CS20: TensorFlow for Deep Learning Research

Lecture 12 (2/23/2014)
Machine Translation,
Sequence-to-sequence and Attention

Slides courtesy of CS22N

Assignment 3

- Chat bot
- Language model
- Word vector transformation
- Project of choice

Overview

Today we will:

- Introduce a new task: Machine Translation

is the primary use-case of

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

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:

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

Translation Model

Models how words and phrases should be translated.
Learnt from parallel data.

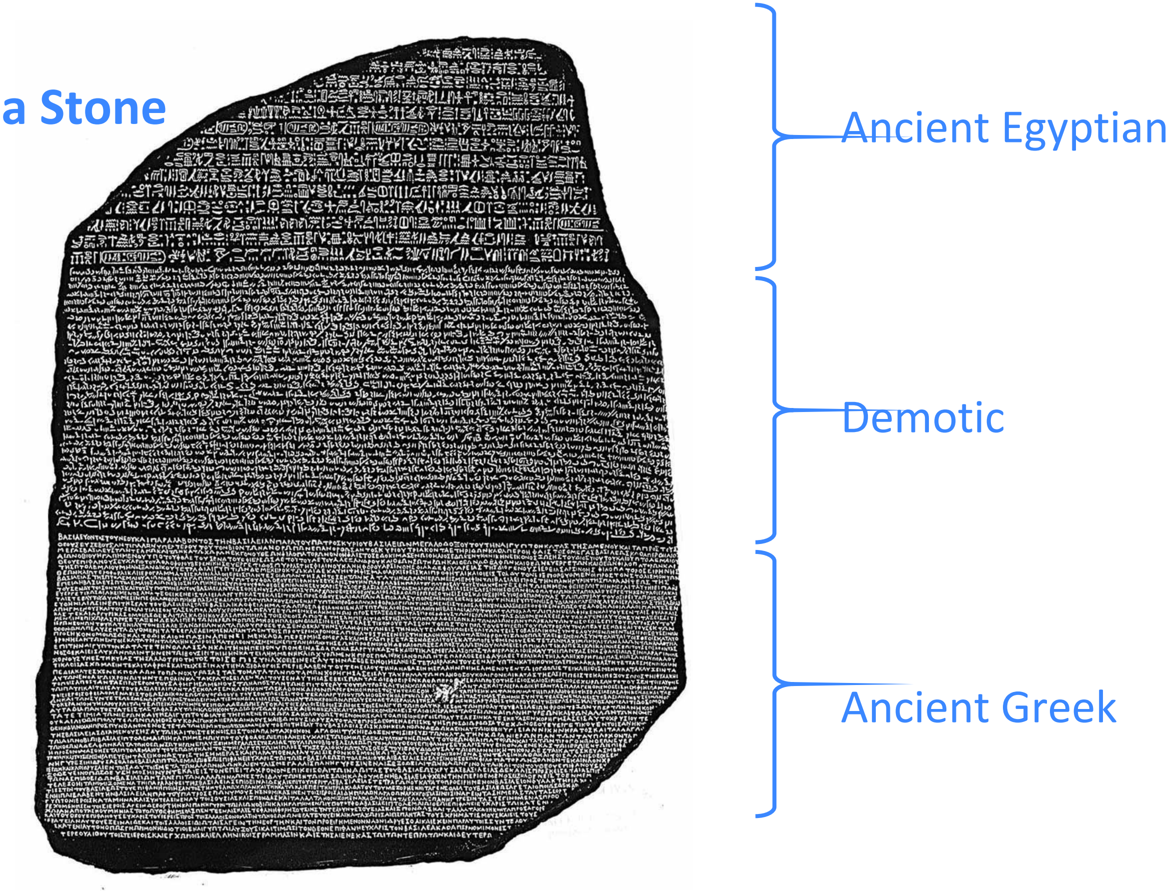
Language Model

Models how to write good English.
Learnt from monolingual data.

