

Attention Network

- Stanford CS 224n Lecture 10, 11, 12 slides
- Stanford CS 231n Lecture 10 slides

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*

2014

Neural
Machine
Translation

MT research

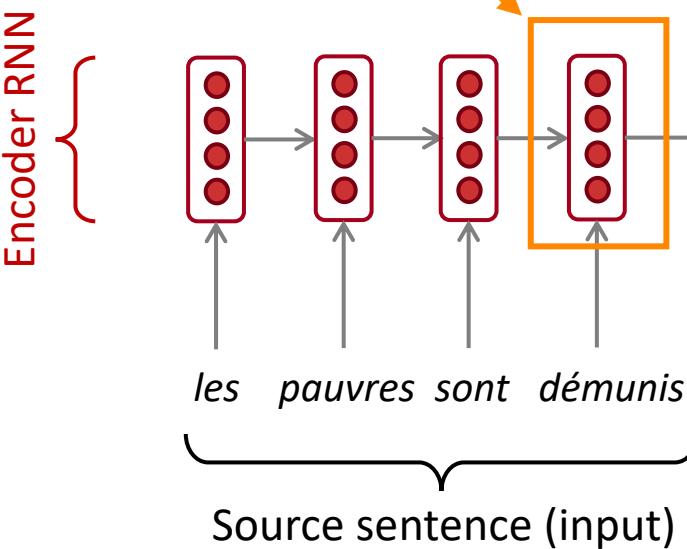
(dramatic reenactment)

Neural Machine Translation (NMT)

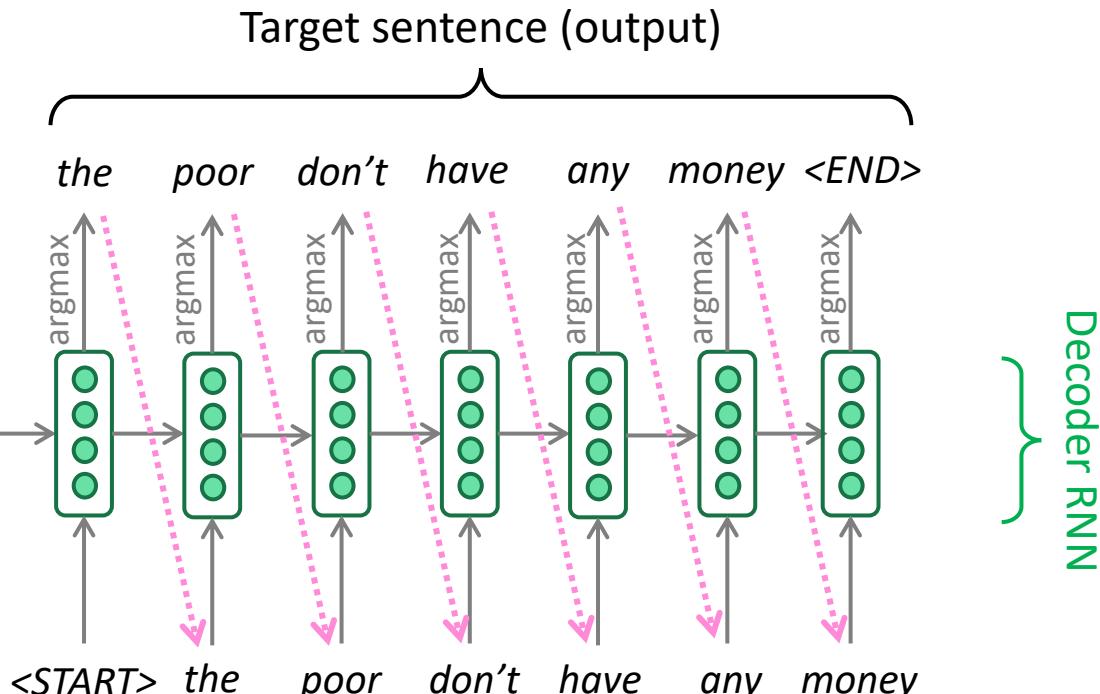
The sequence-to-sequence model

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

Image Captioning

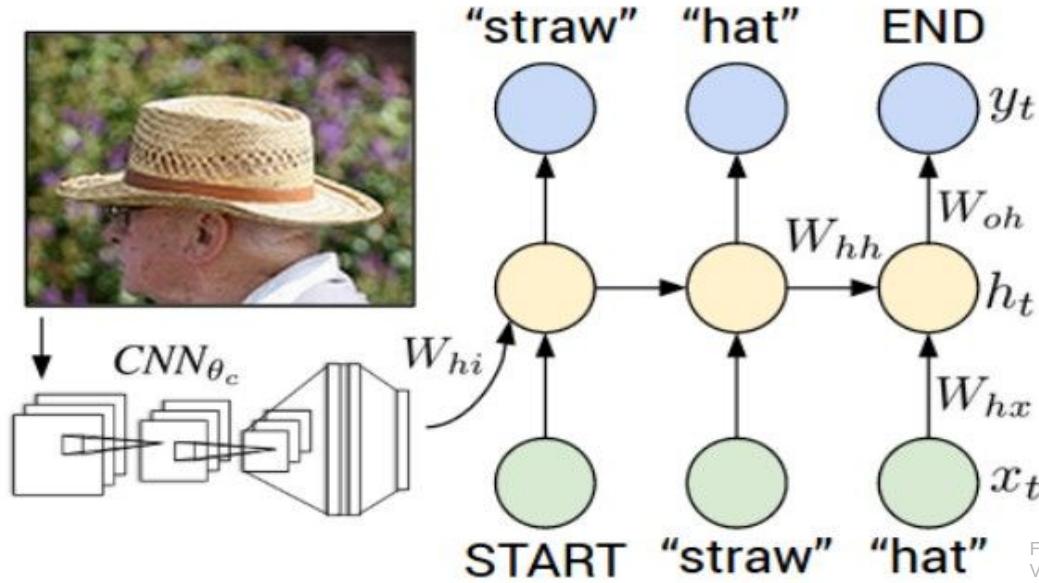


Figure from Karpathy et al, "Deep Visual-Semantic Alignments for Generating Image Descriptions", CVPR 2015; figure copyright IEEE, 2015.
Reproduced for educational purposes.

Explain Images with Multimodal Recurrent Neural Networks, Mao et al.

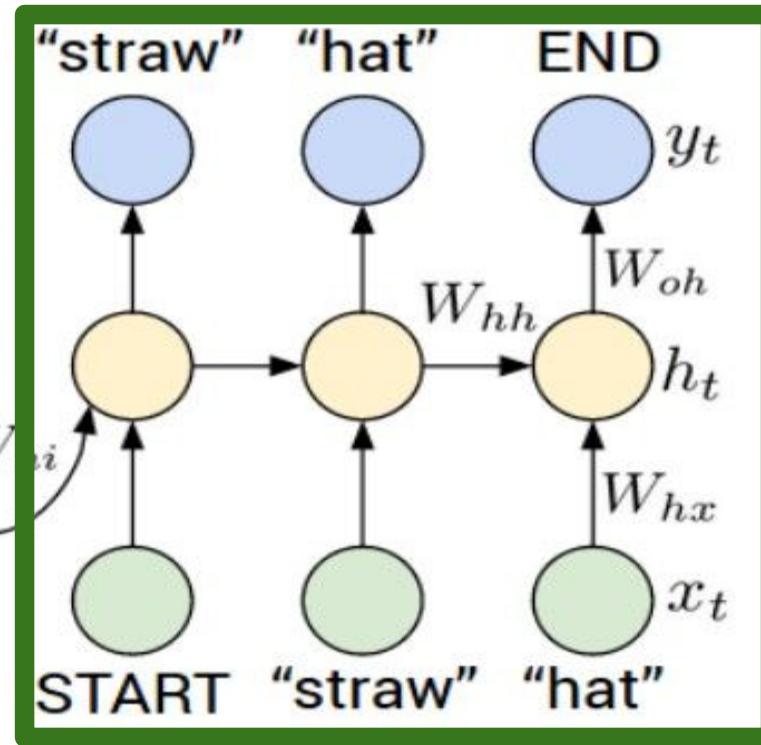
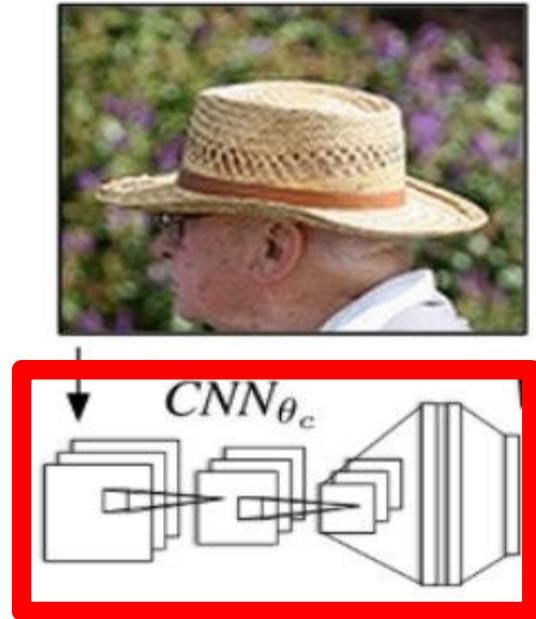
Deep Visual-Semantic Alignments for Generating Image Descriptions, Karpathy and Fei-Fei

Show and Tell: A Neural Image Caption Generator, Vinyals et al.

