

Natural Language Processing with Deep Learning

CS224N/Ling284



Lecture 10:
Machine Translation,
Sequence-to-sequence and Attention

Abigail See

Announcements

- Honor code issues: Assignment 2
- Assignment 3 released
- Azure credits released
- Default final project update:
 - New handout released
 - Submission instructions released
- Custom final project: you should receive feedback on your proposal this week
- Midterm grades: released after lecture

Happy Valentines Day!



RNN-generated candy hearts

Source: <http://aiweirdness.com/post/170820844947/more-candy-hearts-by-neural-network>

Welcome to the second half of the course!

- Remaining lectures are mostly **geared towards projects** *LOL*
- We'll bring you to the **cutting-edge** of NLP+DL research
- Lectures will be more **high-level**
 - **No more gradient computations!** *LOL*
 - Sometimes we'll sketch an overview – if you're interested in a topic, you can read more after class
- **However:** today's lecture will cover two **core** NLP Deep Learning techniques

Overview

Today we will:

- Introduce a new task: Machine Translation



is the primary use-case of

- Introduce a new neural architecture: sequence-to-sequence



seq2seq

is improved by

- Introduce a new neural technique: attention

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*

1950s: Early Machine Translation

Machine Translation research began in the **early 1950s**.

- Mostly Russian → English
(motivated by the Cold War!)



Source: <https://youtu.be/K-HfpsHPmvw>

- Systems were mostly **rule-based**, using a bilingual dictionary to map Russian words to their English counterparts
 - A cool by-product: Quicksort!

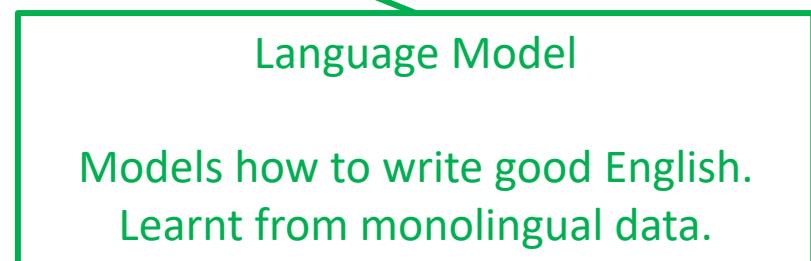
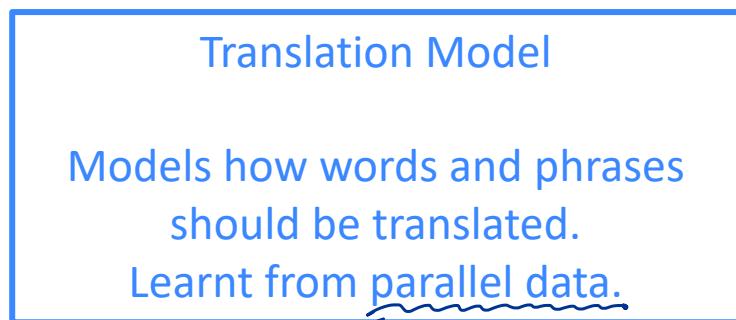
1990s-2010s: Statistical Machine Translation

- Core idea: Learn a **probabilistic model** from **data**
- Suppose we're translating French → English.
- We want to find **best English sentence y , given French sentence x**

$$\operatorname{argmax}_y P(y|x)$$

- Use Bayes Rule to break this down into **two components** to be learnt separately:

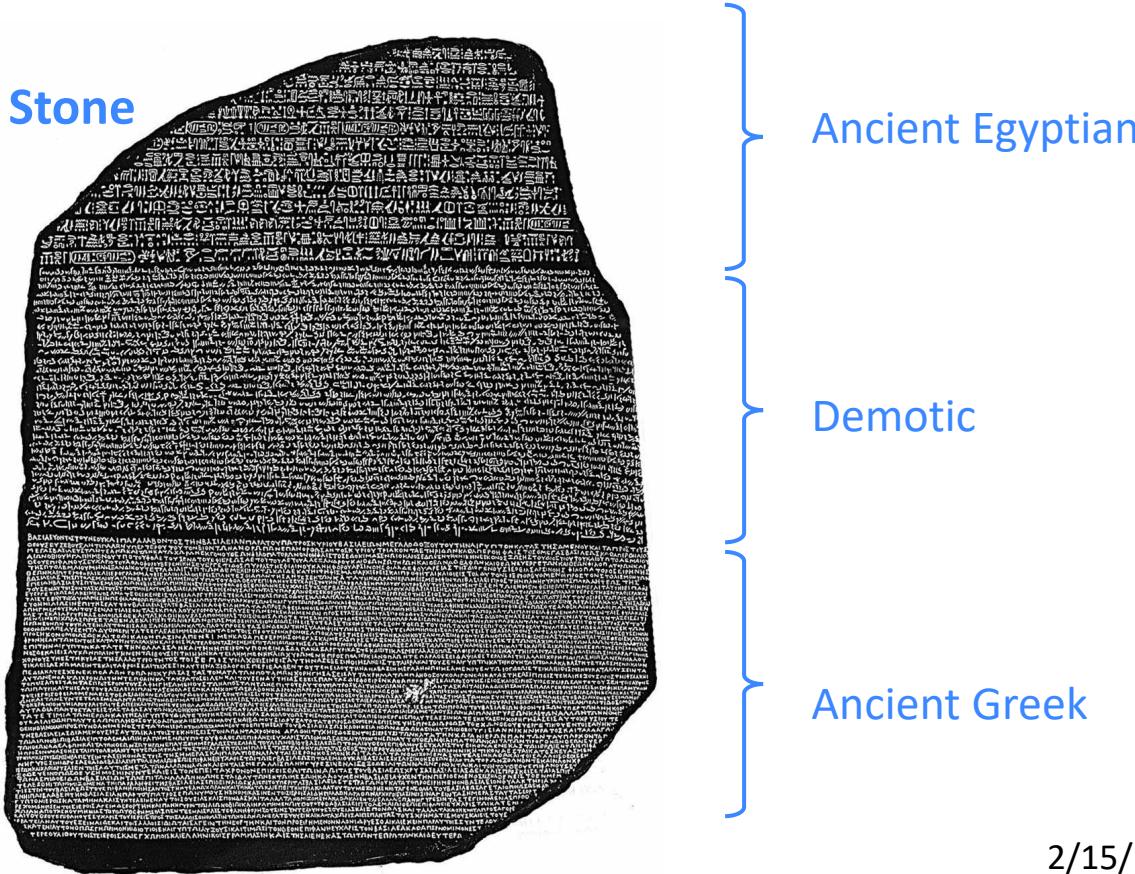
$$= \operatorname{argmax}_y P(x|y)P(y)$$



1990s-2010s: Statistical Machine Translation

- Question: How to learn translation model $P(x|y)$?
- First, need large amount of **parallel data**
(e.g. pairs of human-translated French/English sentences)

The Rosetta Stone



1990s-2010s: Statistical Machine Translation

- Question: How to learn translation model $P(x|y)$?
- First, need large amount of **parallel data**
(e.g. pairs of human-translated French/English sentences)
- Break it down further: we actually want to consider

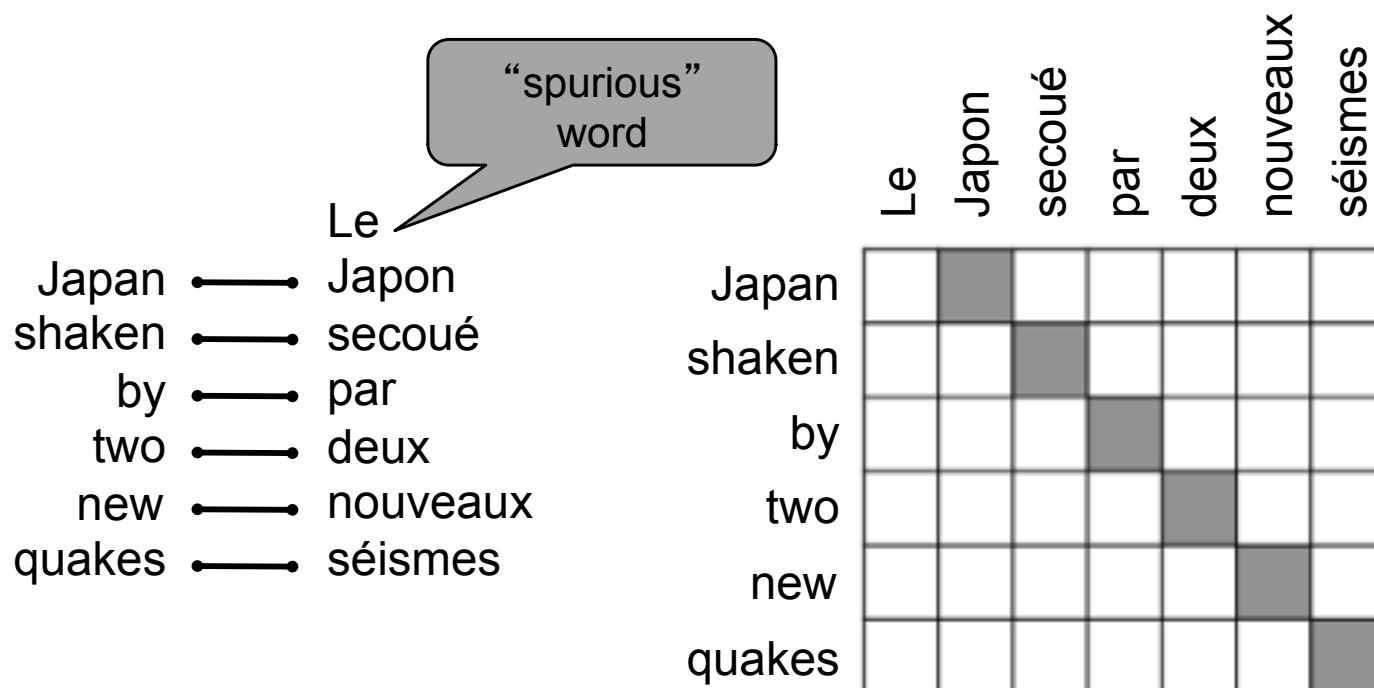
$$P(x, a|y)$$

where a is the **alignment**, i.e. word-level correspondence between French sentence x and English sentence y

What is alignment?

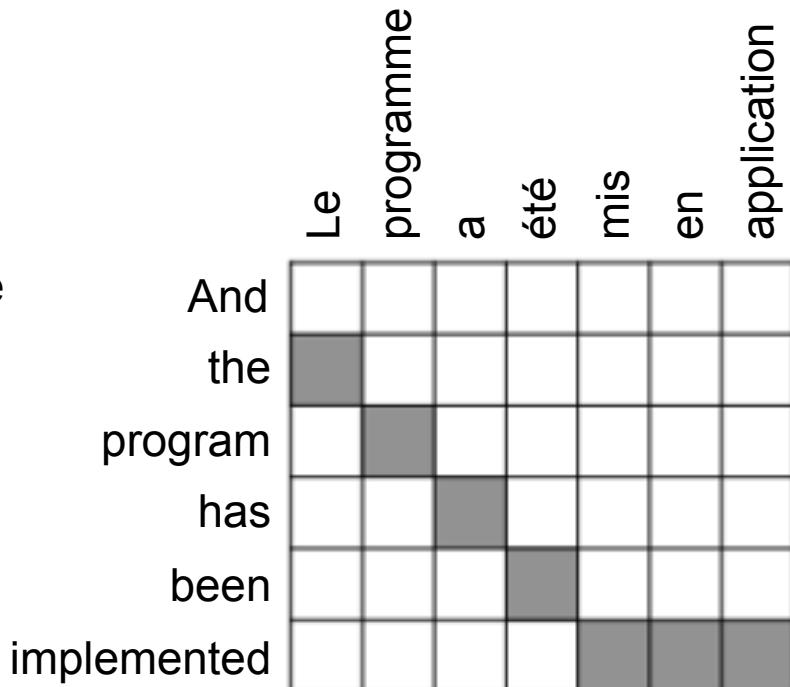
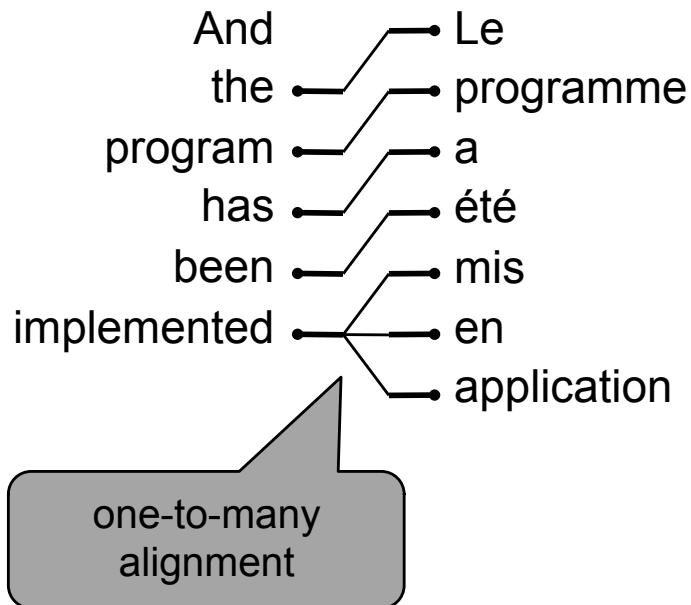
Alignment is the correspondence between particular words in the translated sentence pair.