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

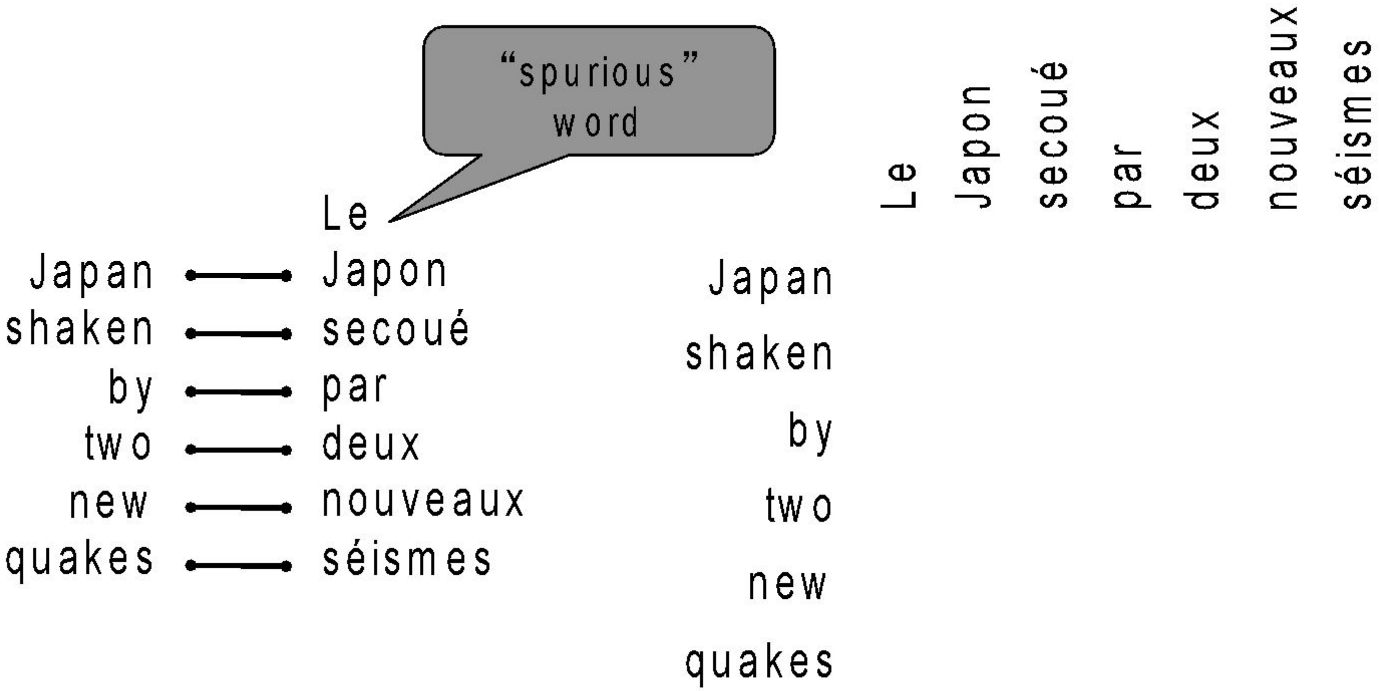
$$P(x, a|y)$$

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

What is alignment?

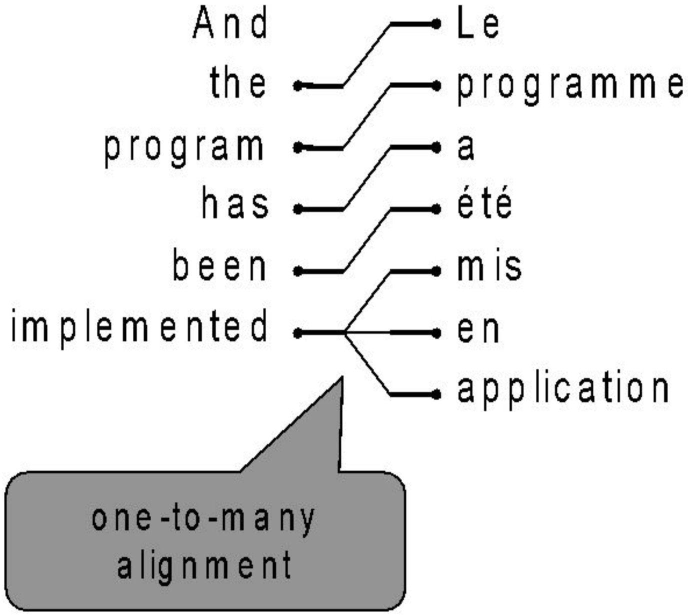
Alignment is the correspondence between particular words in the translated sentence pair.