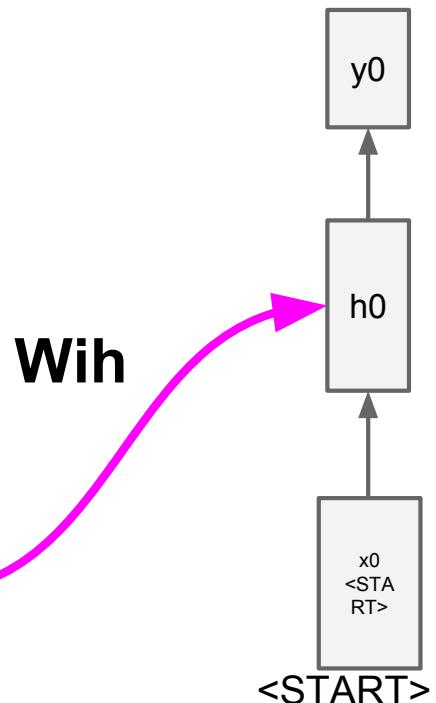
Long-term Recurrent Convolutional Networks for Visual Recognition and Description, Donahue et al.

Learning a Recurrent Visual Representation for Image Caption Generation, Chen and Zitnick

Recurrent Neural Network



Convolutional Neural Network



test image

$$a_t = g(W_{aa} \cdot a_{t-1} + W_{ax}(x_t) + b_a)$$

before:

$$h = \tanh(W_{xh} * x + W_{hh} * h)$$

now:

$$h = \tanh(W_{xh} * x + W_{hh} * h + W_{ih} * v)$$

Many difficulties remains in MT

- Out-of-vocabulary words
- Domain mismatch between train and test data
- Maintaining context over longer text
- Low-resource language pairs
- Using common sense is still hard
- NMT picks up biases in training data
-

• ATTENTION

$$f(x)$$

$$f(x)$$

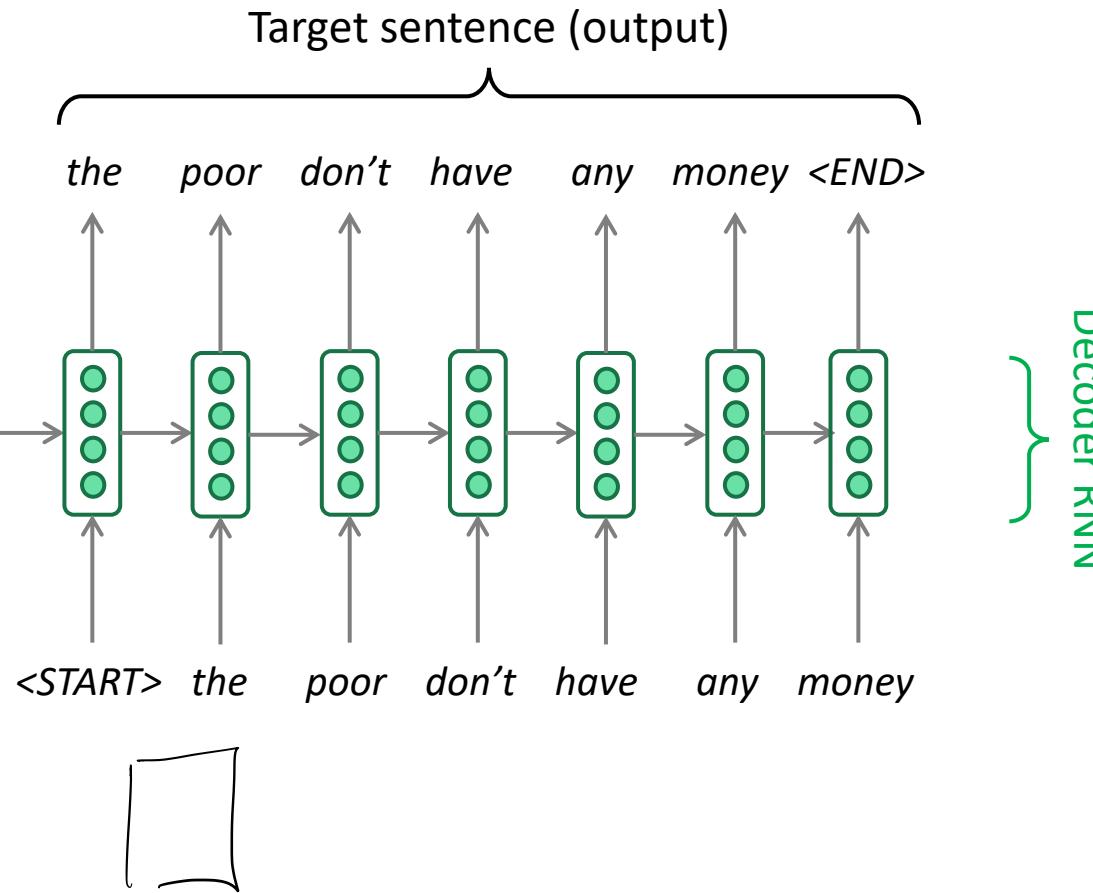
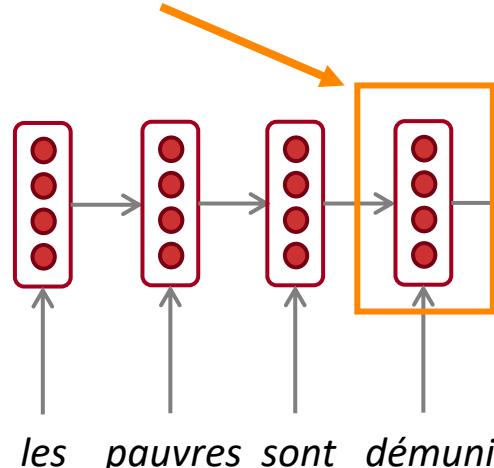
$$(f(x) - g(x))$$

$$y = f(x) + \epsilon$$

g

Sequence-to-sequence: the bottleneck problem

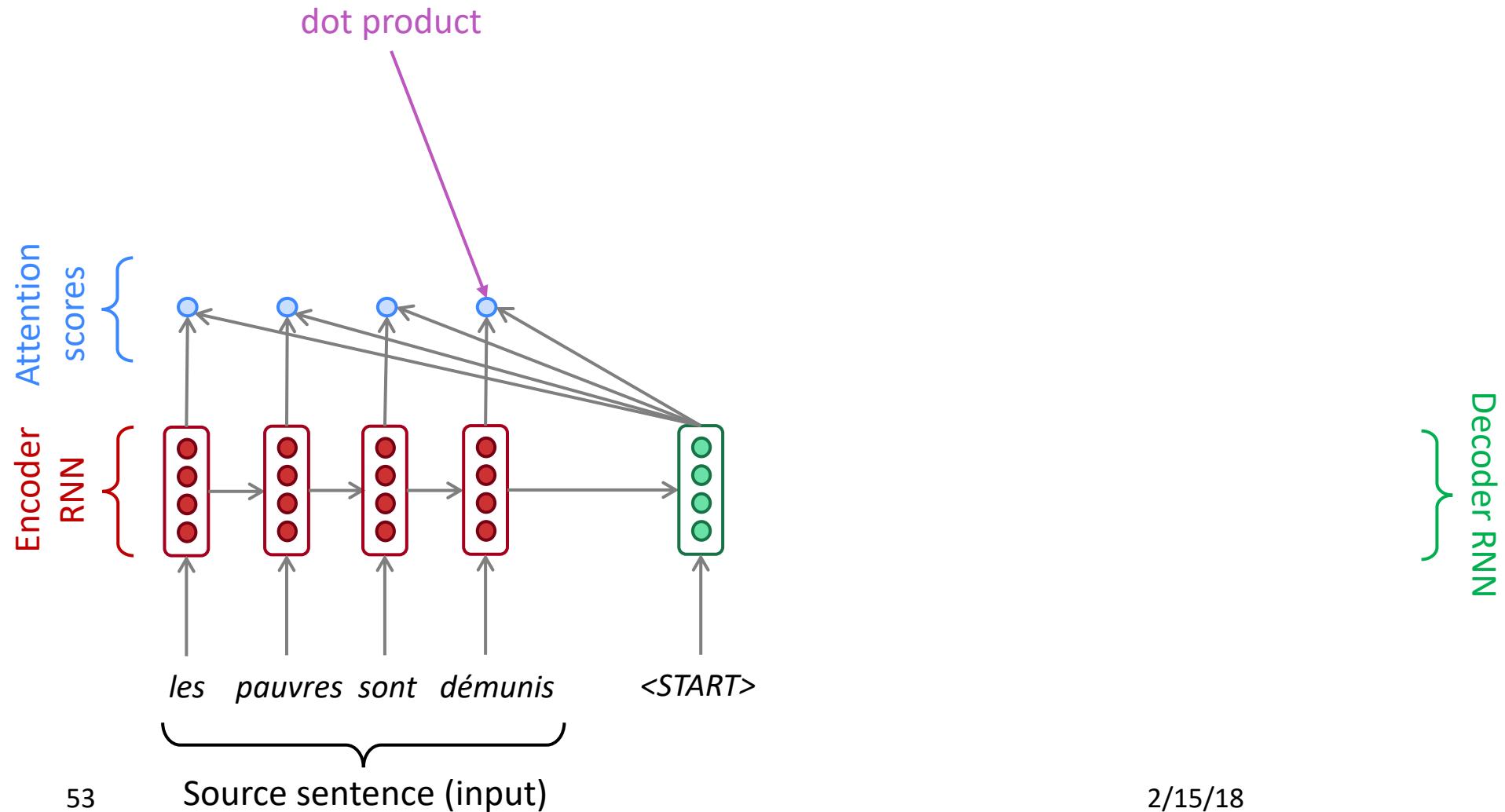
Encoding of the source sentence.
This needs to capture *all* information about the source sentence.
Information bottleneck!