- Note: Some words have no counterpart



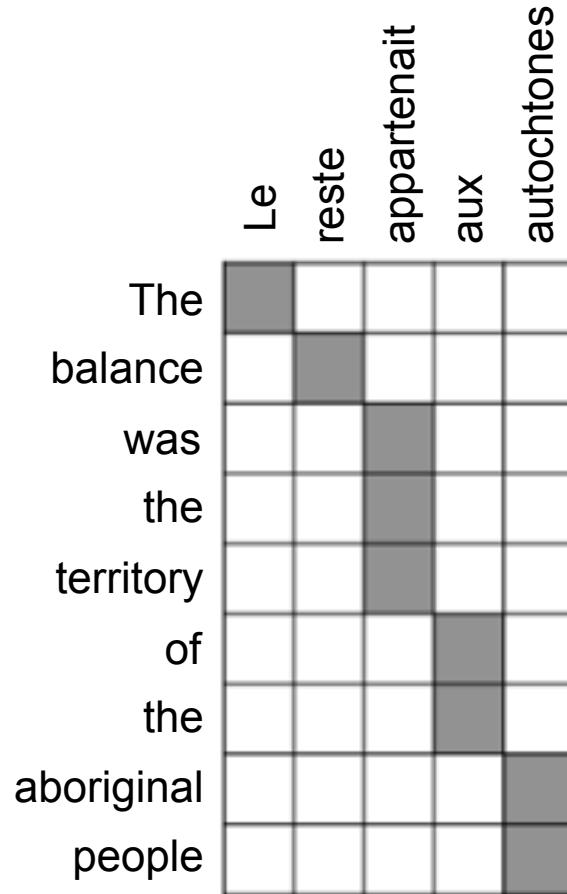
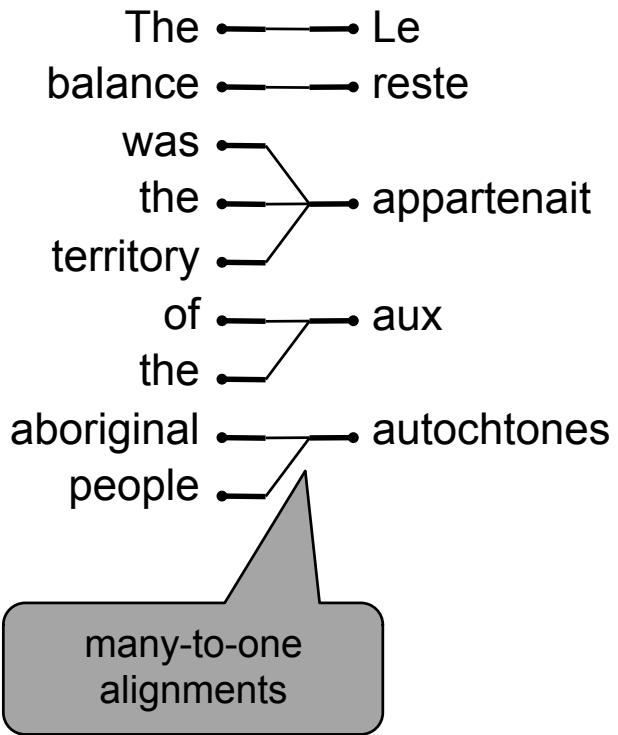
Alignment is complex

Alignment can be one-to-many (these are “fertile” words)



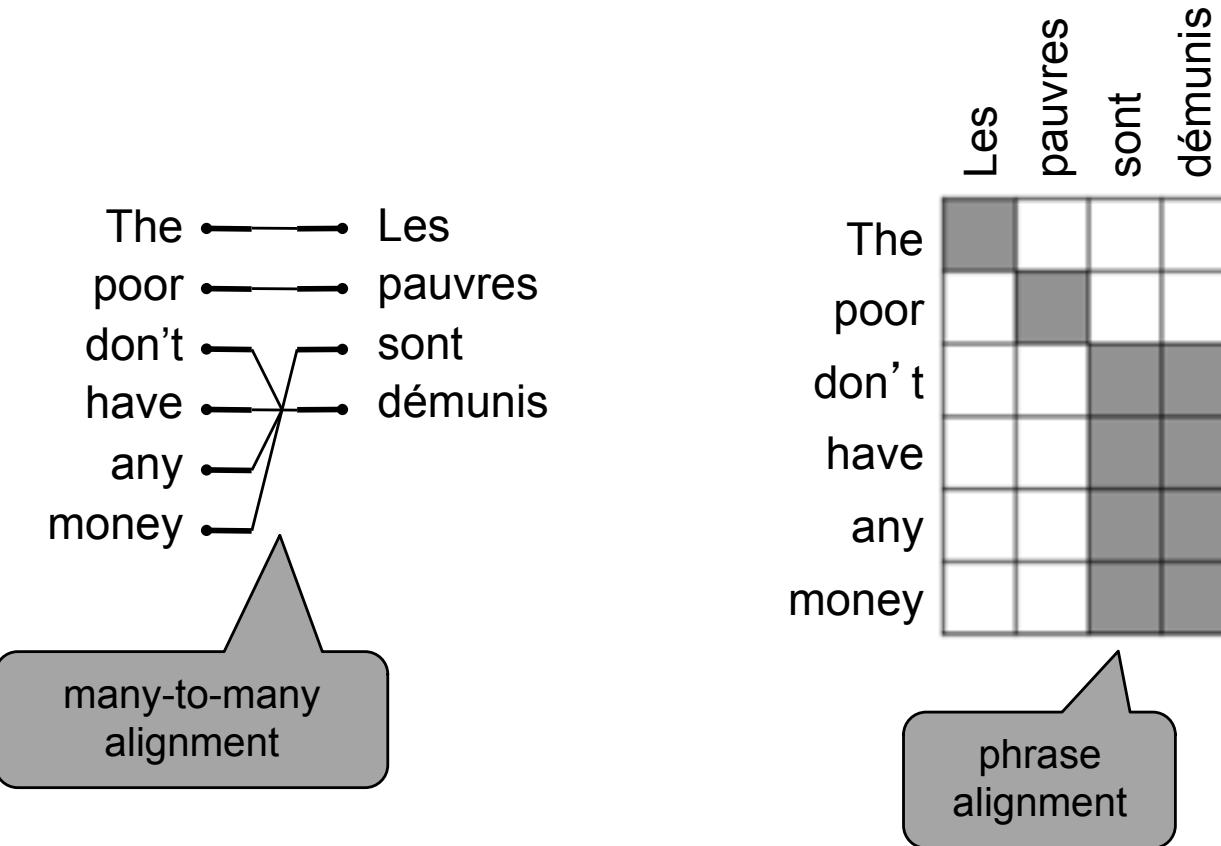
Alignment is complex

Alignment can be many-to-one



Alignment is complex

Alignment can be many-to-many (phrase-level)



1990s-2010s: Statistical Machine Translation

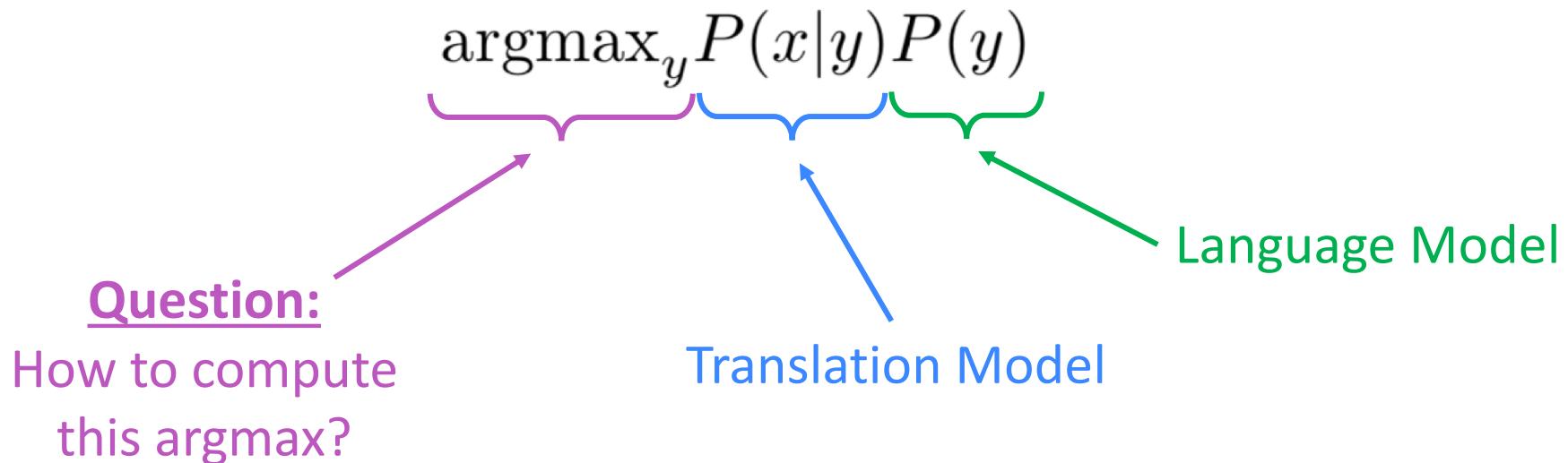
- Question: How to learn translation model $P(x|y)$?
- First, need large amount of **parallel data**
(e.g. pairs of human-translated French/English sentences)
- Break it down further: we actually want to consider

$$P(x, a|y)$$

where a is the **alignment**, i.e. word-level correspondence between French sentence x and English sentence y

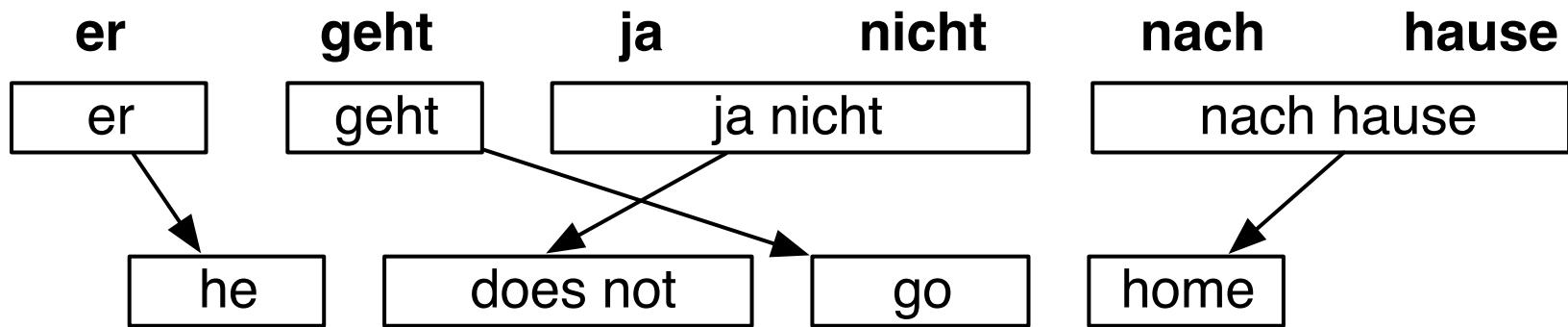
- We learn $P(x, a|y)$ as a combination of many factors, including:
 - Probability of particular words aligning
 - Also depends on position in sentence
 - Probability of particular words having particular fertility
 - Etc.

1990s-2010s: Statistical Machine Translation

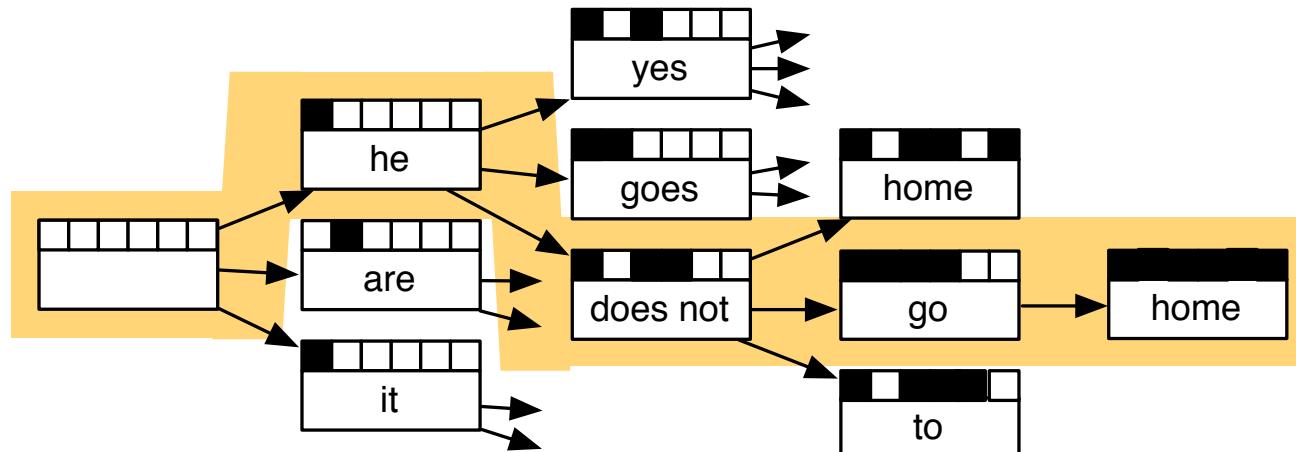
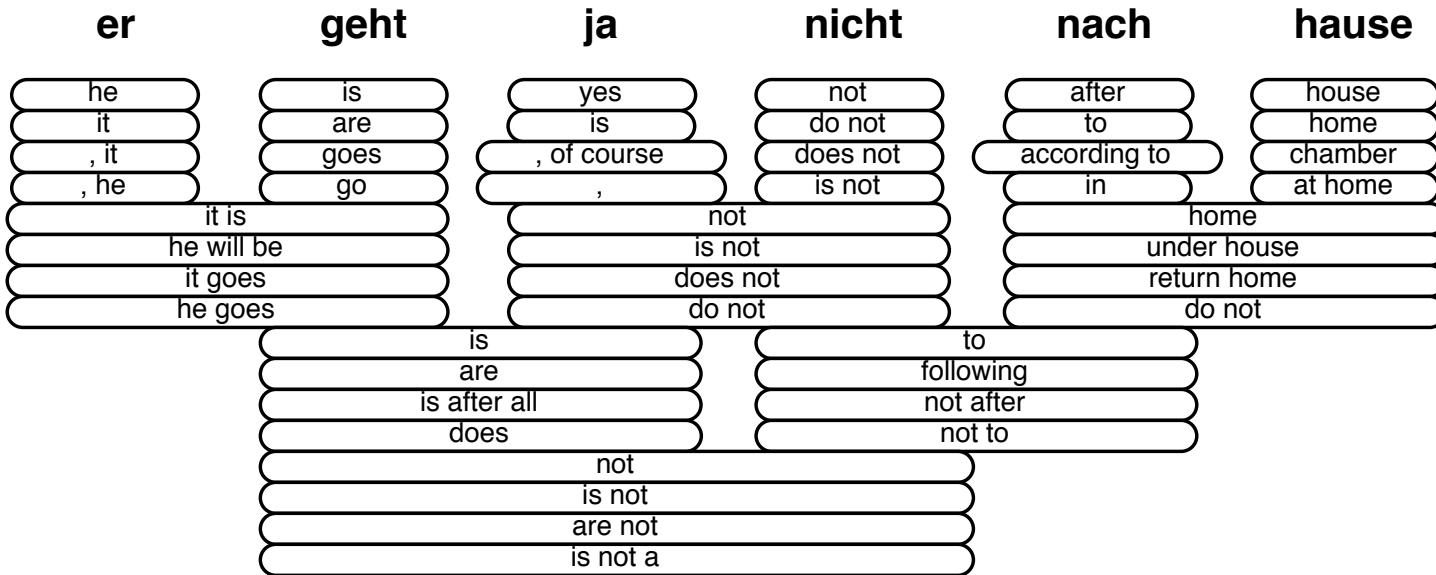


- We could enumerate every possible y and calculate the probability? → Too expensive!
- **Answer:** Use a heuristic search algorithm to gradually build up the the translation, discarding hypotheses that are too low-probability

Searching for the best translation



Searching for the best translation



1990s-2010s: Statistical Machine Translation

- SMT is a **huge research field**
- The best systems are **extremely complex**
 - Hundreds of important details we haven't mentioned here
 - Systems have many **separately-designed subcomponents**
 - Lots of **feature engineering**
 - Need to design features to capture particular language phenomena
 - Require compiling and maintaining **extra resources**
 - Like tables of equivalent phrases
 - Lots of **human effort** to maintain
 - Repeated effort for each language pair!

2014

(dramatic reenactment)

2014

Neural
Machine
Translation

MT research

(dramatic reenactment)

What is Neural Machine Translation?