- Note: Some words have no counterpart



Alignment is complex

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



And
the
program
has
been
implemented

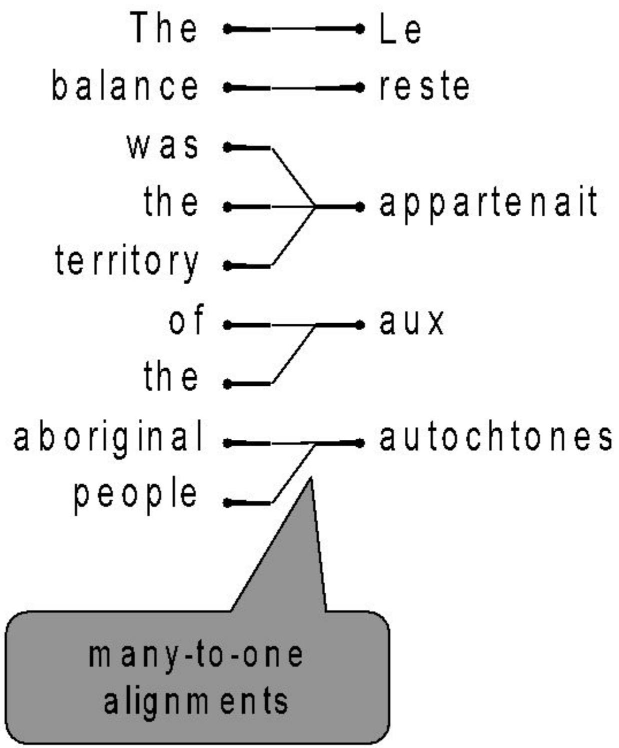
Le
programme
a
été
mis
en
application

one-to-many alignment

	Le	programme	a	été	mis	en	application
And							
the							
program							
has							
been							
implemented							

Alignment is complex

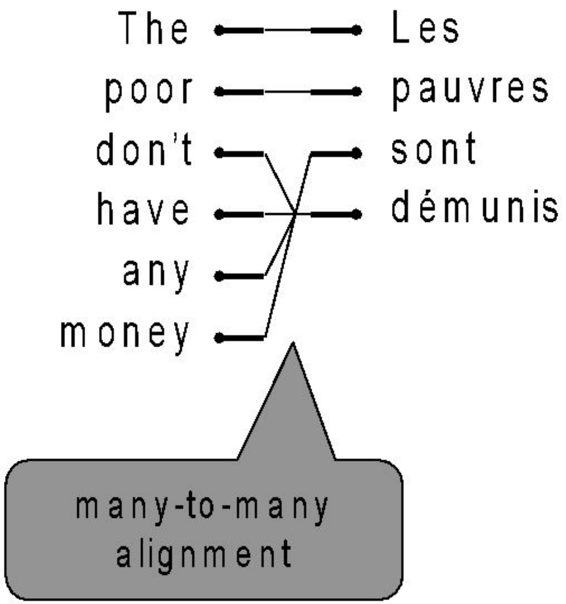
Alignment can be many-to-one

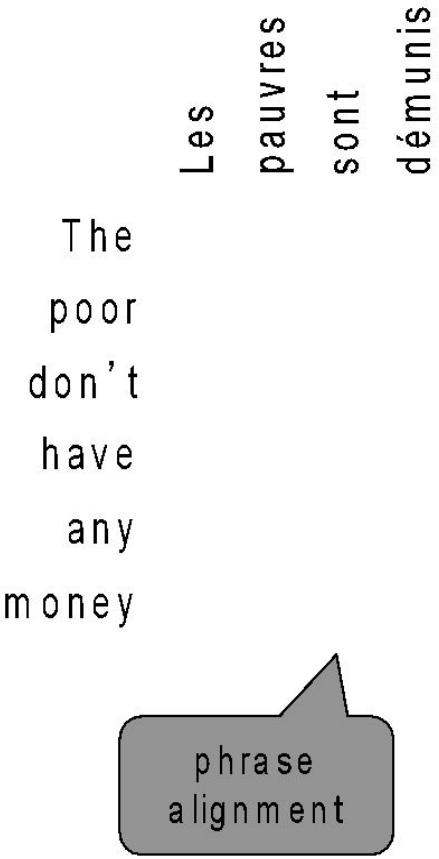


	Le	reste	appartenait	aux	autochtones
The					
balance					
was					
the					
territory					
of					
the					
aboriginal					
people					