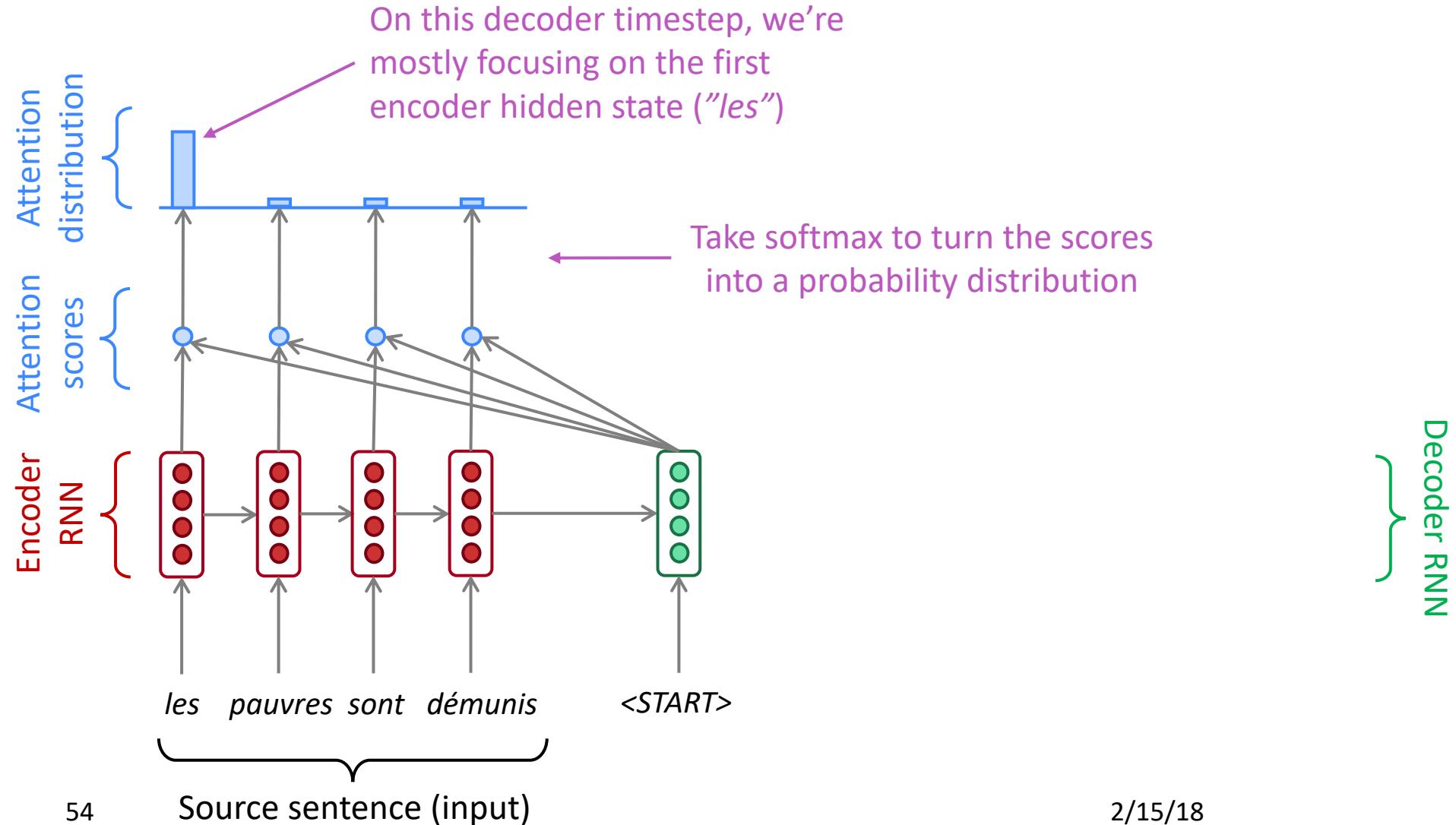
Attention

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

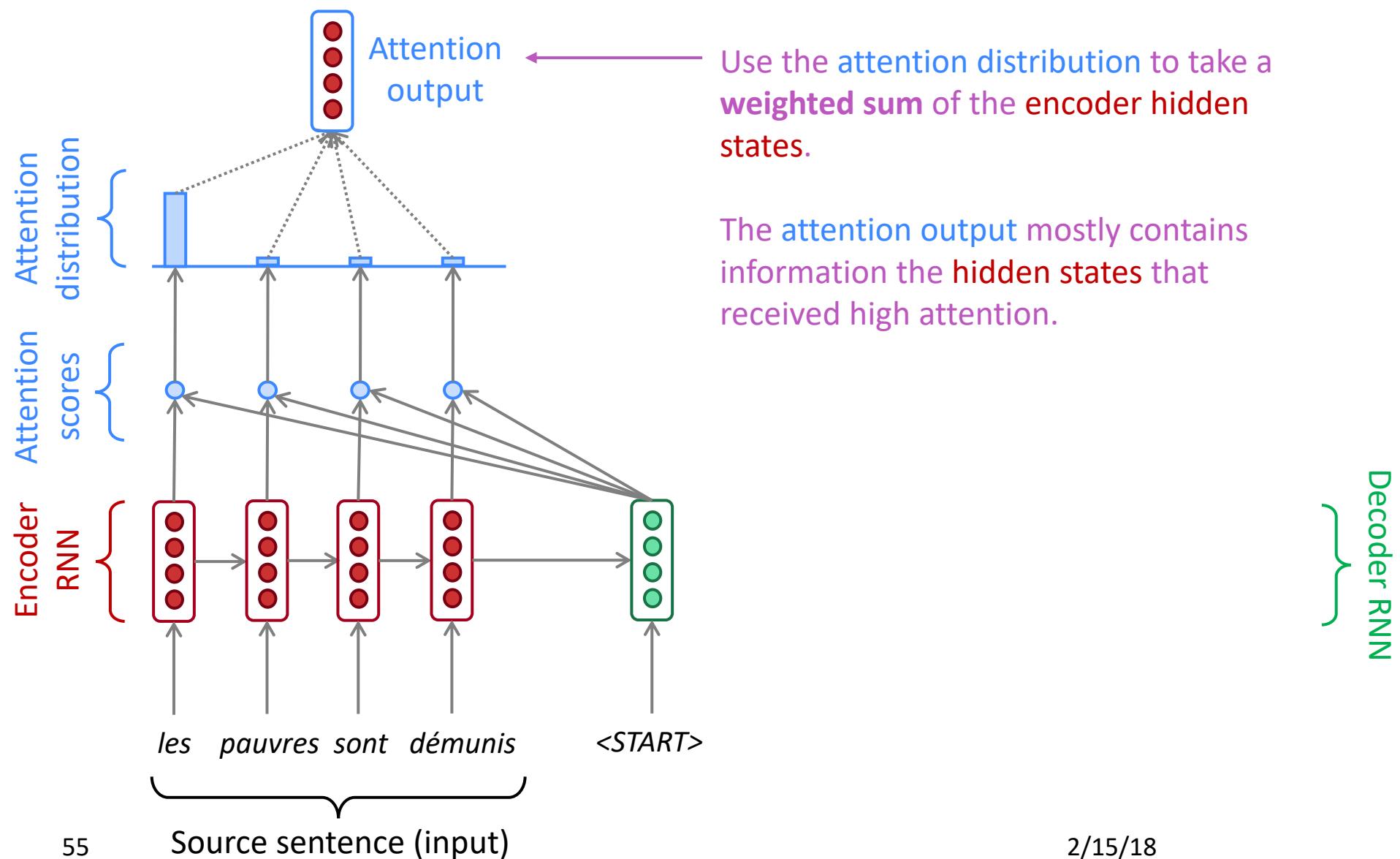
Sequence-to-sequence with attention



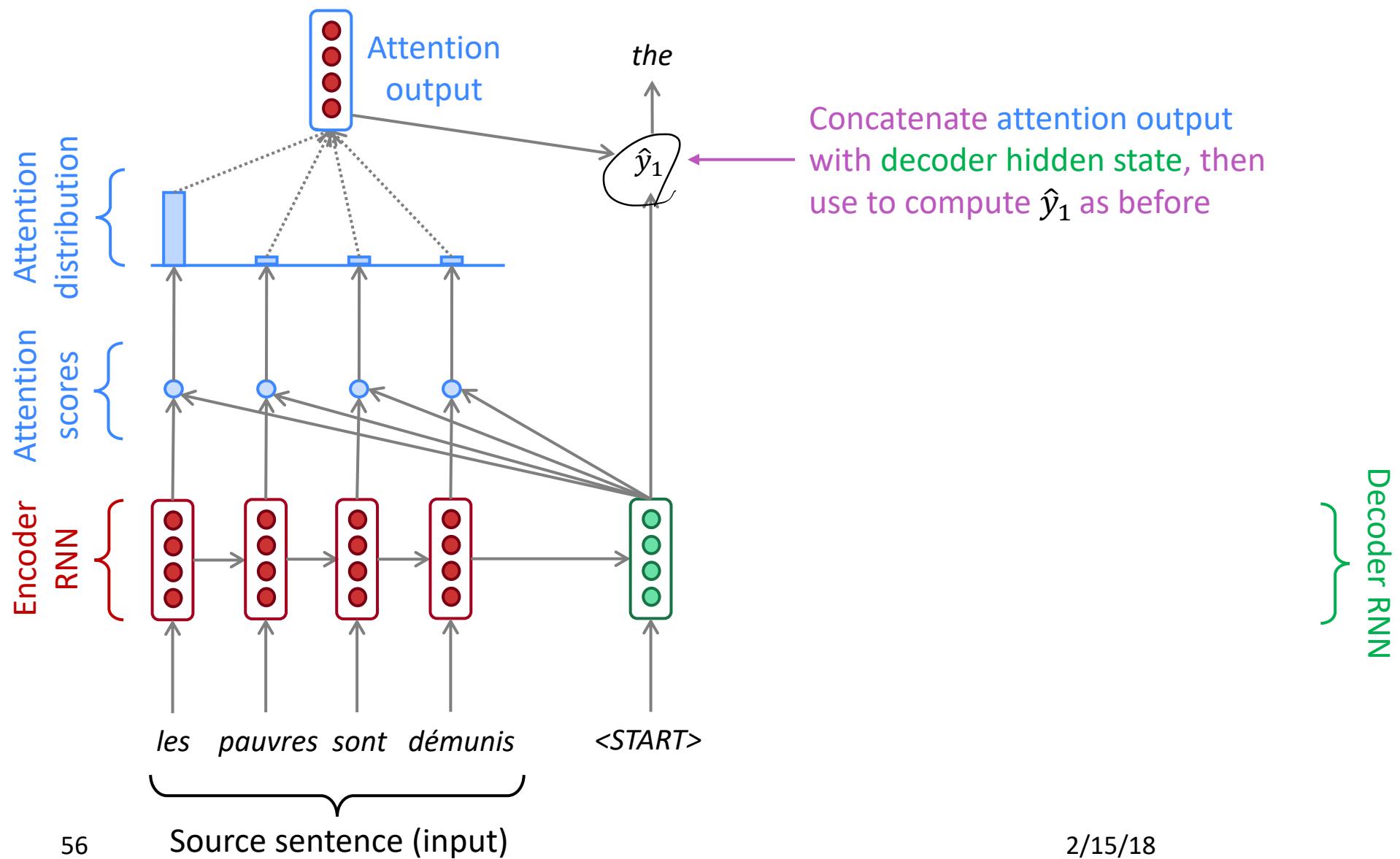
Sequence-to-sequence with attention



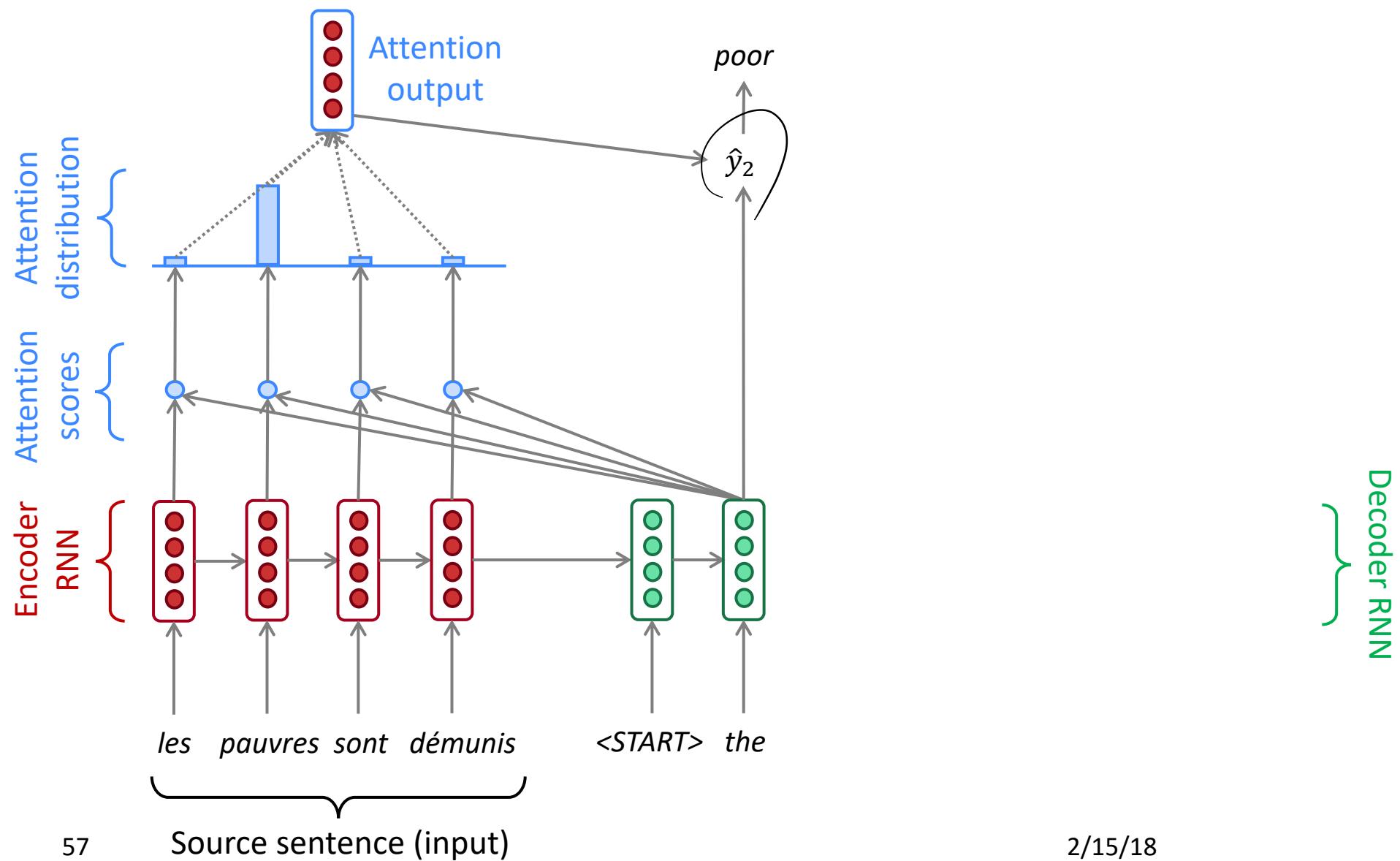
Sequence-to-sequence with attention



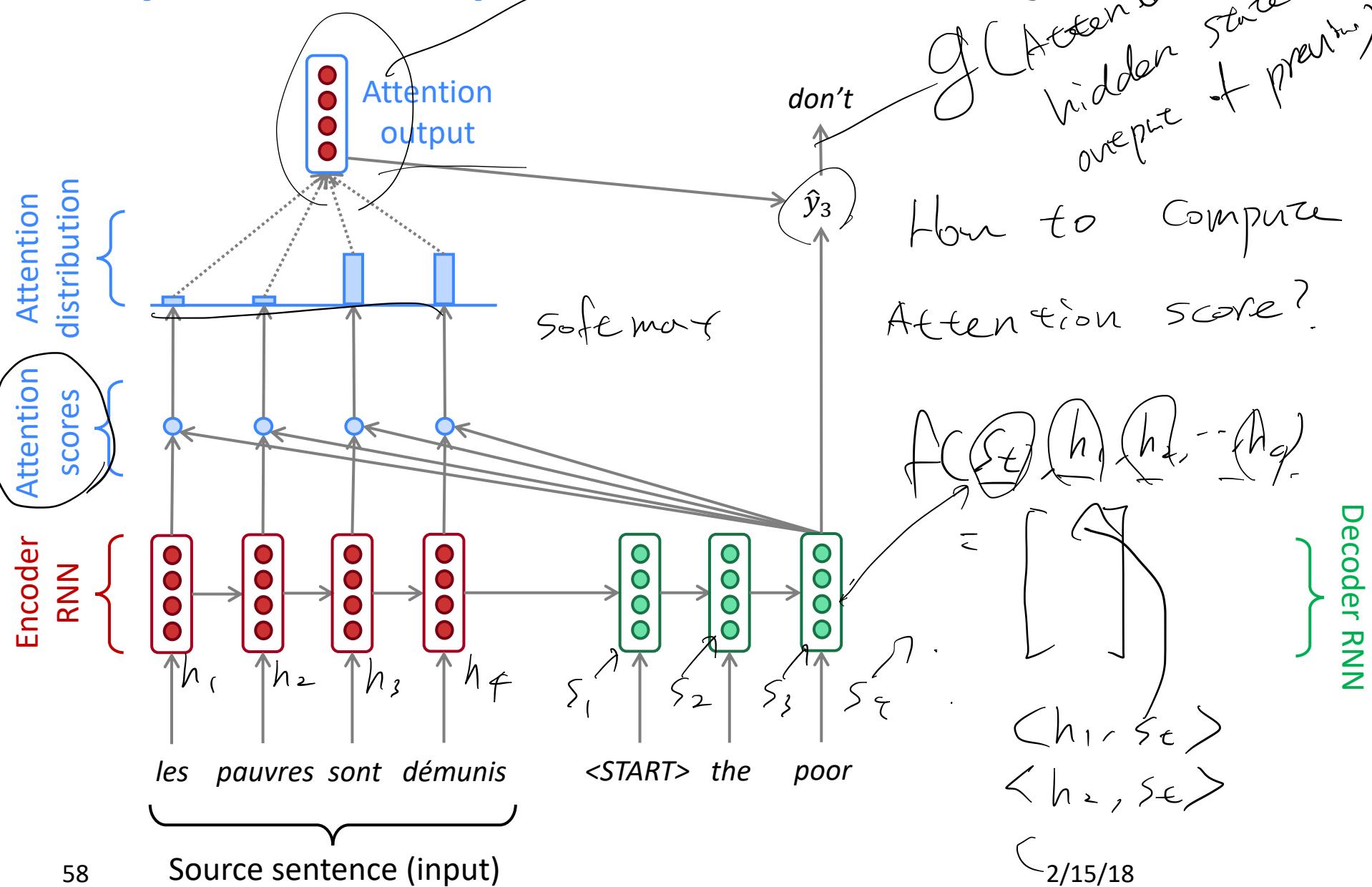
Sequence-to-sequence with attention



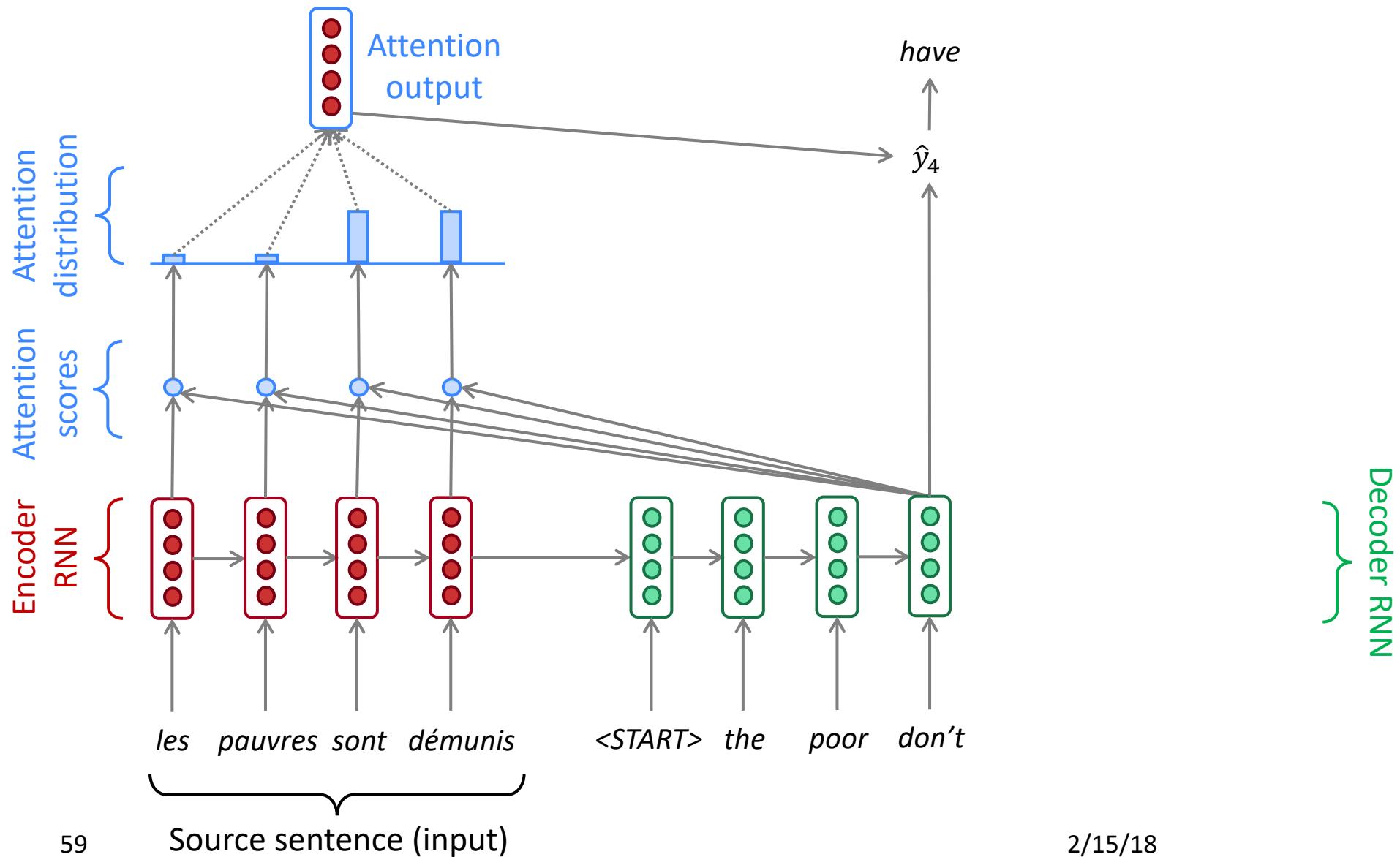
Sequence-to-sequence with attention



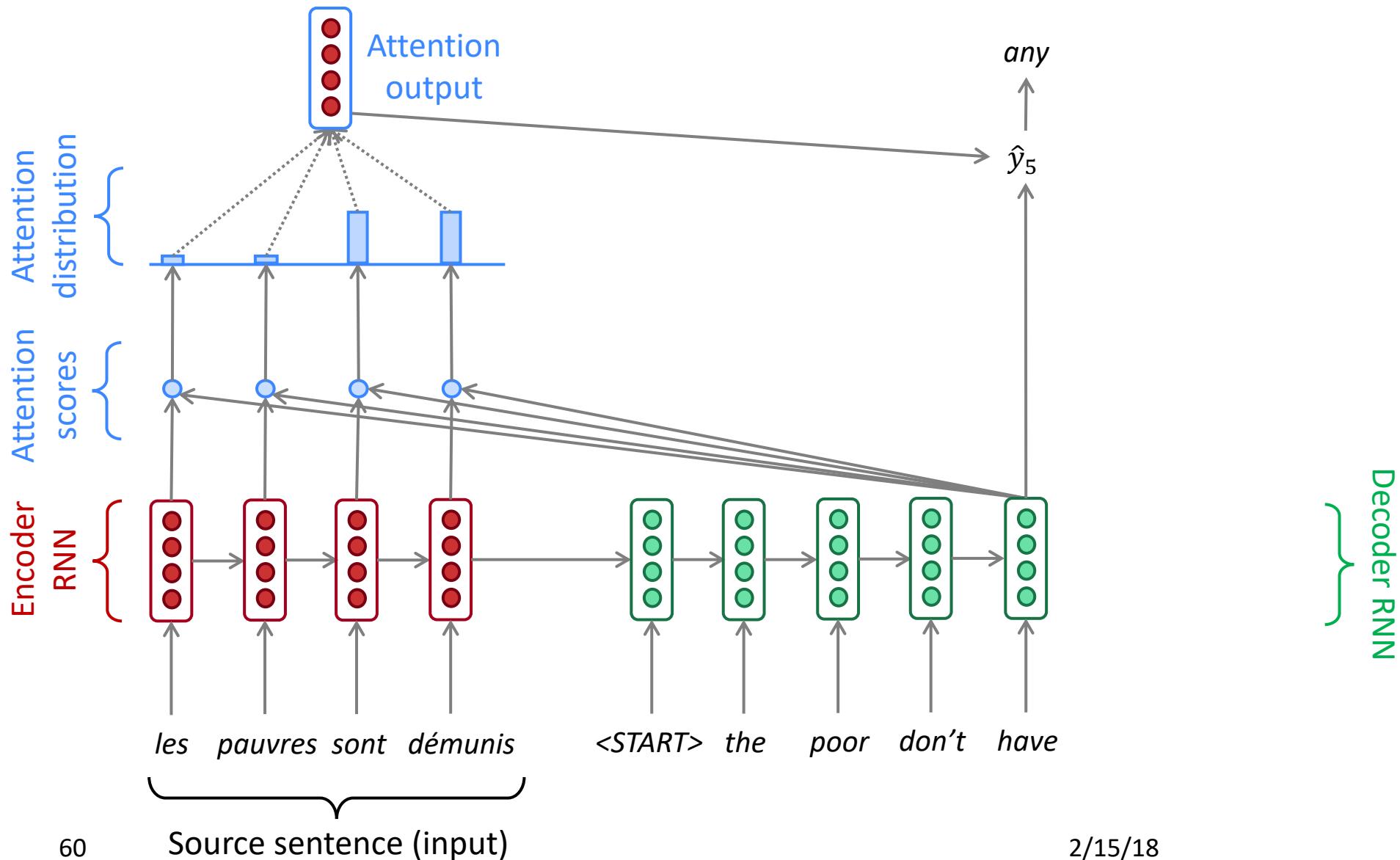
Sequence-to-sequence with attention



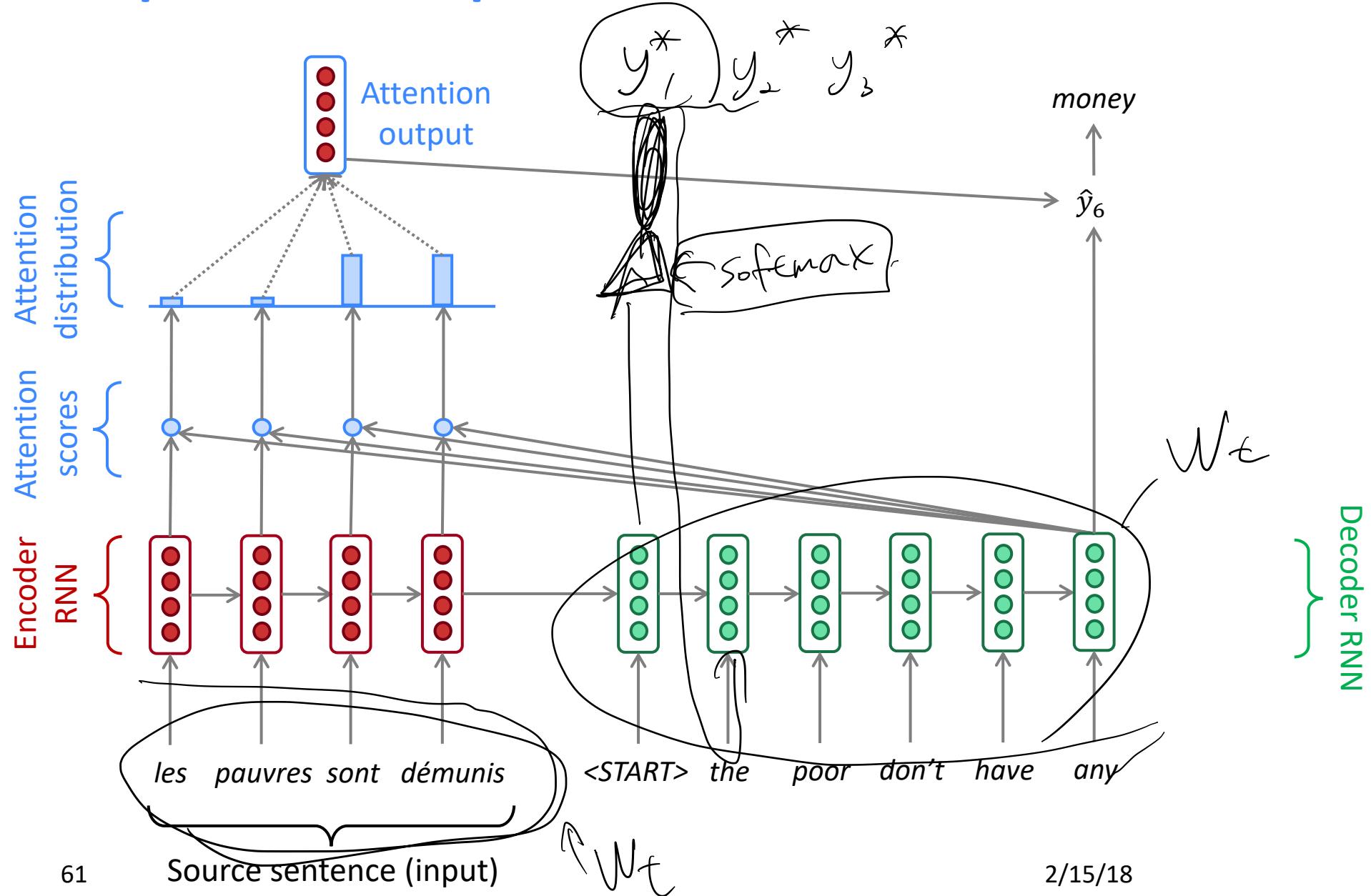
Sequence-to-sequence with attention



Sequence-to-sequence with attention



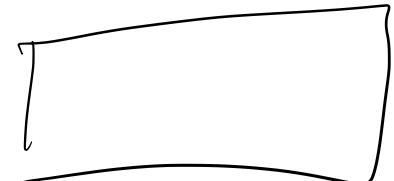
Sequence-to-sequence with attention



Attention: in equations