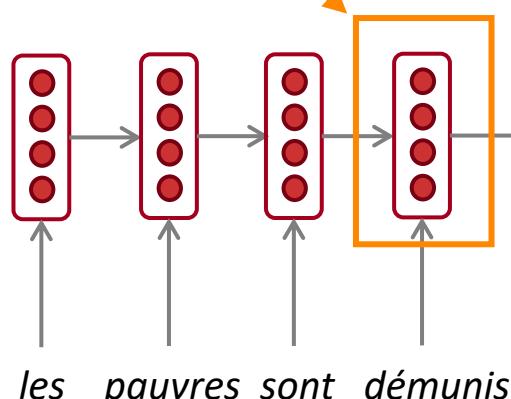
- Neural Machine Translation (NMT) is a way to do Machine Translation with a *single neural network*
- The neural network architecture is called sequence-to-sequence (aka seq2seq) and it involves *two RNNs*.

Neural Machine Translation (NMT)

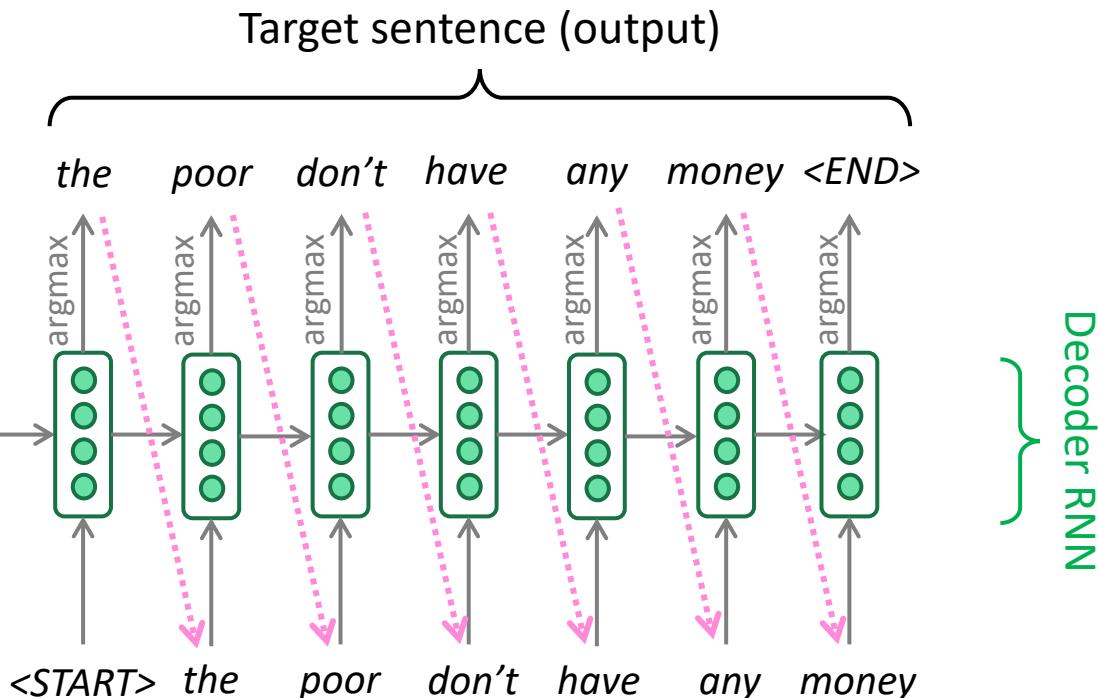
The sequence-to-sequence model

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 $P(y|x)$:

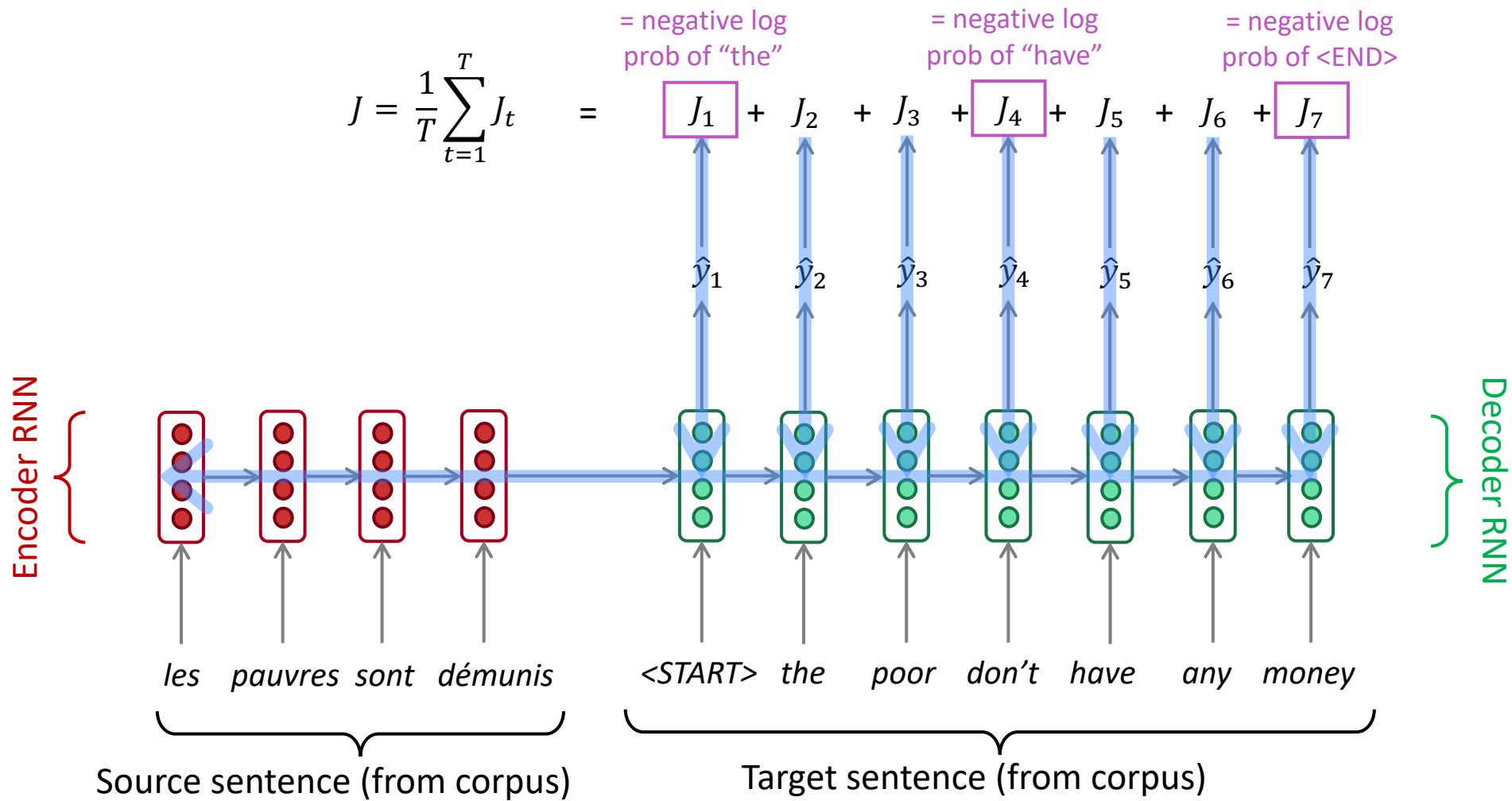
$$P(y|x) = P(y_1|x) P(y_2|y_1, x) P(y_3|y_1, y_2, x) \dots, P(y_T|y_1, \dots, y_{T-1}, x)$$



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...

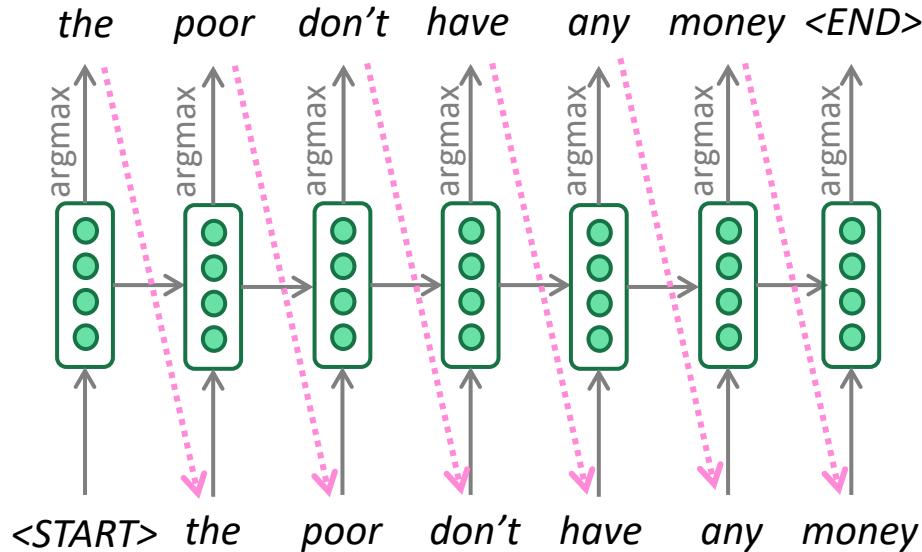
Training a Neural Machine Translation system



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

2 份搜索

Beam search decoding

$$\tilde{y} = (y_1, y_2 \dots y_T)$$

- 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!