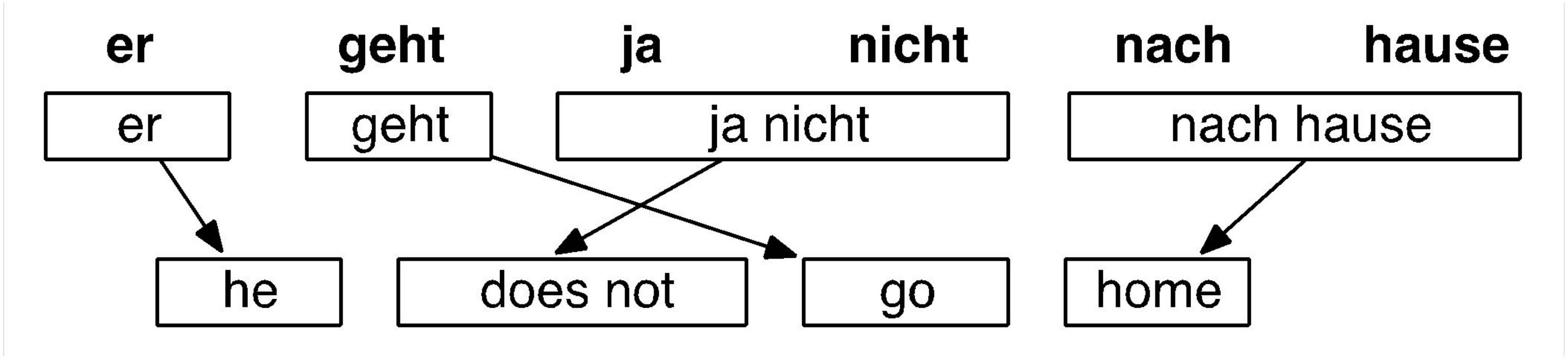
Alignment is complex

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

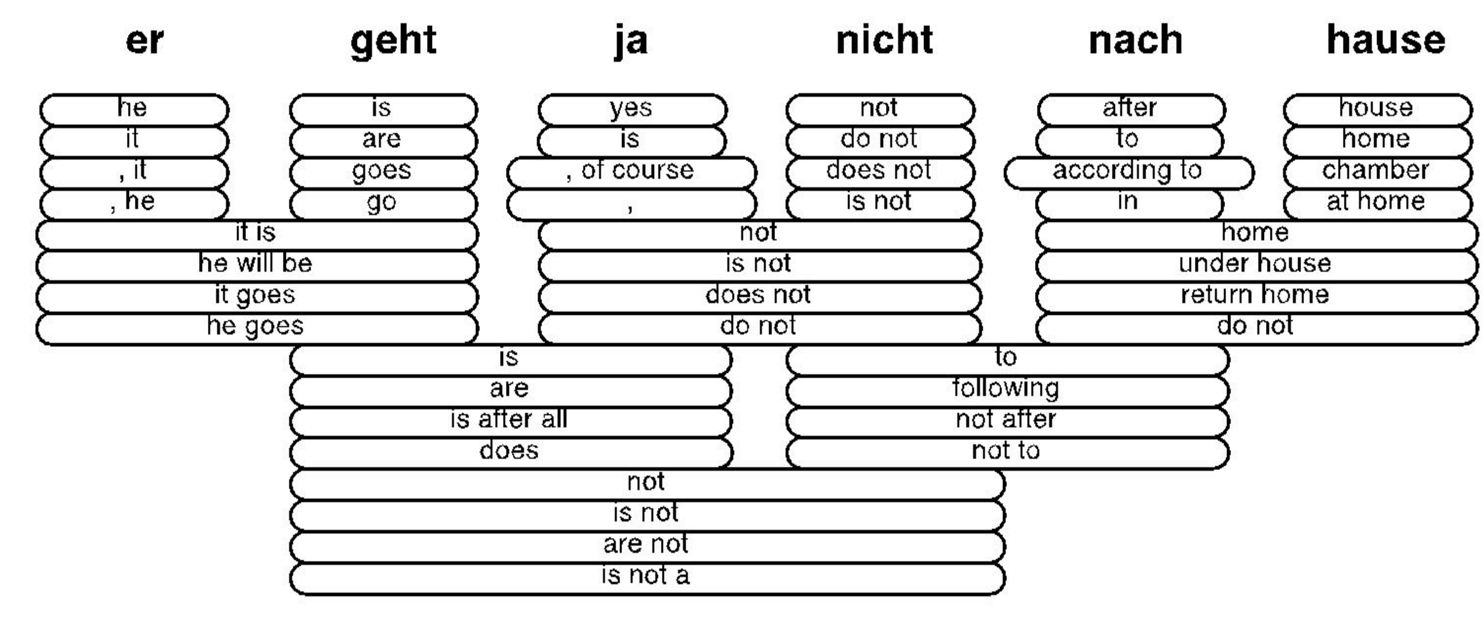




Searching for the best translation



Searching for the best translation



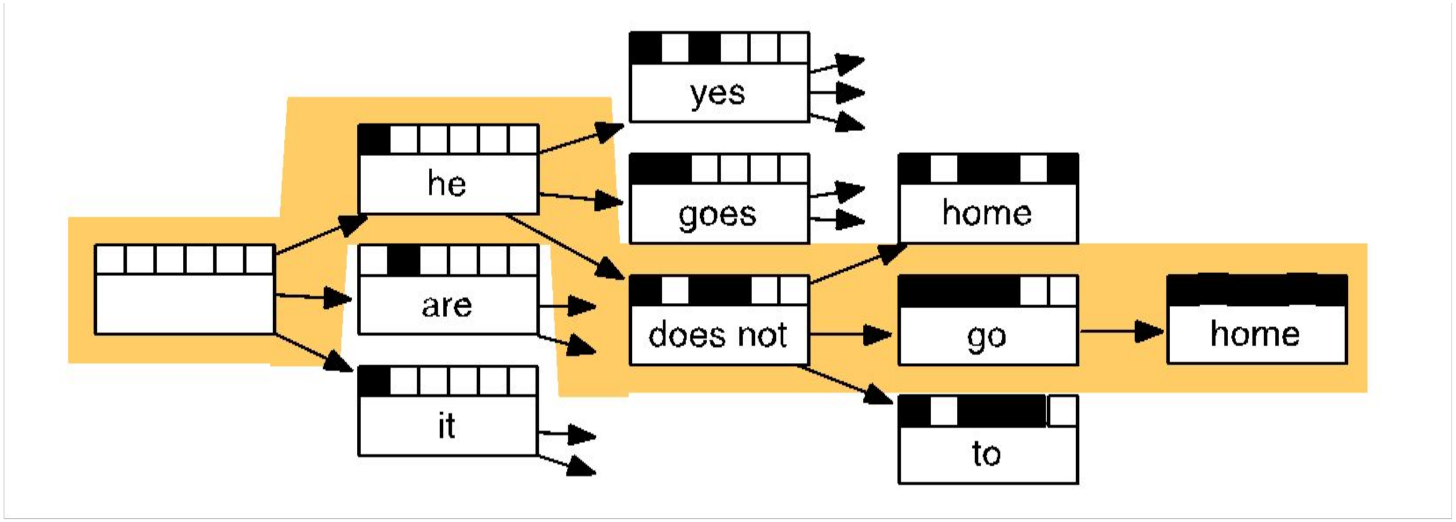
er geht ja nicht nach hause

he | is | yes | not | after | house
it | are | is | do not | to | home
, it | goes | , of course | does not | according to | chamber
, he | go | , | is not | in | at home

it is | not | home
he will be | is not | under house
it goes | does not | return home
he goes | do not | do not