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

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



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

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

- 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

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

There are *several* attention variants

- We have some *values* $\underline{\mathbf{h}_1}, \dots, \underline{\mathbf{h}_N} \in \mathbb{R}^{d_1}$ and a *query* $\underline{s} \in \mathbb{R}^{d_2}$
- Attention always involves computing the *attention output* $\mathbf{a} \in \mathbb{R}^{d_1}$ (sometimes called the *context vector*) from the *attention scores* $\mathbf{e} \in \mathbb{R}^N$ (or *attention logits*) like so:

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

$$\mathbf{a} = \sum_{i=1}^N \alpha_i \mathbf{h}_i \in \mathbb{R}^{d_1} \quad (\text{take weighted sum})$$

- However, there are *several ways* you can compute $\mathbf{e} \in \mathbb{R}^N$

Attention variants

There are **several ways** you can compute $e \in \mathbb{R}^N$ from $\mathbf{h}_1, \dots, \mathbf{h}_N \in \mathbb{R}^{d_1}$ and $\mathbf{s} \in \mathbb{R}^{d_2}$:

- Basic dot-product attention: $e_i = \underbrace{\mathbf{s}^T \mathbf{h}_i}_{\in \mathbb{R}}$
 - Note: this assumes $d_1 = d_2$
 - This is the version we saw earlier
- Multiplicative attention: $e_i = \underbrace{\mathbf{s}^T \mathbf{W} \mathbf{h}_i}_{\in \mathbb{R}}$
 - Where $\mathbf{W} \in \mathbb{R}^{d_2 \times d_1}$ is a weight matrix
- Additive attention: $e_i = \mathbf{v}^T \tanh(\mathbf{W}_1 \mathbf{h}_i + \mathbf{W}_2 \mathbf{s}) \in \mathbb{R}$