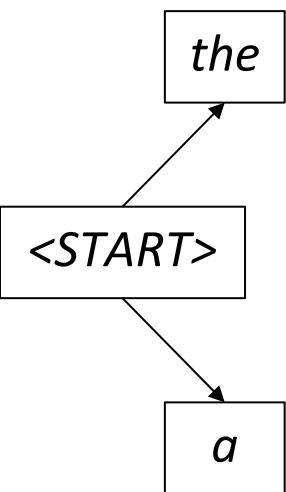
Beam search decoding: example

Beam size = 2

<START>

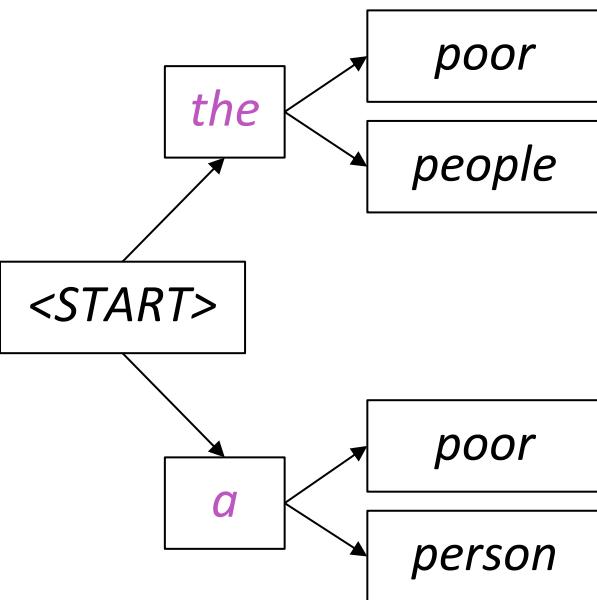
Beam search decoding: example

Beam size = 2



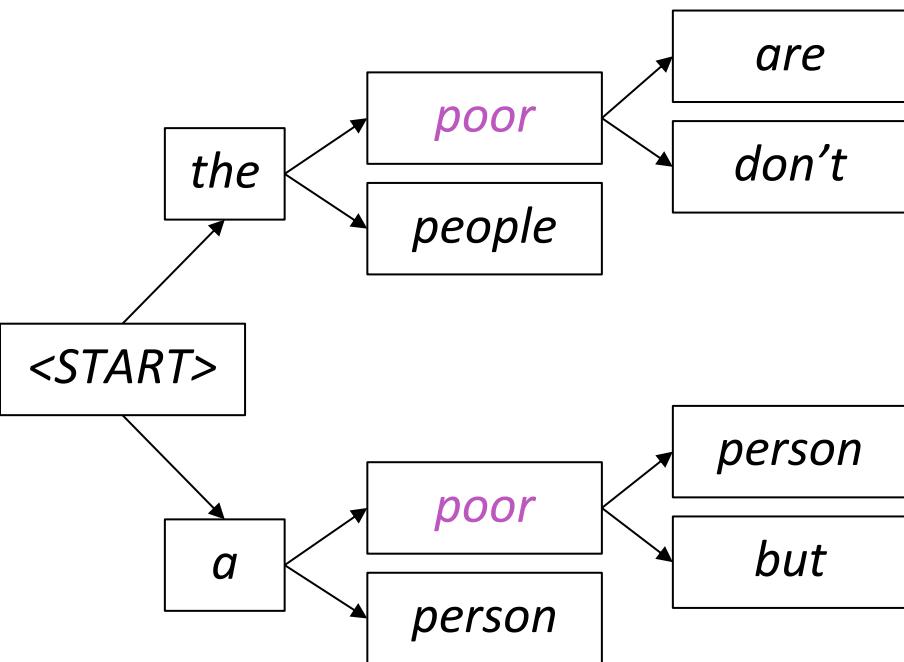
Beam search decoding: example

Beam size = 2



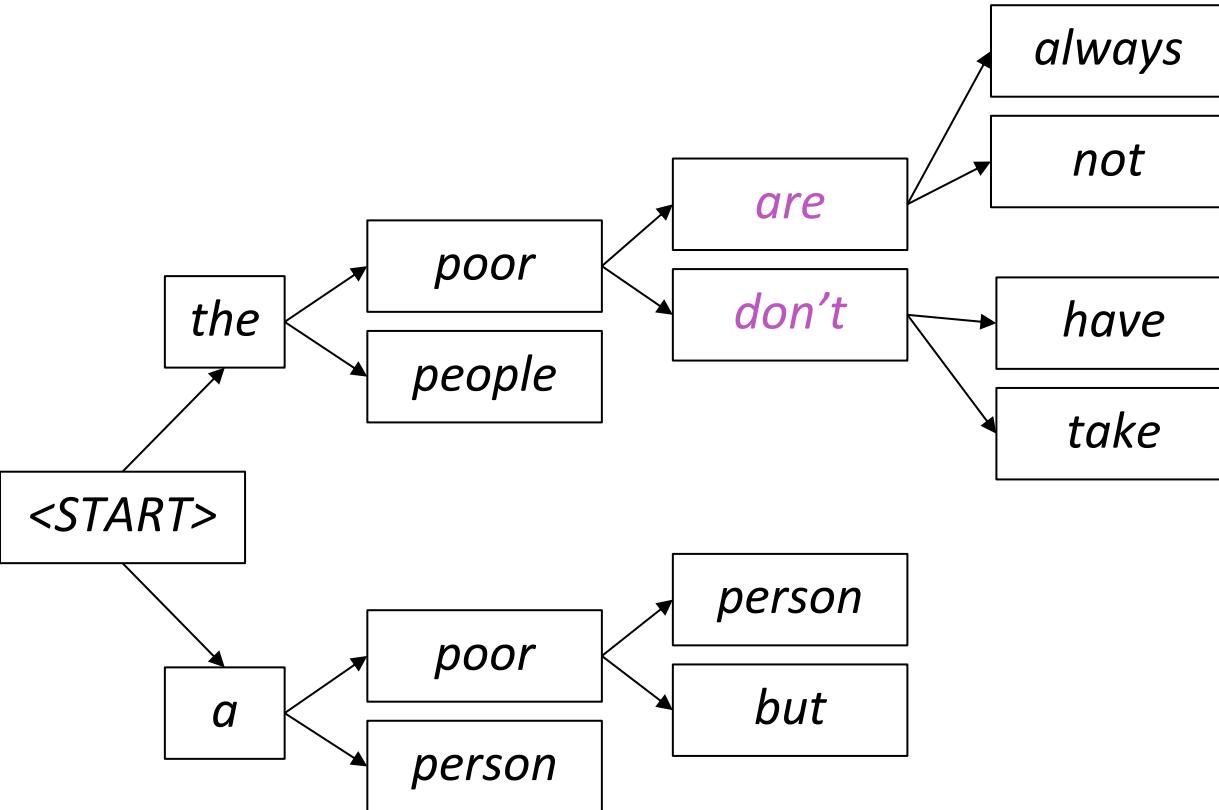
Beam search decoding: example

Beam size = 2



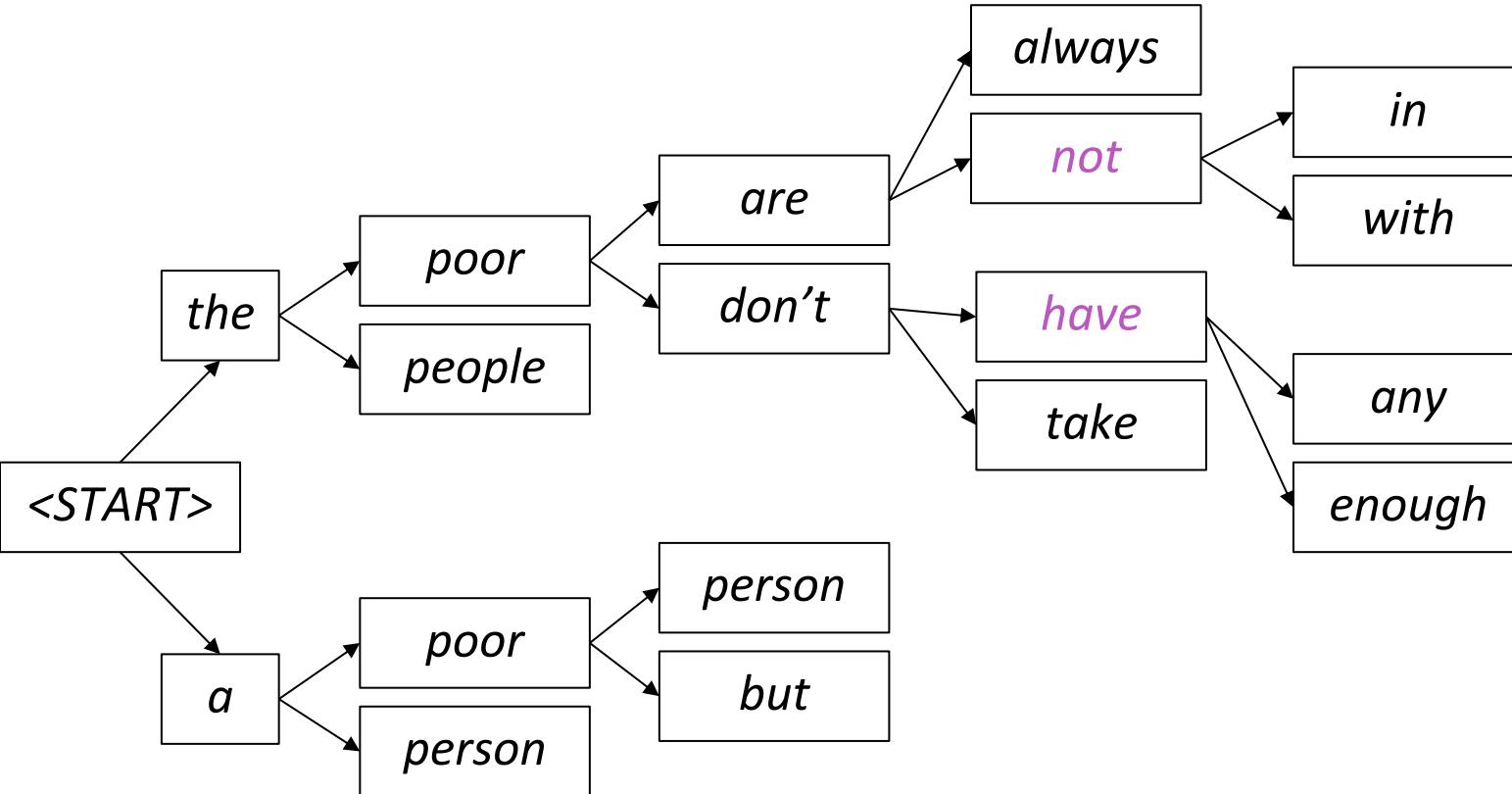
Beam search decoding: example

Beam size = 2



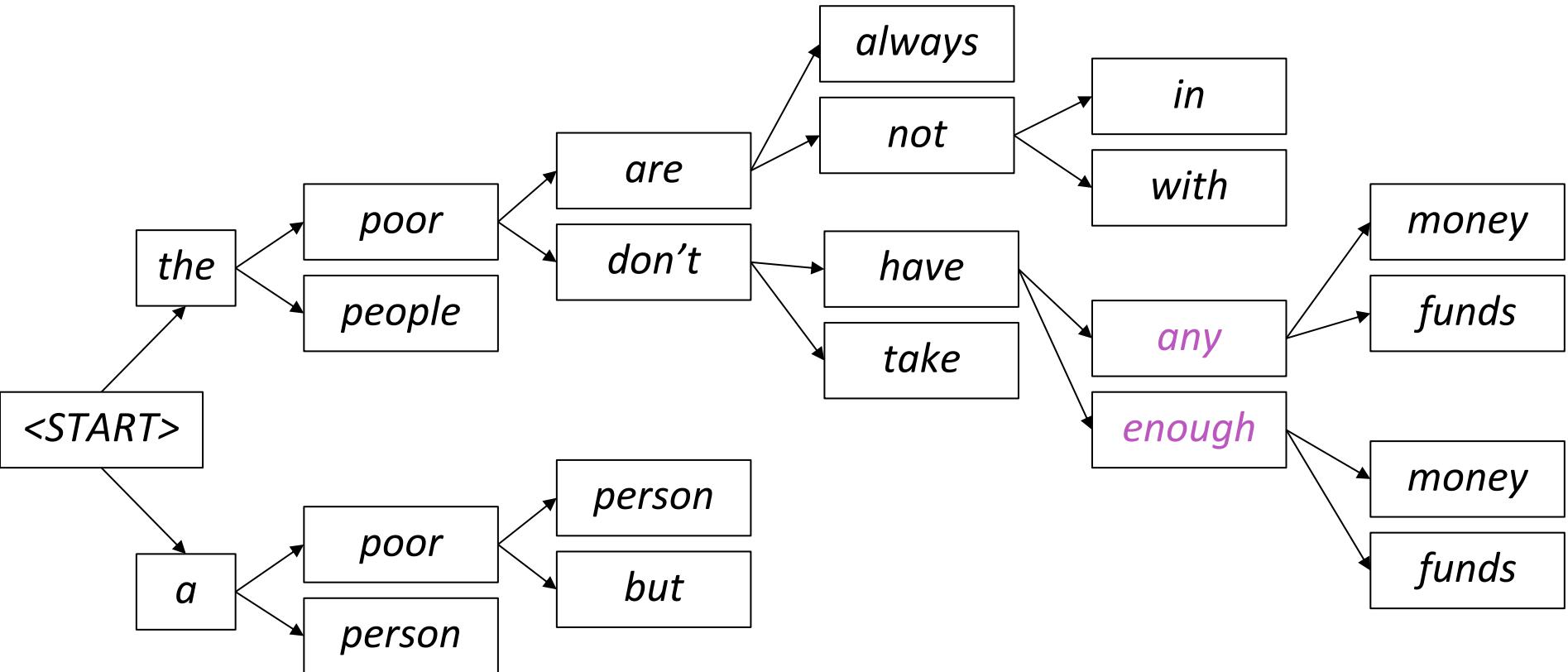
Beam search decoding: example

Beam size = 2



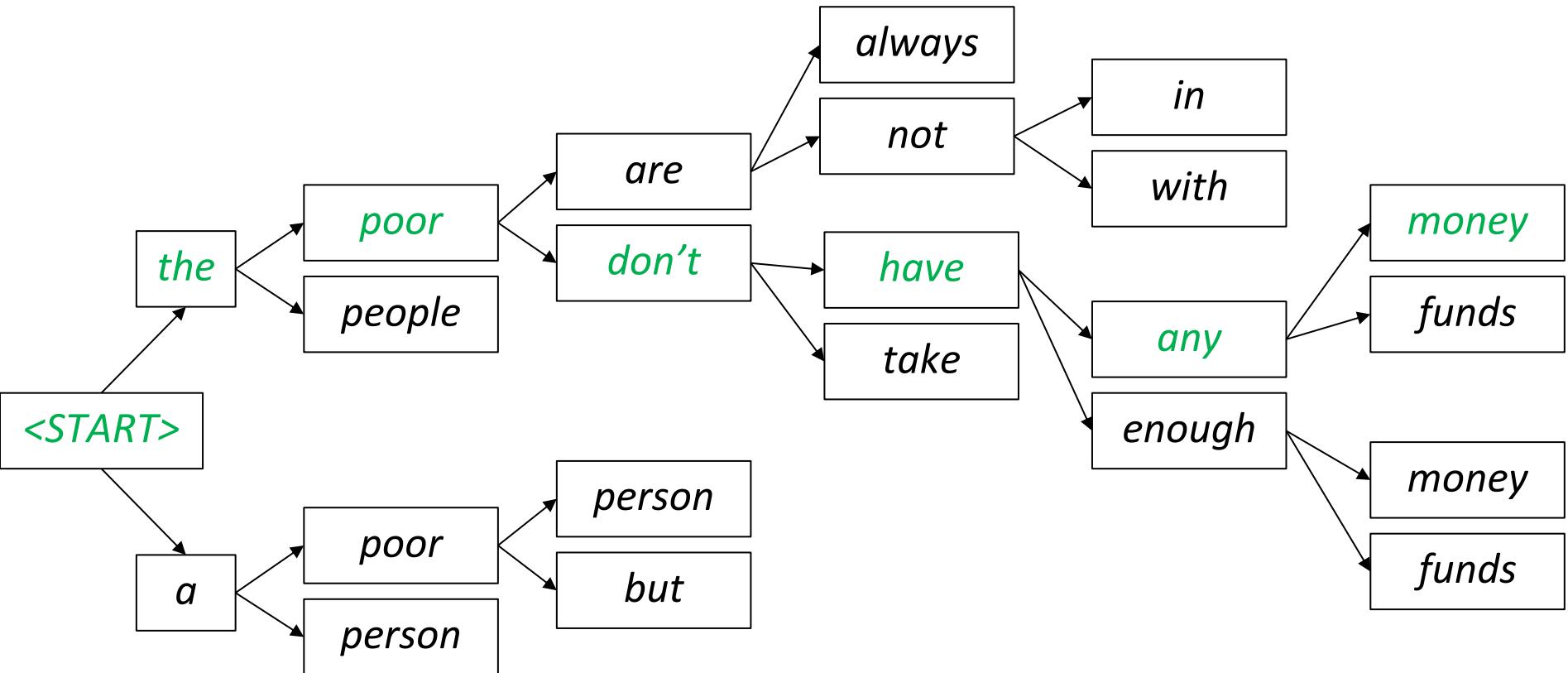
Beam search decoding: example

Beam size = 2



Beam search decoding: example

Beam size = 2



Advantages of NMT

Statistical Machine Translation, Neural Machine Translation

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!

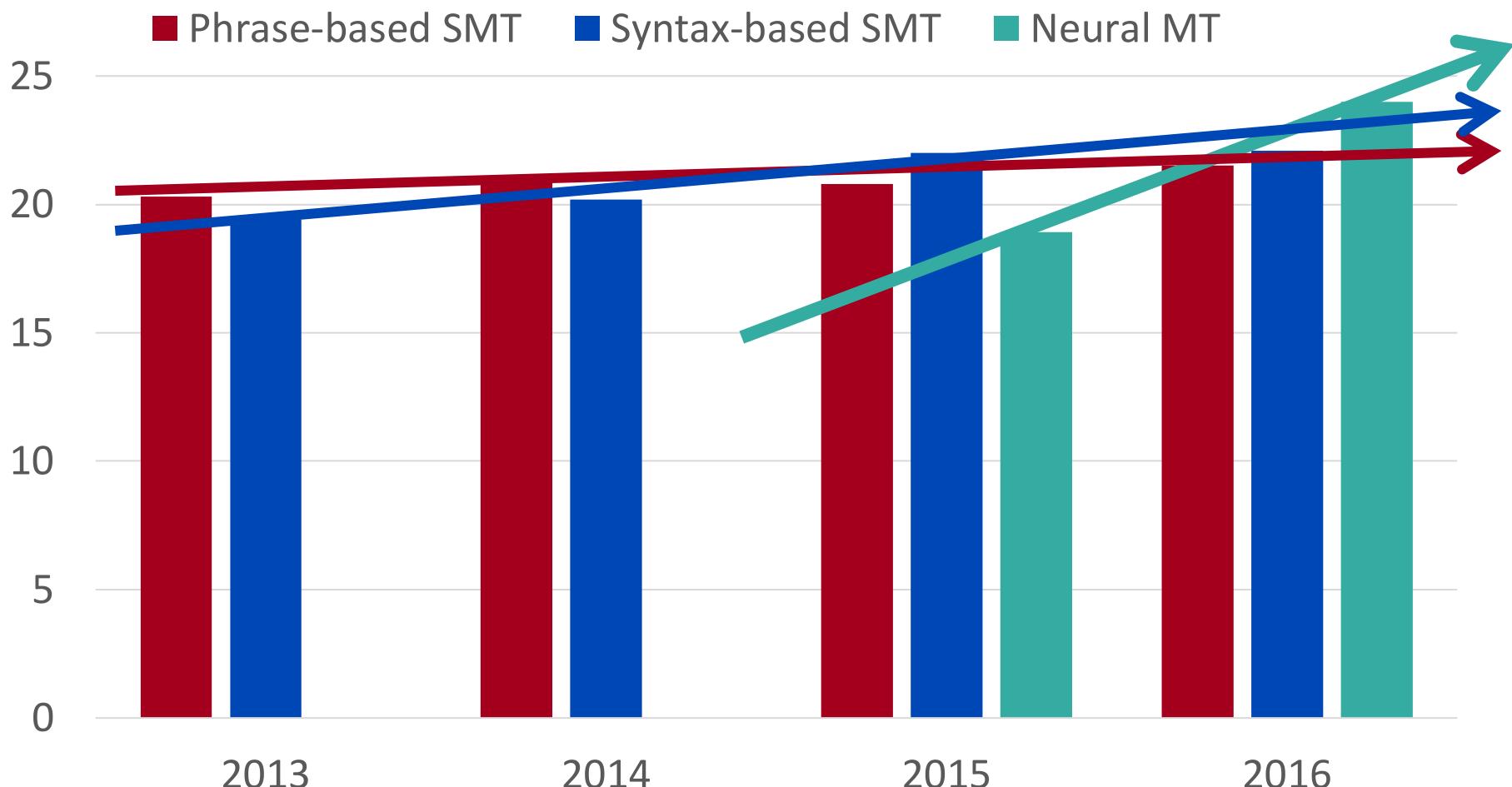
How do we evaluate Machine Translation?