is | to
are | following
is after all | not after
does | not to

not
is not
are not
is not a

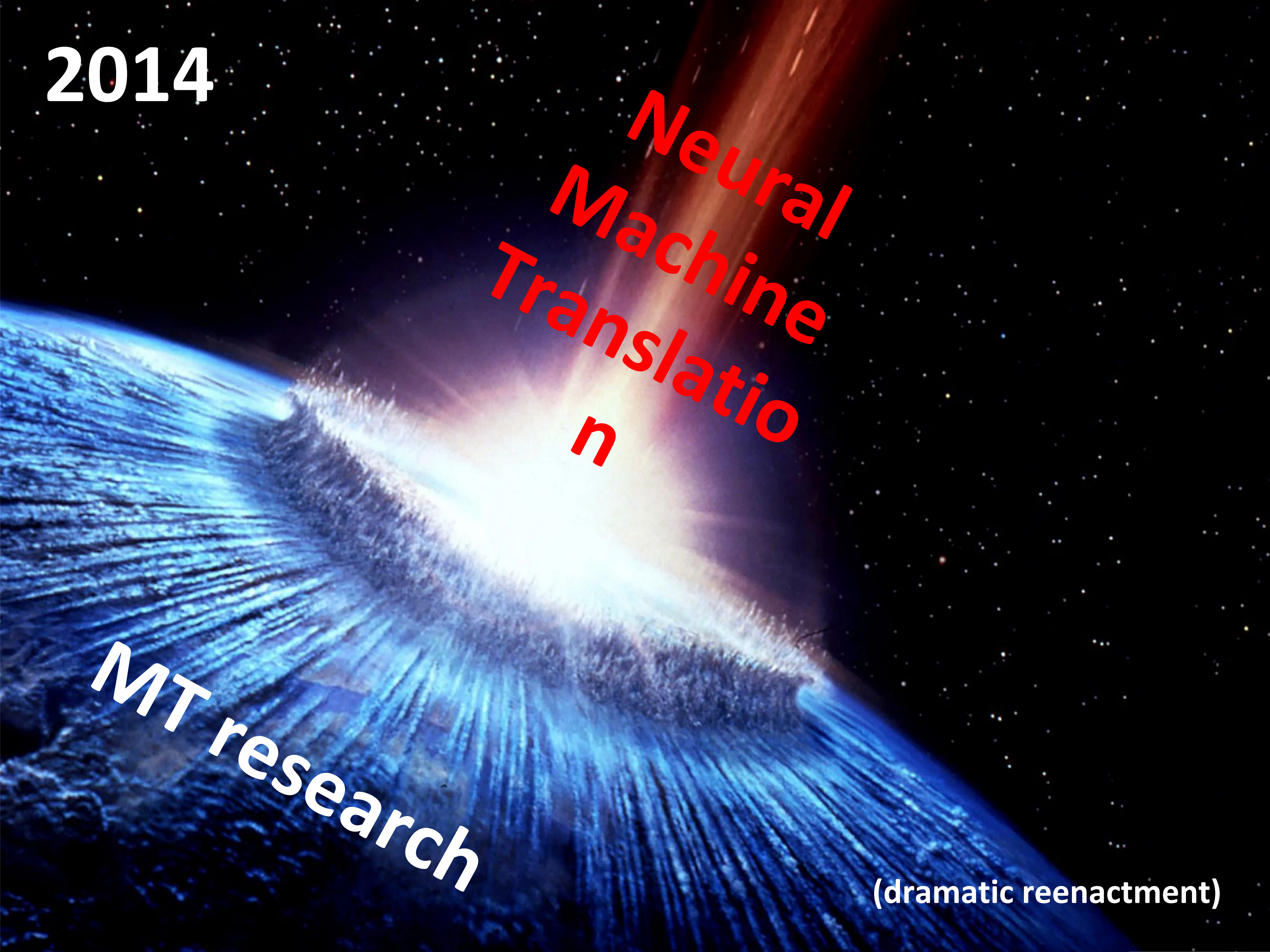


1990s-2010s: Statistical Machine Translation

- 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
Translatio
n

MT research

(dramatic reenactment)