 - Where $\mathbf{W}_1 \in \mathbb{R}^{d_3 \times d_1}$, $\mathbf{W}_2 \in \mathbb{R}^{d_3 \times d_2}$ are weight matrices and $\mathbf{v} \in \mathbb{R}^{d_3}$ is a weight vector

More information: <http://ruder.io/deep-learning-nlp-best-practices/index.html#attention>

The First Attention Paper

Published as a conference paper at ICLR 2015

NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE

Dzmitry Bahdanau

Jacobs University Bremen, Germany

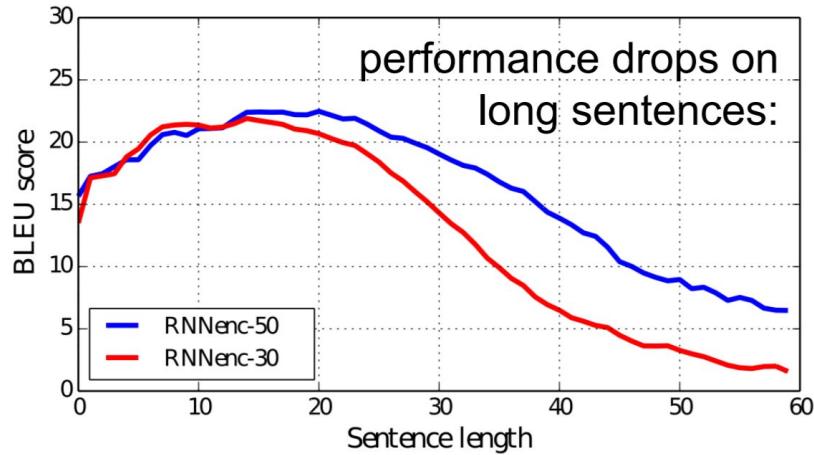
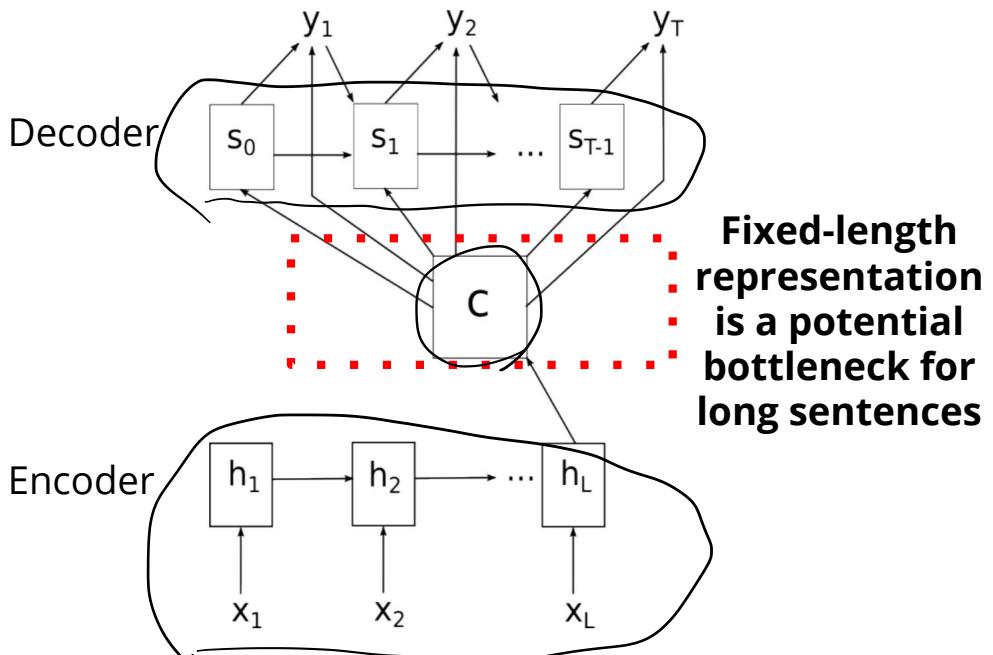
 **KyungHyun Cho**

 **Yoshua Bengio***

Université de Montréal

Motivation

Existing encoder-decoder frameworks



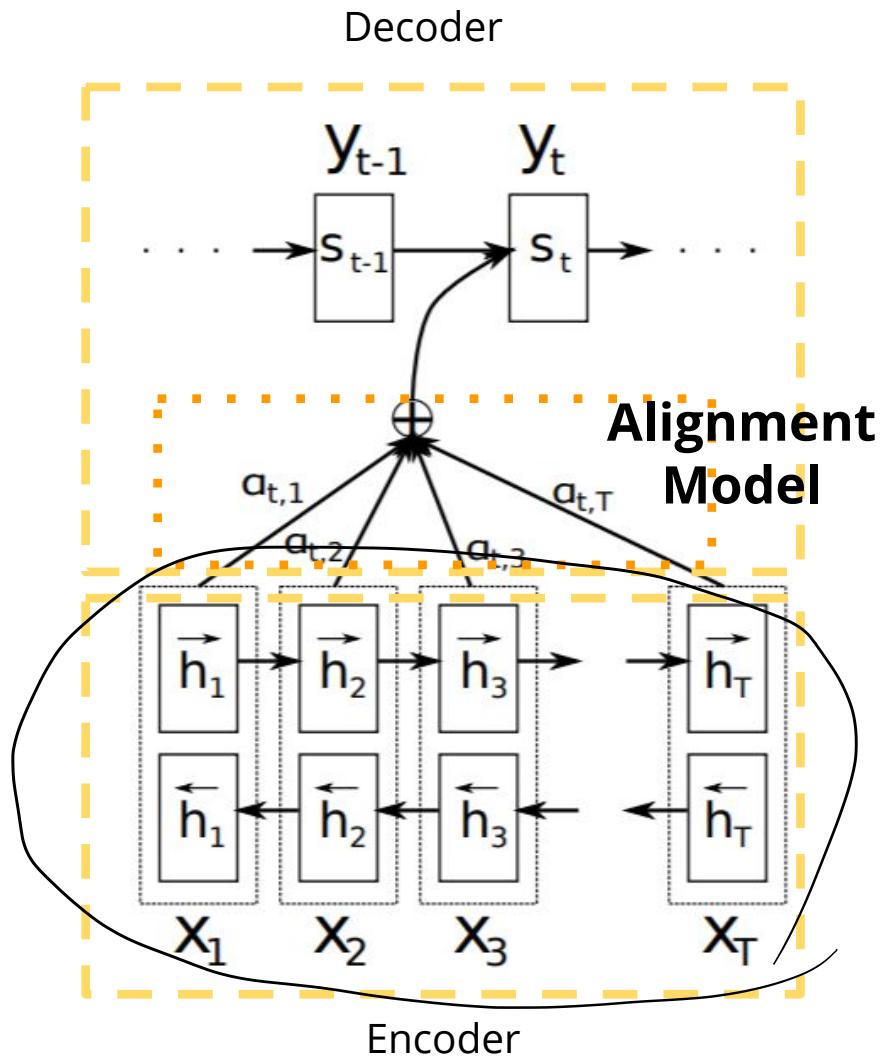
Evaluated on news-test-2014 from
WMT '14 English-French parallel corpora

