attention is great

- Attention significantly improves NMT performance
 - It's very useful to allow decoder to focus on certain parts of the source
- Attention solves the bottleneck problem
 - Attention allows decoder to look directly at source; bypass bottleneck
- Attention helps with vanishing gradient problem
 - Provides shortcut to faraway states
- Attention provides some interpretability
 - By inspecting attention distribution, we can see what the decoder was focusing on
 - We get alignment for free!
 - This is cool because we never explicitly trained an alignment system
 - The network just learned alignment by itself