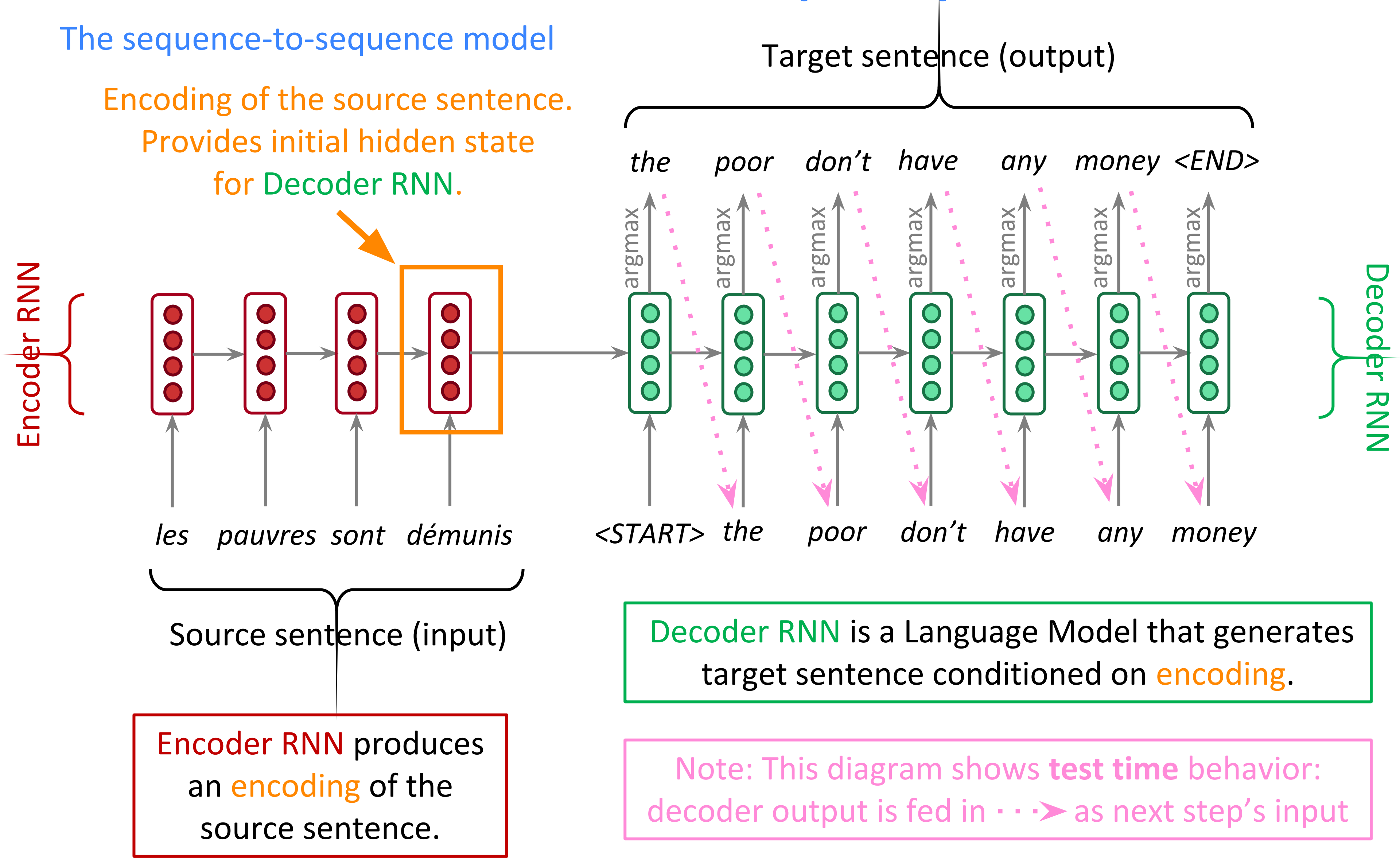
What is Neural Machine Translation?

- 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)

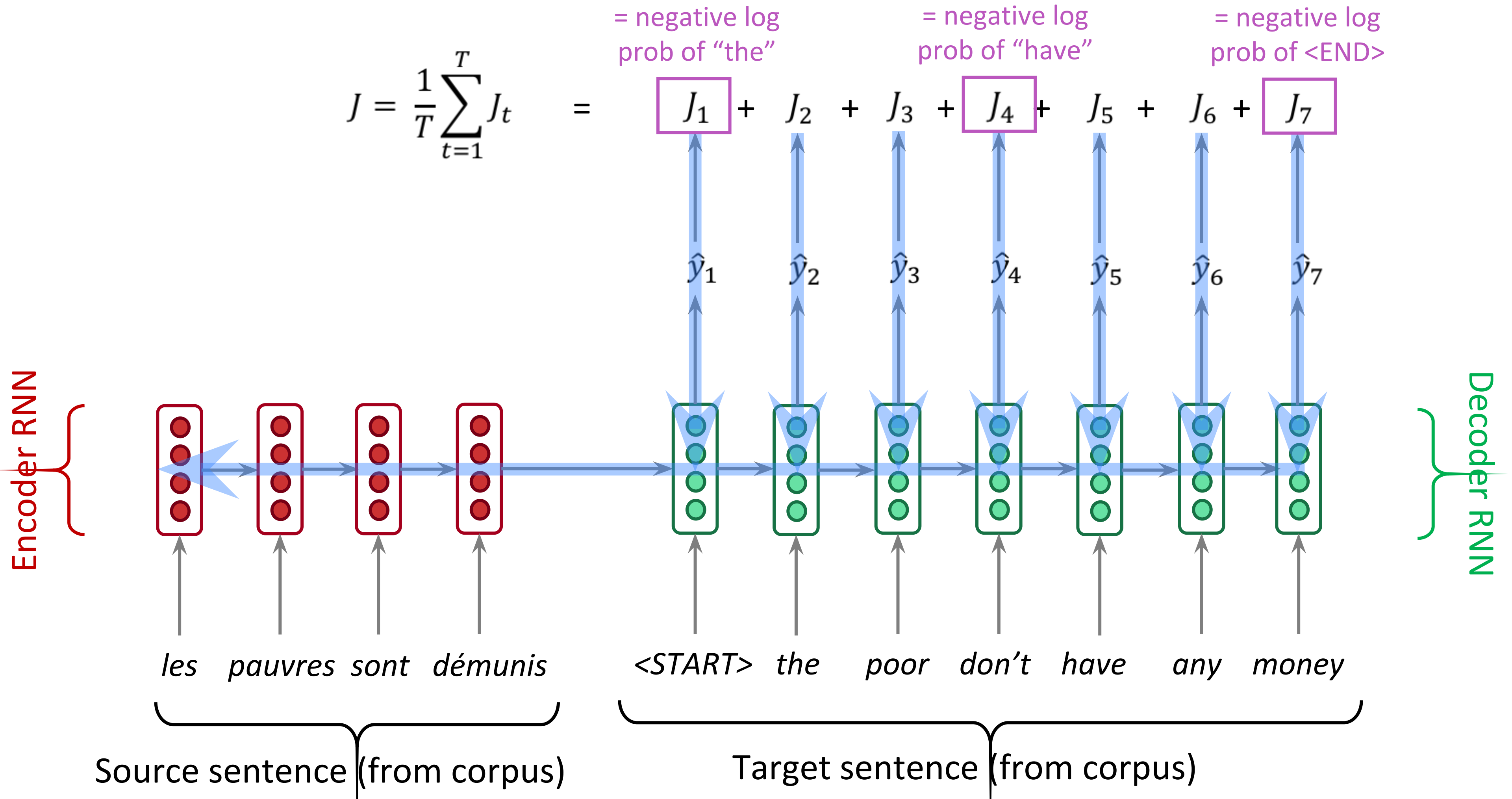
- The sequence-to-sequence model is an example of a **Conditional Language Model**.
  - **Language Model** because the decoder is predicting the next word of the target sentence *y*
  - **Conditional** because its predictions are *also* conditioned on the source sentence *x*