Method

For each generated word y_t in the translation, soft-searches for a set of positions $(1, \dots, T)$ in a source sentence $x = (x_1, \dots, x_T)$ where the most relevant information is concentrated.

The predicted target word y_t is based on the context vectors $\mathbf{c}_t = \sum a_{t,j} \mathbf{h}_j$ associated with these source positions and all the previous generated target words s_{t-1}, y_{t-1} .

$$\mathbf{c}_t = \begin{bmatrix} \mathbf{h}_j \\ \vdots \\ \mathbf{h}_j \end{bmatrix}$$

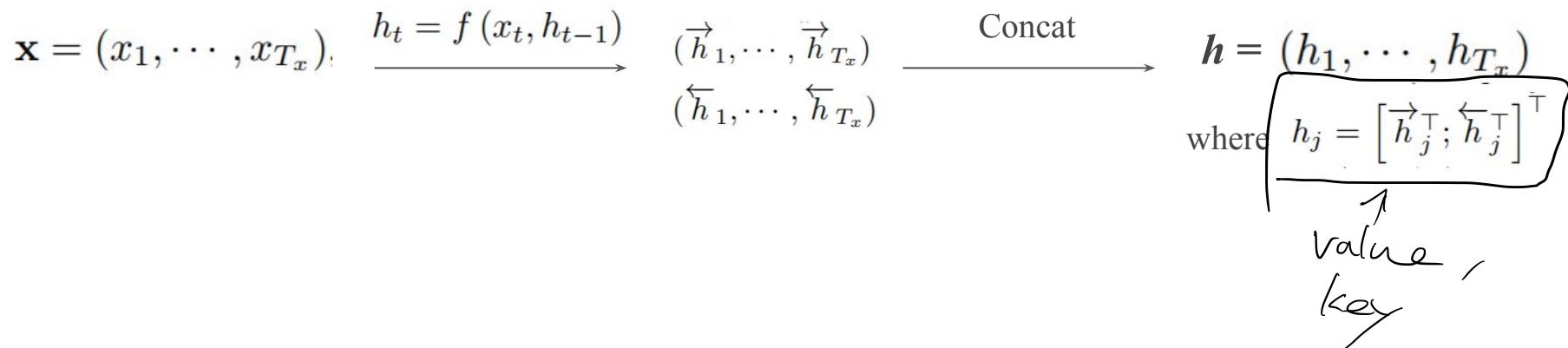
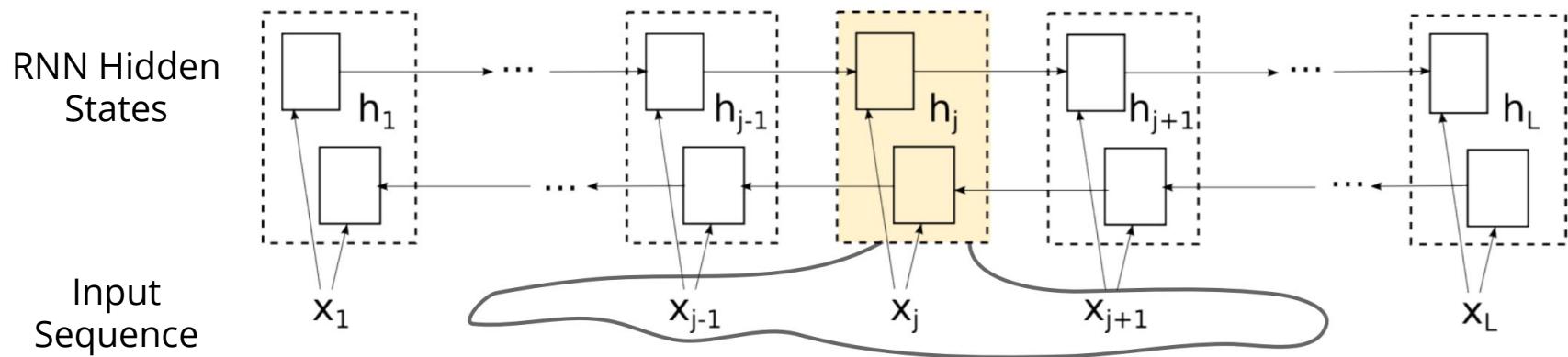


$$\overrightarrow{h}_t = \vec{f}(x_t, \overleftarrow{h}_{t-1})$$

$$\overleftarrow{h}_t = \overleftarrow{f}(x_t, \overrightarrow{h}_{t+1})$$

Encoder - Bidirectional RNN

- Map input sentence x to a sequence of annotations h
- Each annotation summarizes the preceding words and the following words



$$\underbrace{P(y_i | \{y_1, \dots, y_{i-1}\}, c_i, s_{i-1})}_{s_i = g(y_{i-1}, s_{i-1}, c_i)} = \overbrace{P(y_i | y_{i-1}, c_i, s_{i-1})}$$

Decoder - Alignment Model

- Compute alignment α_{ij}

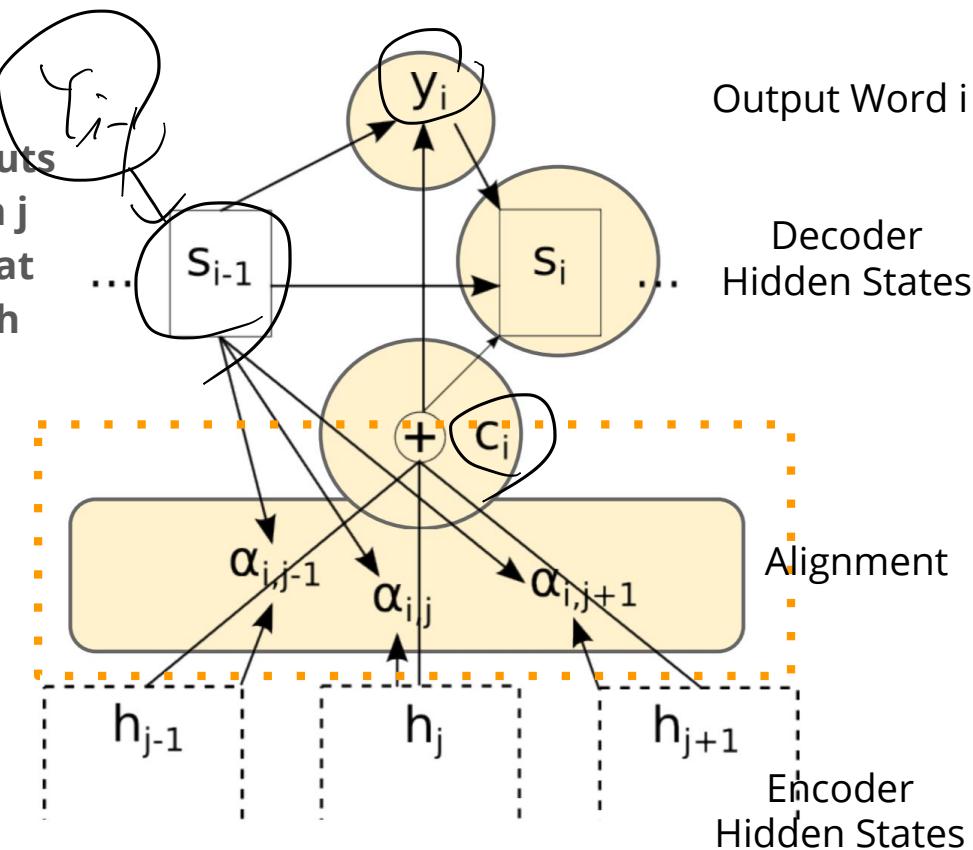
$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}$$