BLEU (Bilingual Evaluation Understudy)

- BLEU compares the machine-written translation to one or several human-written translation(s), and computes a **similarity score** based on:
 - *n*-gram precision (usually up to 3 or 4-grams)
 - Penalty for too-short system translations
- BLEU is **useful** but **imperfect**
 - There are many valid ways to translate a sentence
 - So a **good** translation can get a **poor** BLEU score because it has low *n*-gram overlap with the human translation ☹

MT progress over time

[Edinburgh En-De WMT newstest2013 Cased BLEU; NMT 2015 from U. Montréal]



Source: http://www.meta-net.eu/events/meta-forum-2016/slides/09_sennrich.pdf

NMT: the biggest success story of NLP Deep Learning

Neural Machine Translation went from a **fringe research activity** in **2014** to the **leading standard method** in **2016**

- **2014:** First seq2seq paper published
- **2016:** Google Translate switches from SMT to NMT
- This is amazing!
 - SMT systems, built by **hundreds** of engineers over many **years**, outperformed by NMT systems trained by a **handful** of engineers in a few **months**

So is Machine Translation solved?

- **Nope!**
- Many difficulties remain:
 - Out-of-vocabulary words
 - Domain mismatch between train and test data
 - Maintaining context over longer text
 - Low-resource language pairs

So is Machine Translation solved?

- **Nope!**
- Using common sense is still hard

The image shows a screenshot of the Google Translate interface. On the left, under "English", the text "paper jam" is displayed with an "Edit" link. On the right, under "Spanish", the translation "Mermelada de papel" is shown. The interface includes language selection dropdowns, microphone and speaker icons, and a copy button. Below the main window, there are links "Open in Google Translate" and "Feedback".



?

So is Machine Translation solved?

- **Nope!**
- NMT picks up **biases** in training data

The screenshot shows a machine translation interface with Malay input and English output. The Malay input "Dia bekerja sebagai jururawat." is translated to "She works as a nurse." The Malay input "Dia bekerja sebagai pengaturcara." is translated to "He works as a programmer." A pink arrow points from the word "Dia" in the first sentence to the gendered outputs, highlighting the lack of gender specification in the original input.

Malay - detected ▾

English ▾

Dia bekerja sebagai jururawat.

Dia bekerja sebagai pengaturcara. Edit

She works as a nurse.

He works as a programmer.

Didn't specify gender

Source: <https://hackernoon.com/bias-sexist-or-this-is-the-way-it-should-be-ce1f7c8c683c>

So is Machine Translation solved?

- Nope!
- Uninterpretable systems do strange things

The image shows a machine translation interface with two panels. The left panel has language selection buttons for English, Spanish, Japanese, Detect language, and a dropdown menu. The right panel has buttons for English, Spanish, Arabic, and a blue 'Translate' button. Below these are two lists of text. The first list contains many occurrences of the Japanese character 'が' (ga). The second list shows the English translations corresponding to each 'が': 'But', 'Peel', 'A pain is', 'I feel a strange feeling', 'My stomach', 'Strange feeling', 'Strange feeling', 'Having a bad appearance', 'My bad gray', 'Strong but burns', 'Strong but burns', 'There was a bad shape but a bad shape', 'It is prone to burns, but also a burn', and 'Strong but burnished'. At the bottom of the right panel are five small icons: a star, a square, a triangle, a double arrow, and a left arrow.

Japanese	English Translation
が	But
ががが	Peel
がががが	A pain is
ががががが	I feel a strange feeling
がががががが	My stomach
ががががががが	Strange feeling
がががががががが	Strange feeling
ががががががががが	Having a bad appearance
ががががががががが	My bad gray
ががががががががが	Strong but burns
ががががががががが	Strong but burns
がががががががががが	There was a bad shape but a bad shape
ががががががががが	It is prone to burns, but also a burn
ががががががががが	Strong but burnished

Source: <http://languagelog.ldc.upenn.edu/nll/?p=35120#more-35120>

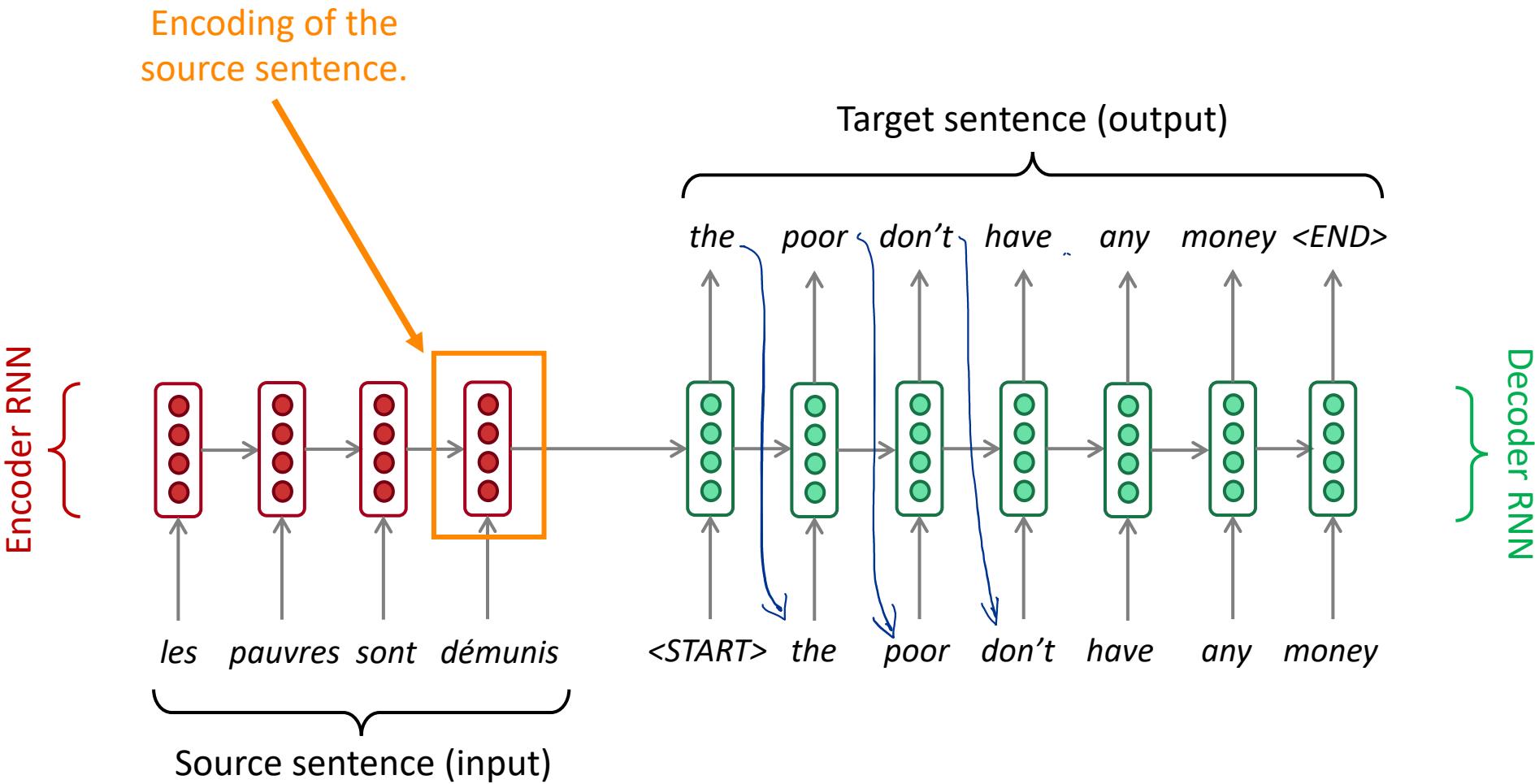
NMT research continues

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



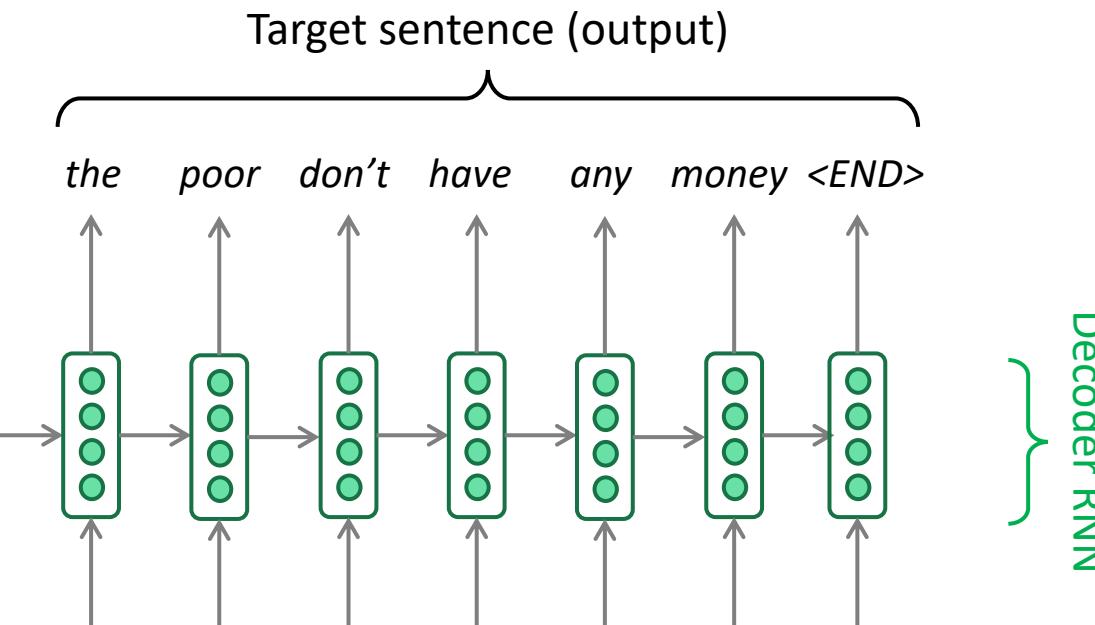
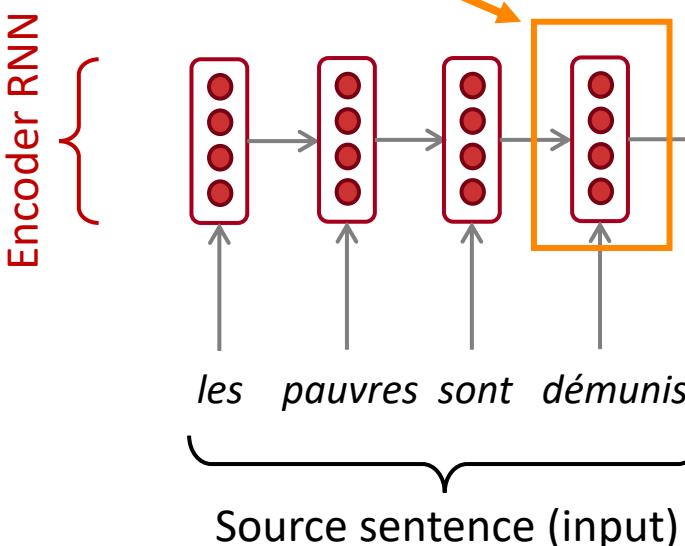
Problems with this architecture?

Sequence-to-sequence: the bottleneck problem

Encoding of the source sentence.

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

Information bottleneck!