How well the inputs around position j and the output at position i match

Use simple feedforward NN to compute e_{ij} based on s_{i-1} and h_j

$$e_{ij} = v^T \tanh(Ws_{i-1} + Vh_j)$$

where v, W, V are trainable weights



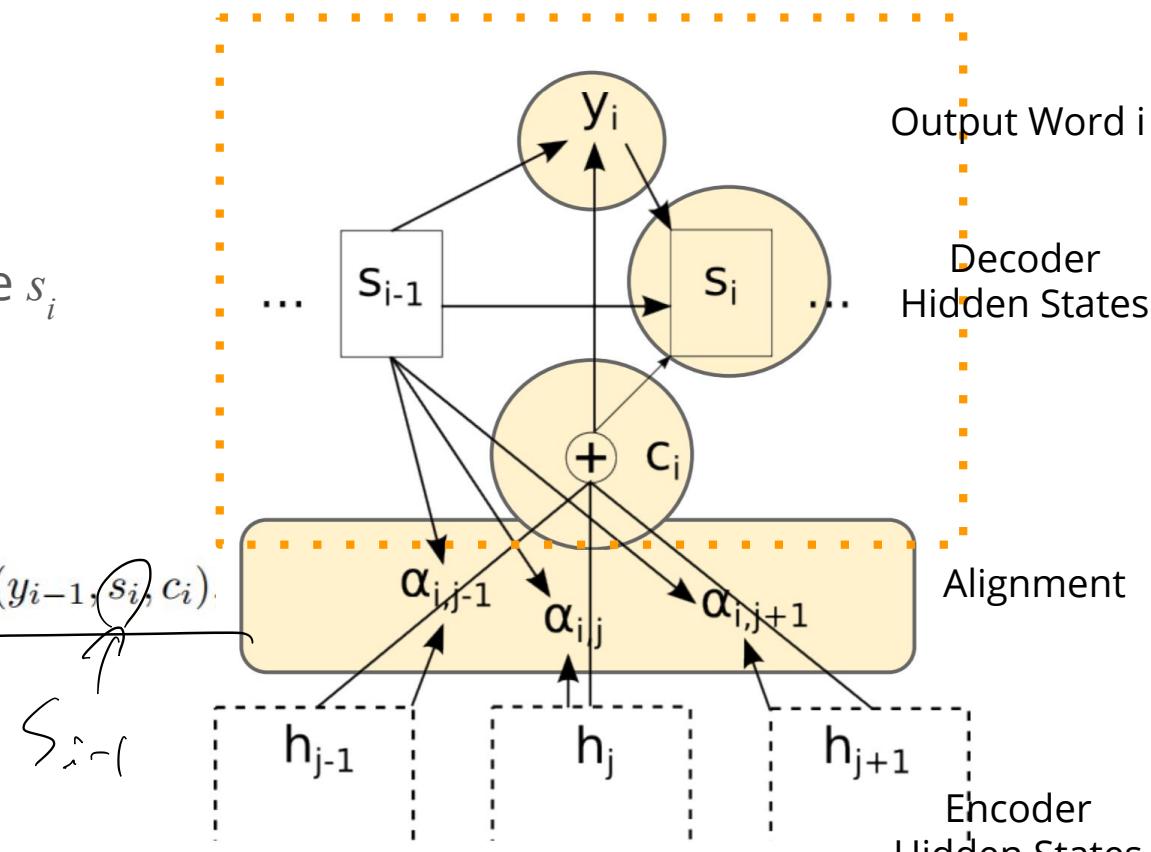
Decoder

- Compute context c_i
- Compute new decoder state s_i
- Generate new output y_i

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j$$

$$s_i = f(s_{i-1}, y_{i-1}, c_i)$$

$$\text{argmax } p(y_i | y_1, \dots, y_{i-1}, \mathbf{x}) = g(y_{i-1}, s_i, c_i)$$



$$g(y_{i-1}, s_i, c_i)$$

$$g(y_{i-1}, s_i, c_i)$$

Various score functions

Effective Approaches to Attention-based Neural Machine Translation

Minh-Thang Luong Hieu Pham Christopher D. Manning
Computer Science Department, Stanford University, Stanford, CA 94305
`{lmthang, hyhieu, manning}@stanford.edu`

$$\text{score}(\mathbf{h}_t, \bar{\mathbf{h}}_s) = \begin{cases} \mathbf{h}_t^\top \bar{\mathbf{h}}_s & \textit{dot} \\ \mathbf{h}_t^\top \mathbf{W}_a \bar{\mathbf{h}}_s & \textit{general} \\ \mathbf{v}_a^\top \tanh(\mathbf{W}_a[\mathbf{h}_t; \bar{\mathbf{h}}_s]) & \textit{concat} \end{cases}$$

CNN, RNN, Attention.

Attention Is All You Need

Ashish Vaswani*

Google Brain

avaswani@google.com

Noam Shazeer*

Google Brain

noam@google.com

Niki Parmar*

Google Research

nikip@google.com

Jakob Uszkoreit*

Google Research

usz@google.com

Llion Jones*

Google Research

llion@google.com

Aidan N. Gomez* †

University of Toronto

aidan@cs.toronto.edu

Lukasz Kaiser*

Google Brain

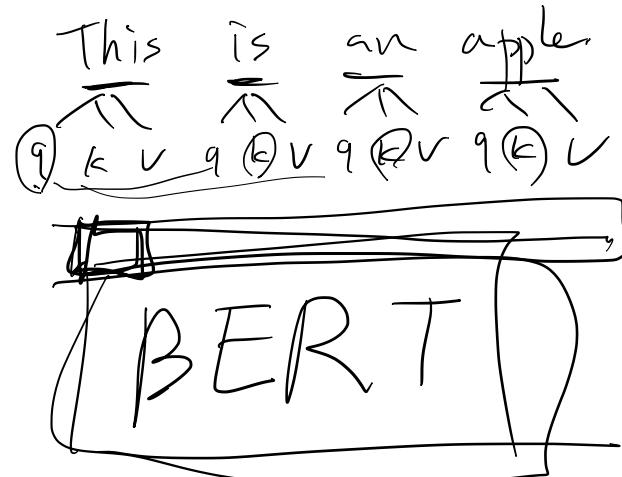
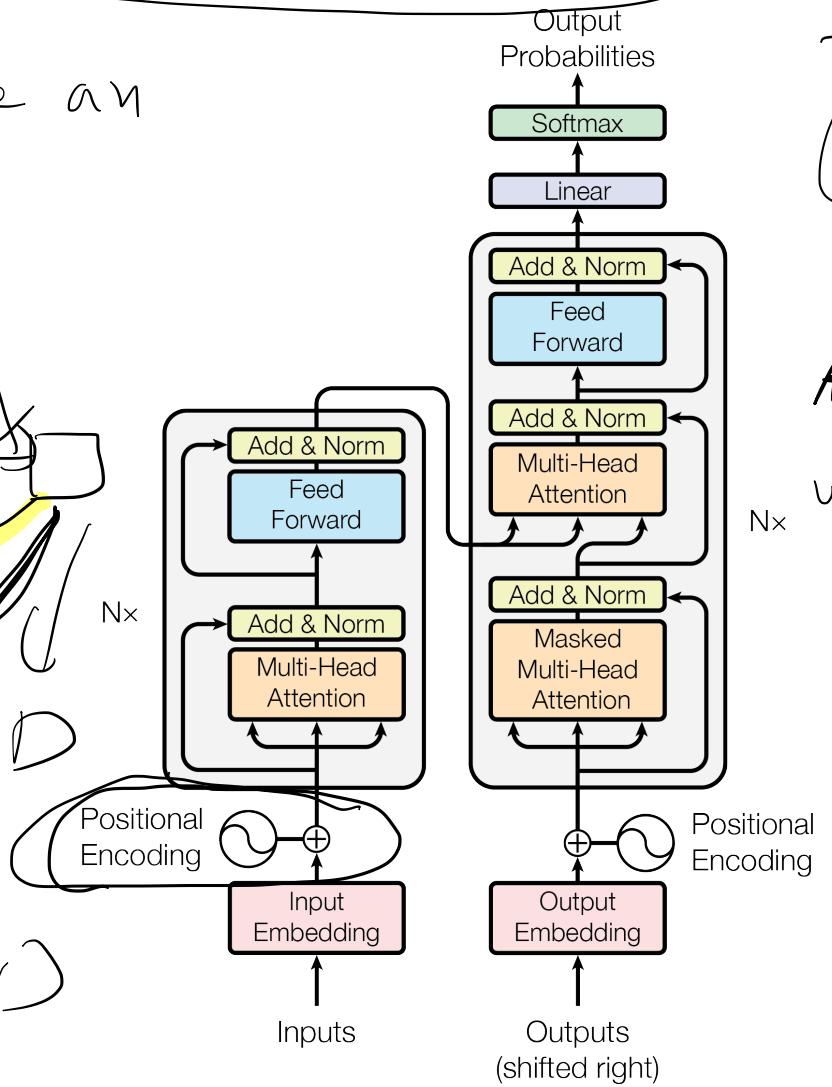
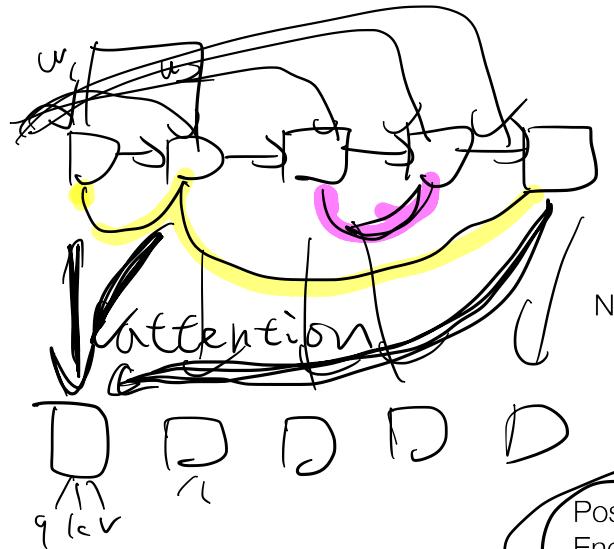
lukaszkaiser@google.com

Illia Polosukhin* ‡

illia.polosukhin@gmail.com

Transformer Network

Is this apple an



$A(q, k, v)$

word f query
key
value

$$= \frac{e^{q \cdot k_i}}{\sum_j e^{q \cdot k_j}}$$