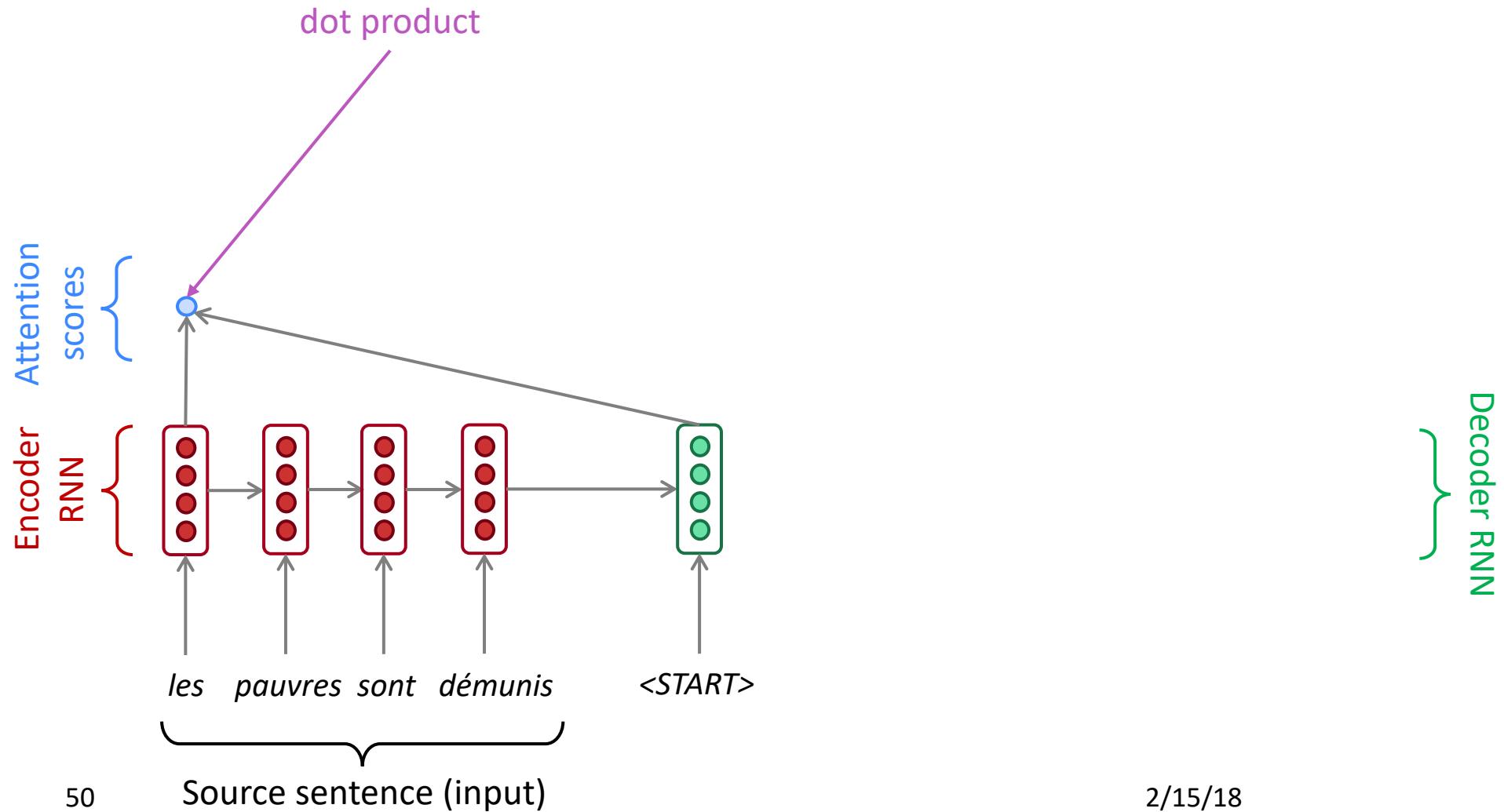
Attention

- **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

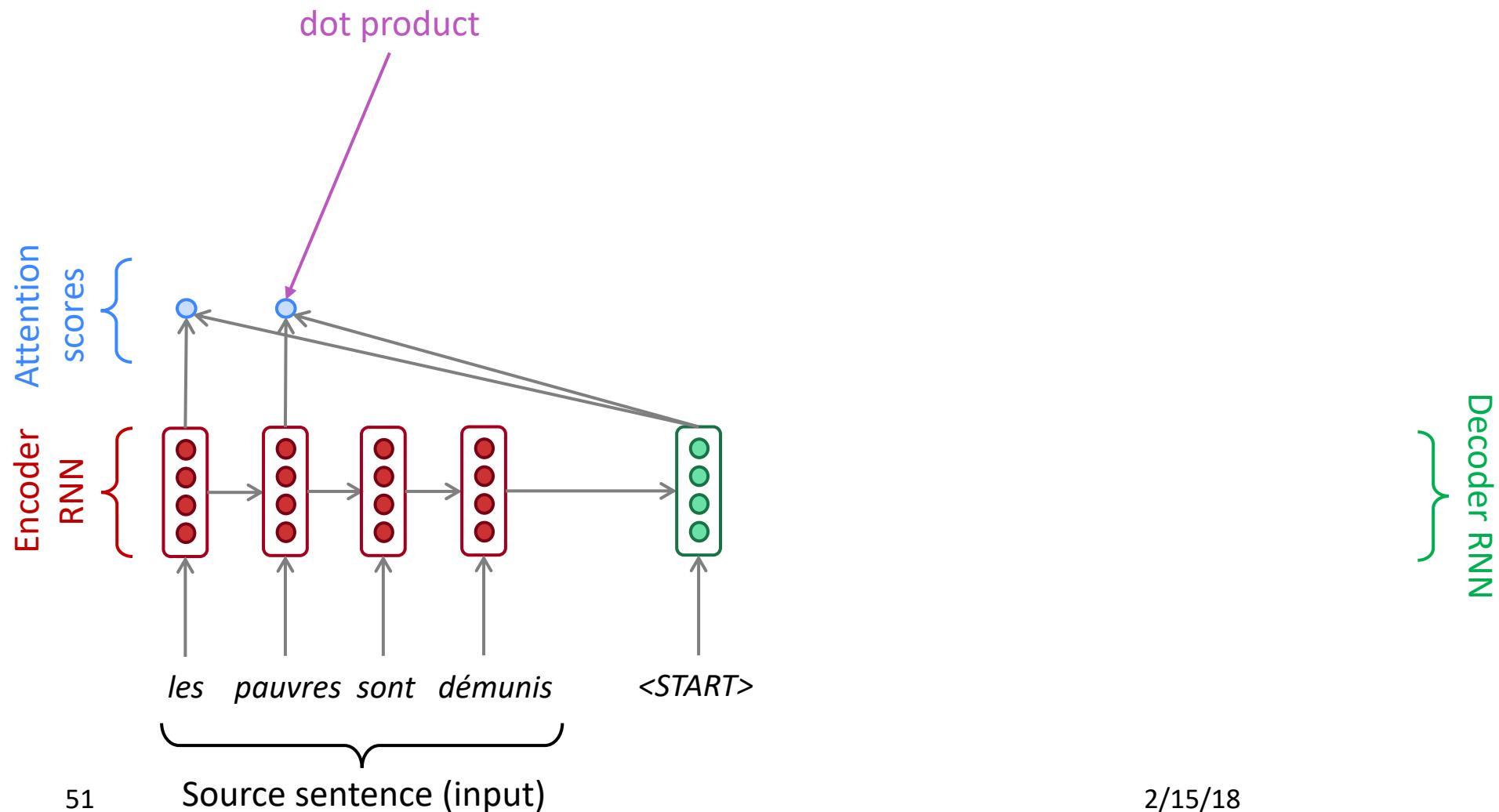


- First we will show via diagram (no equations), then we will show with equations

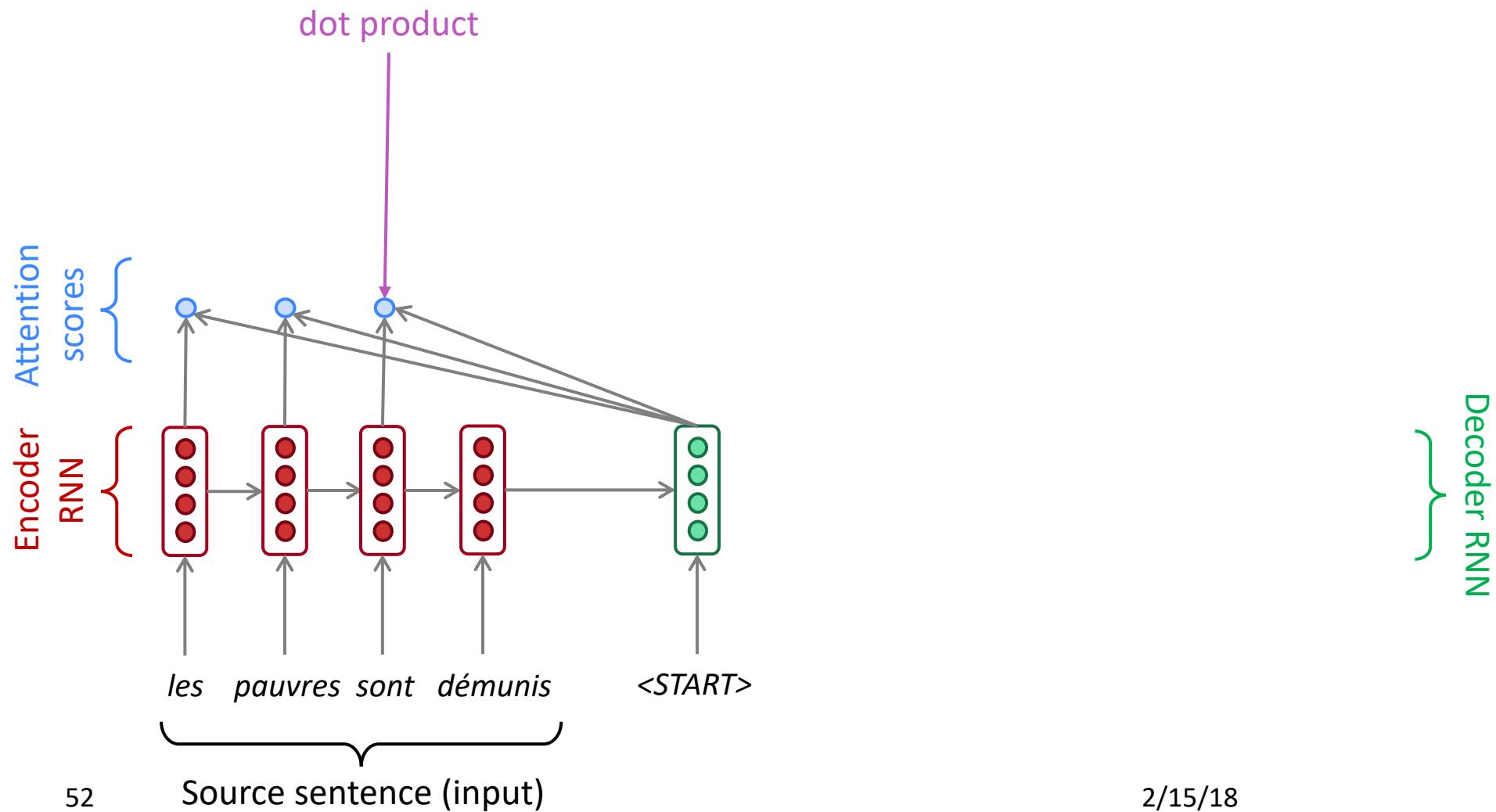
Sequence-to-sequence with attention



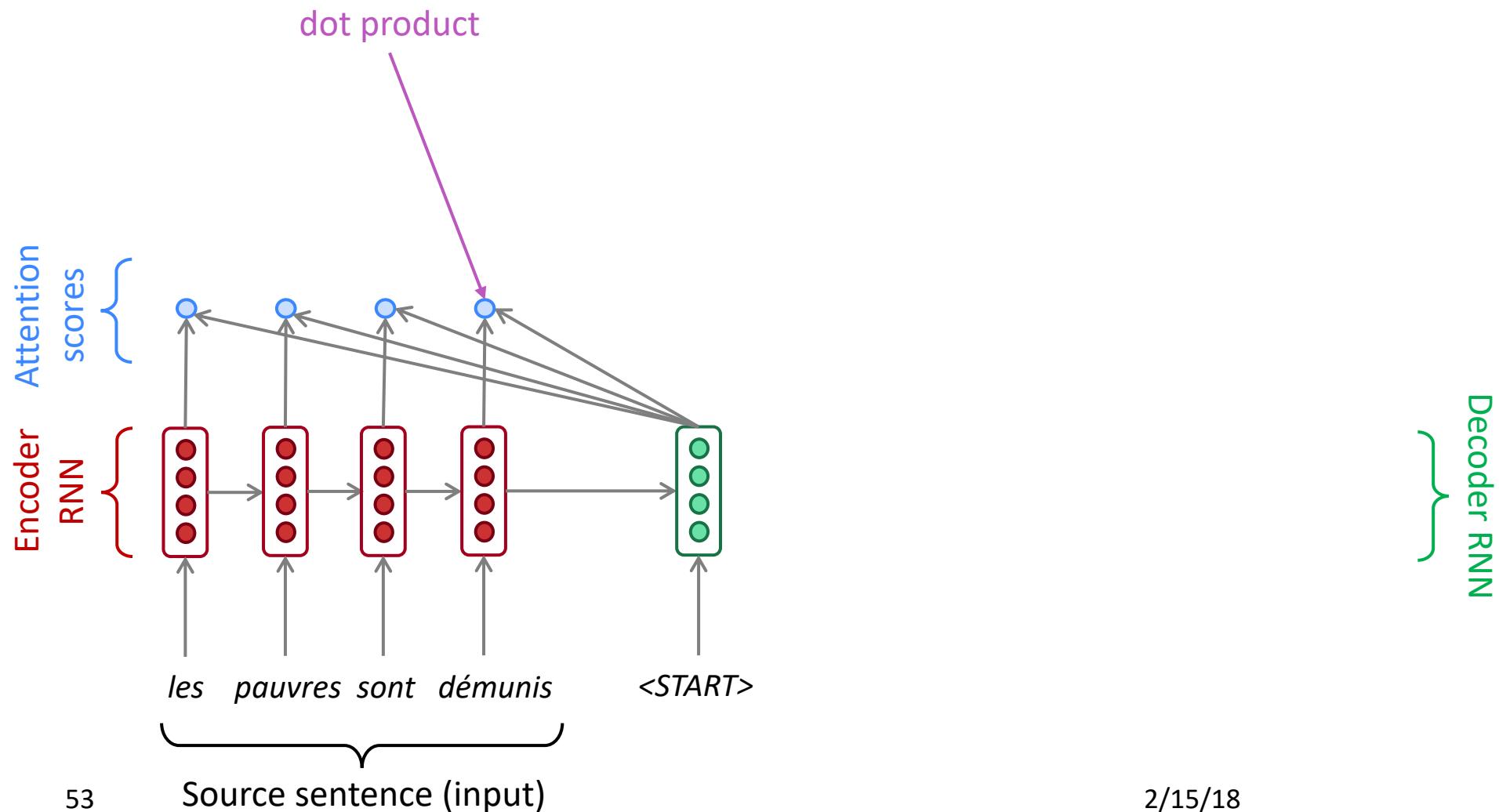
Sequence-to-sequence with attention



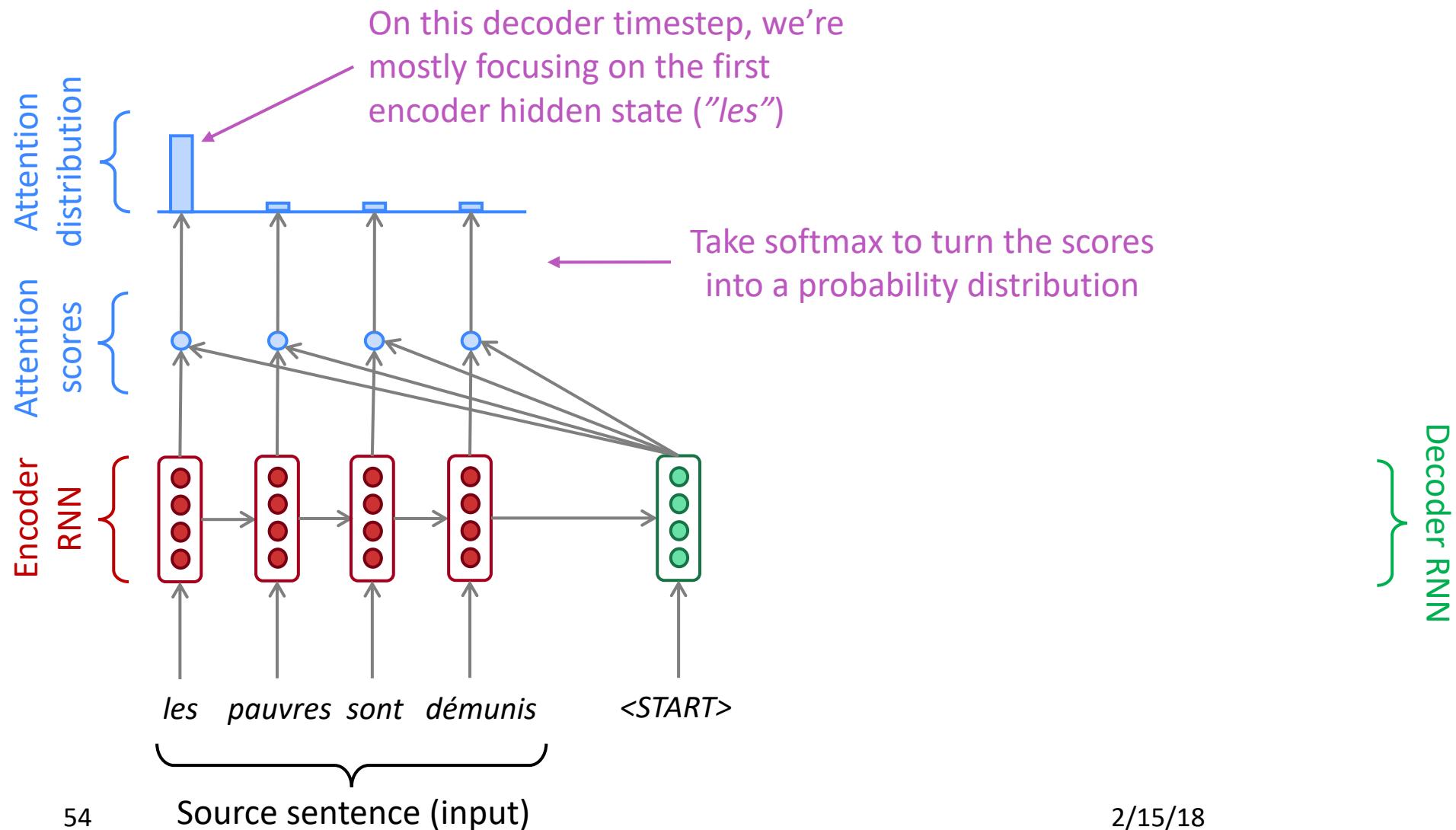
Sequence-to-sequence with attention



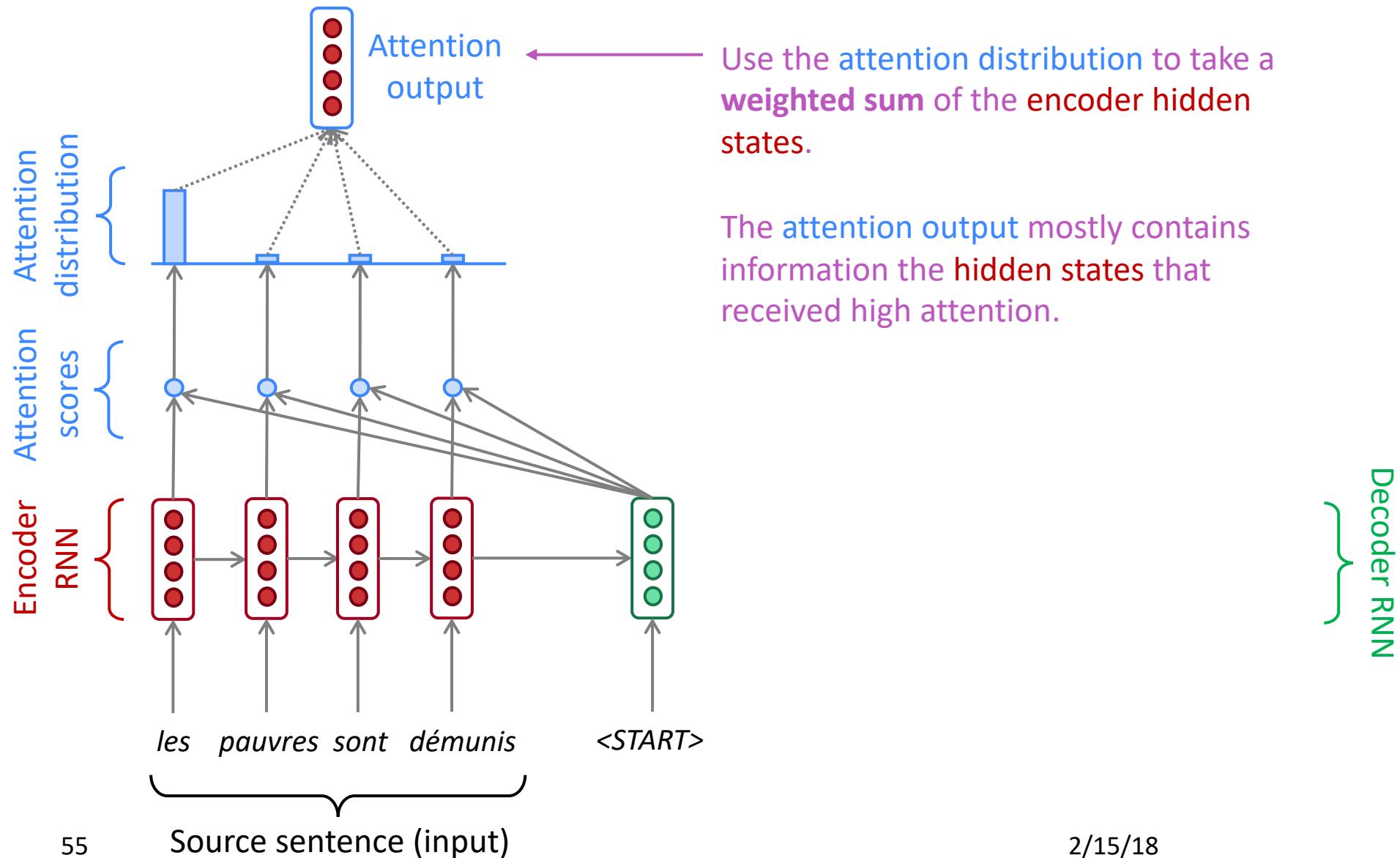
Sequence-to-sequence with attention



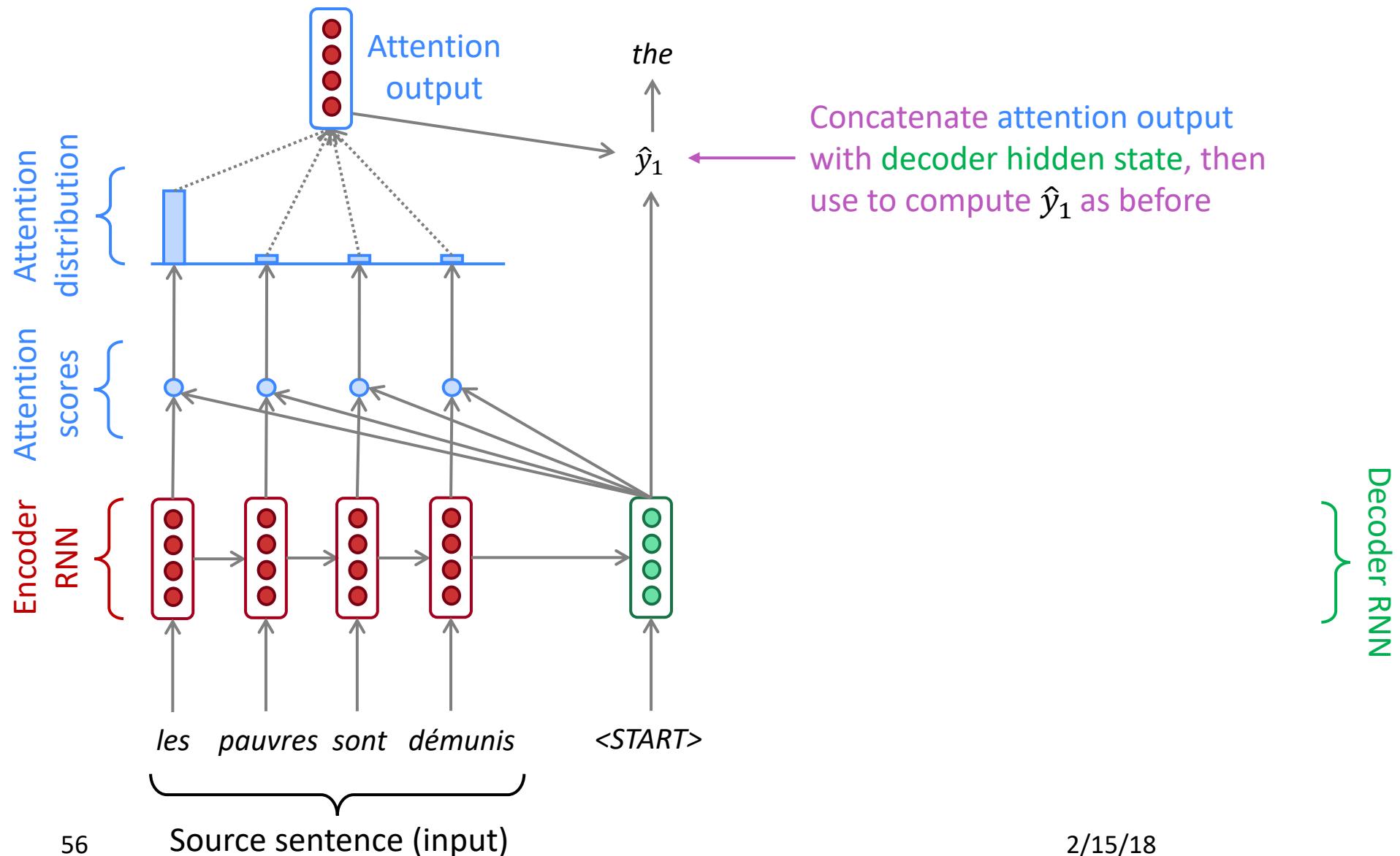
Sequence-to-sequence with attention



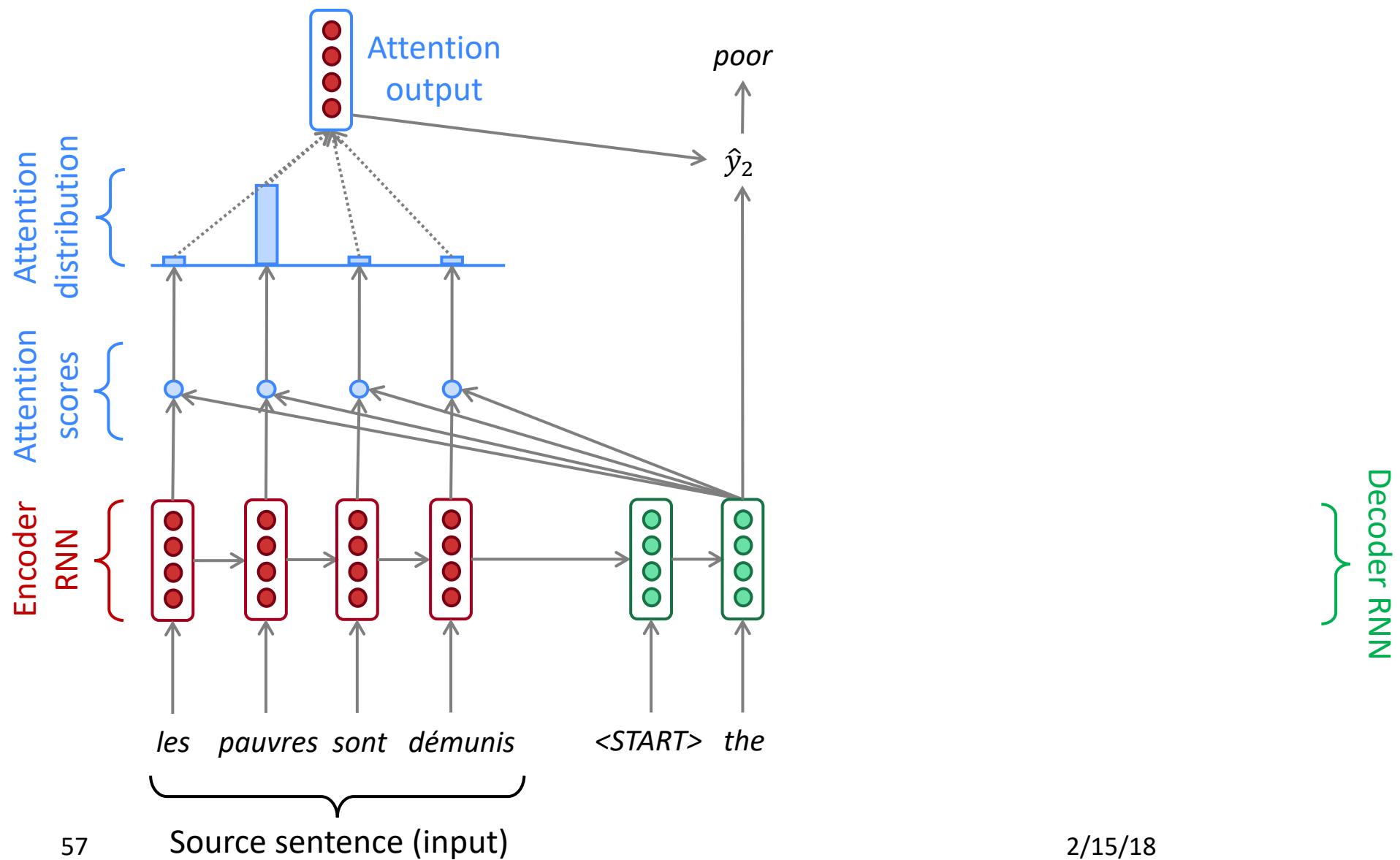
Sequence-to-sequence with attention



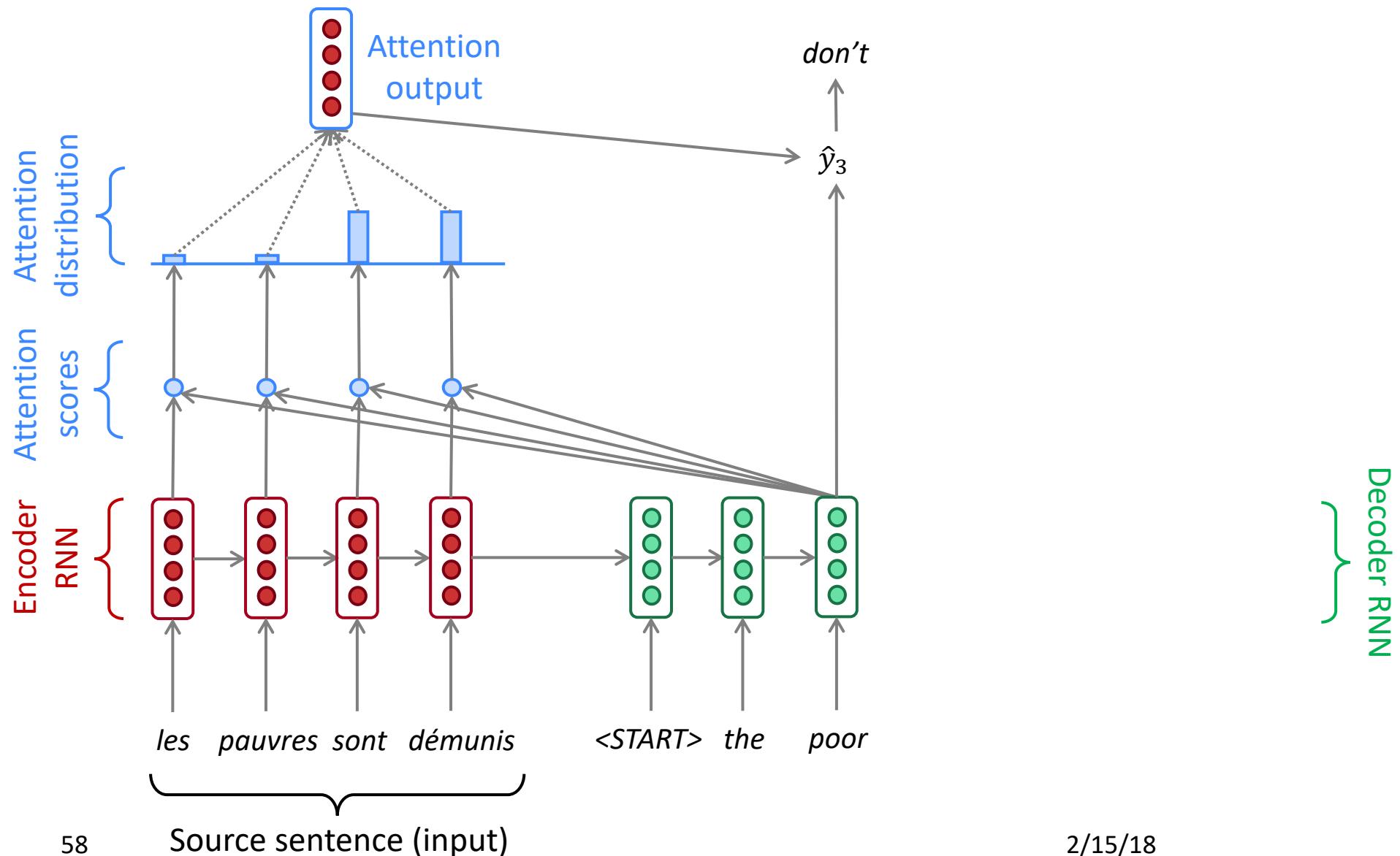
Sequence-to-sequence with attention



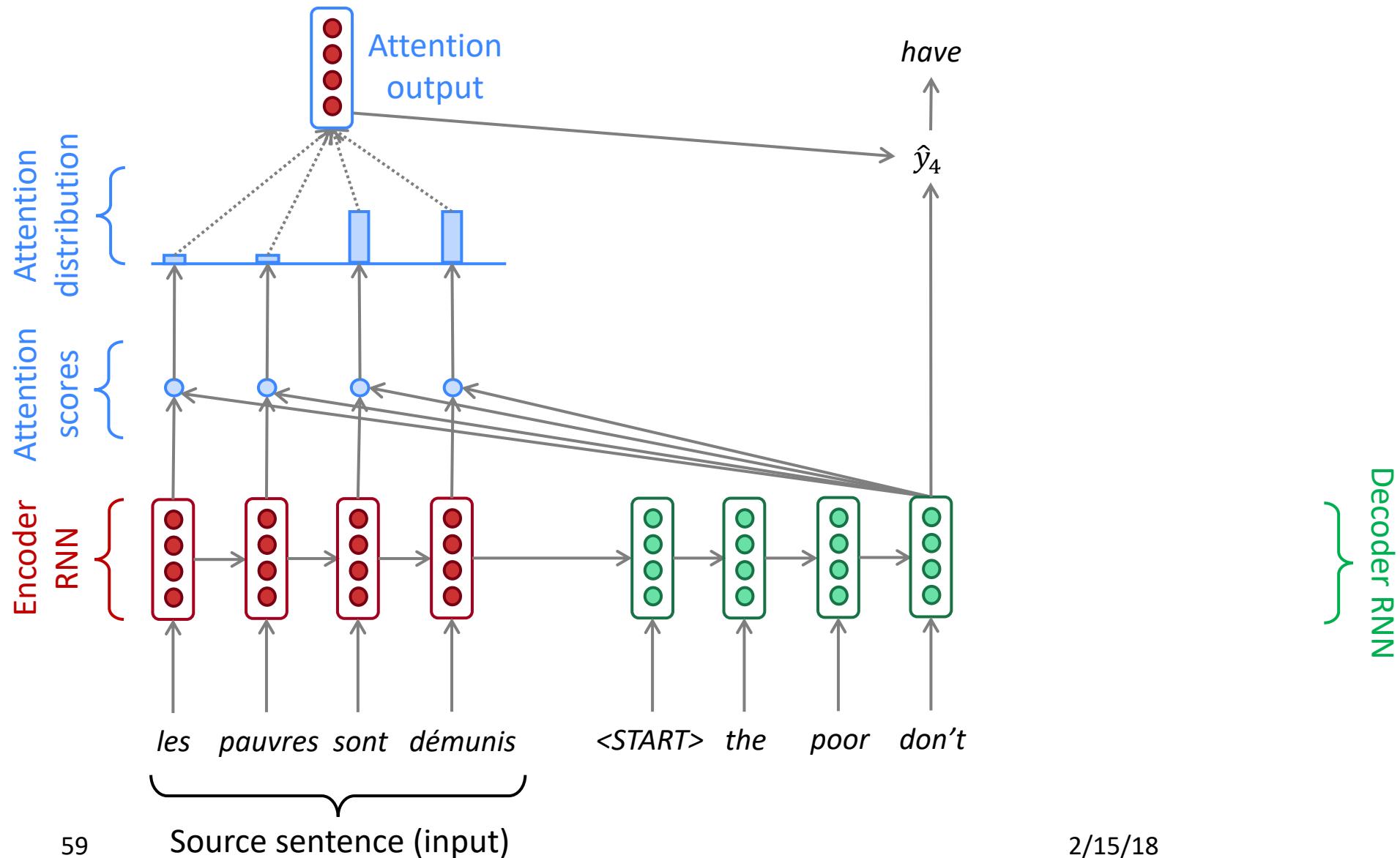
Sequence-to-sequence with attention



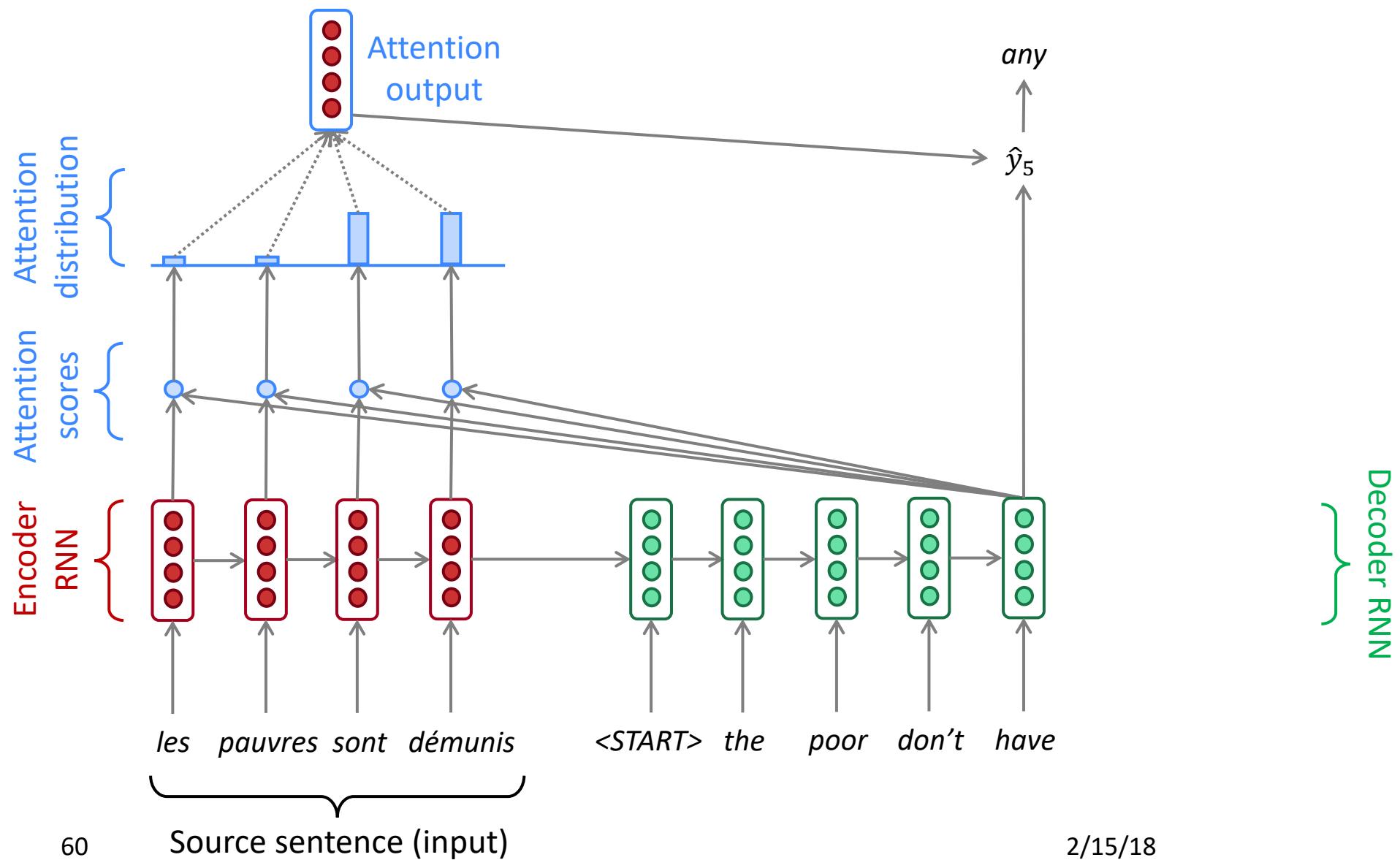
Sequence-to-sequence with attention



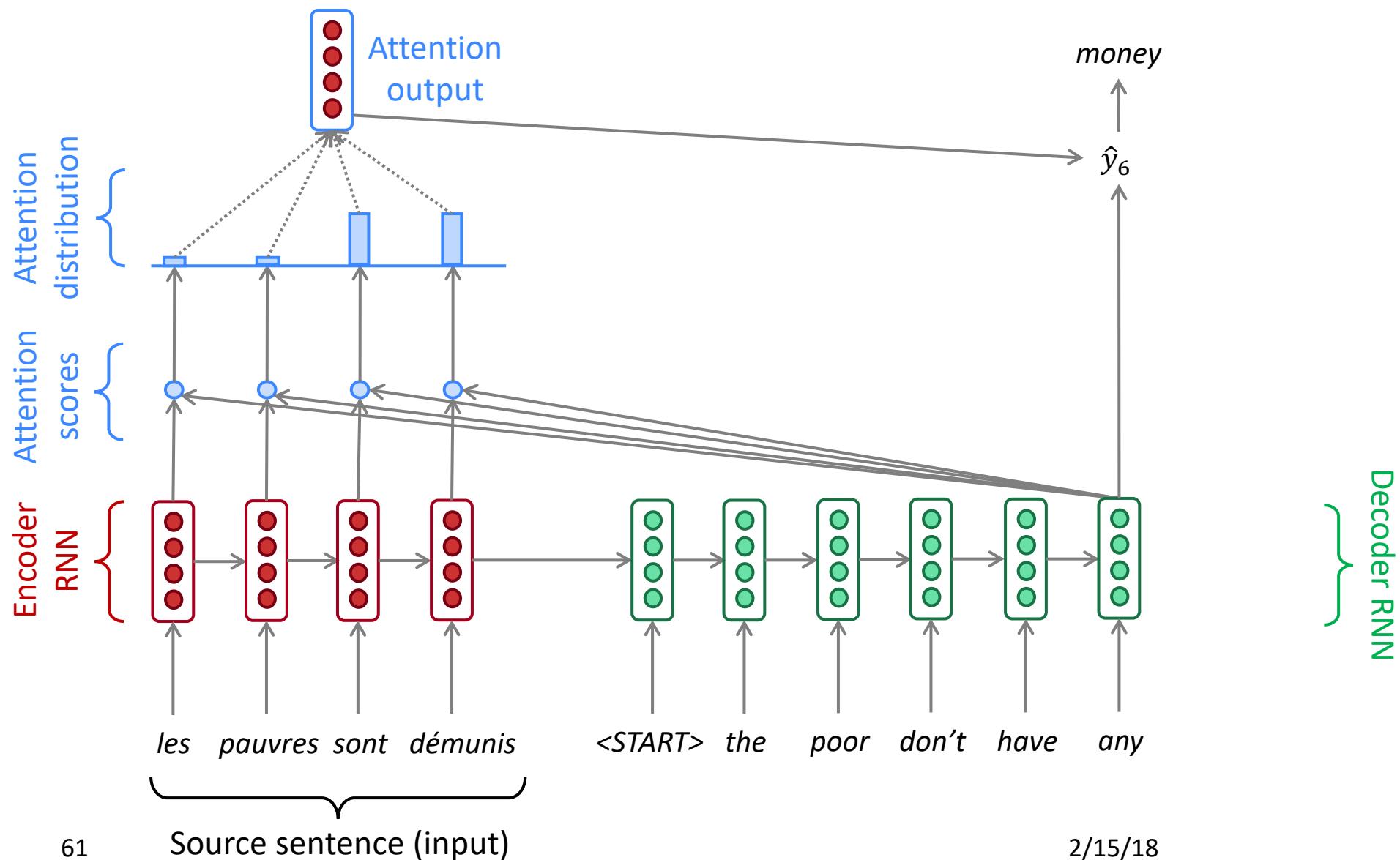
Sequence-to-sequence with attention



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

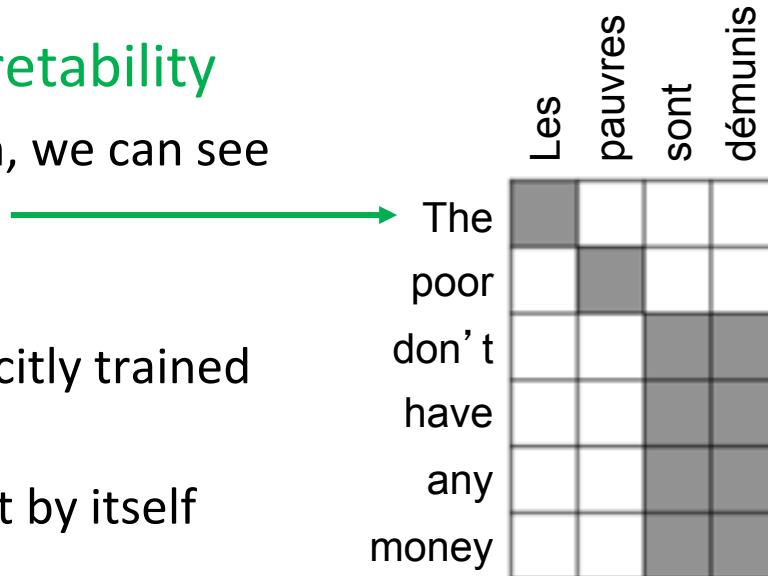
$$a_t = \sum_{i=1}^N \alpha_i^t h_i \in \mathbb{R}^h$$

- 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



Recap

- We learned the history of Machine Translation (MT)
- Since 2014, Neural MT rapidly replaced intricate Statistical MT
- Sequence-to-sequence is the architecture for NMT (uses 2 RNNs)
- Attention is a way to *focus on particular parts* of the input
 - Improves sequence-to-sequence a lot!



Sequence-to-sequence is versatile!

- Sequence-to-sequence is useful for *more than just MT*
- Many NLP tasks can be phrased as sequence-to-sequence:
 - **Summarization** (long text → short text)
 - **Dialogue** (previous utterances → next utterance)
 - **Parsing** (input text → output parse as sequence)
 - **Code generation** (natural language → Python code)

Next time

- More **types** of attention
- More **uses** for attention
- More advanced sequence-to-sequence techniques