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

$$P(y|x) = P(y_1|x)\, P(y_2|y_1, x)\, P(y_3|y_1, y_2, x) \ldots, \underbrace{P(y_T|y_1, \ldots, y_{T-1}, x)}_{\substack{\text{Probability of next target word, given} \\ \text{target words so far and source sentence } x}}$$

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

Training a Neural Machine Translation system



= negative log prob of "the"

= negative log prob of "have"

= negative log prob of <END>

$$J = \frac{1}{T}\sum_{t=1}^{T} J_t \quad = \quad \boxed{J_1} + J_2 + J_3 + \boxed{J_4} + J_5 + J_6 + \boxed{J_7}$$

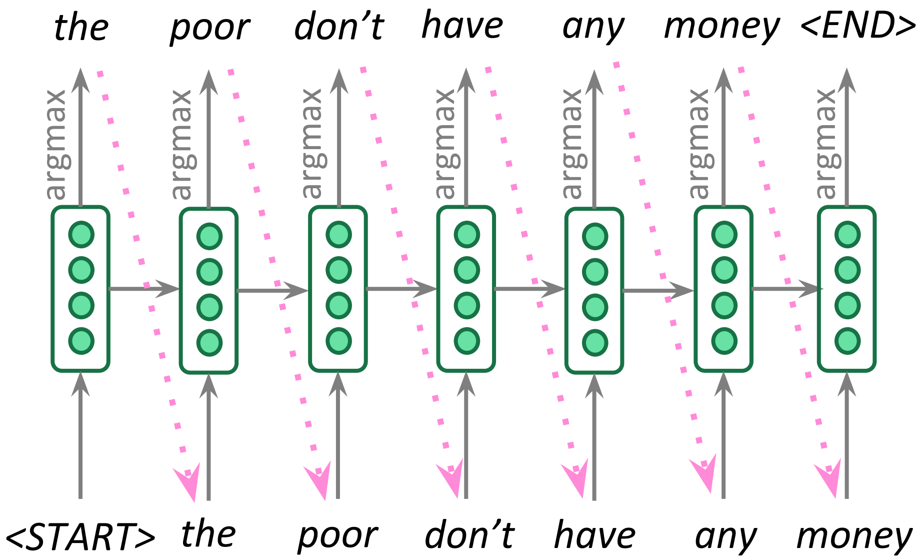
Source sentence (from corpus)

Target sentence (from corpus)

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

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

- We could try enumerating all *y* → 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!

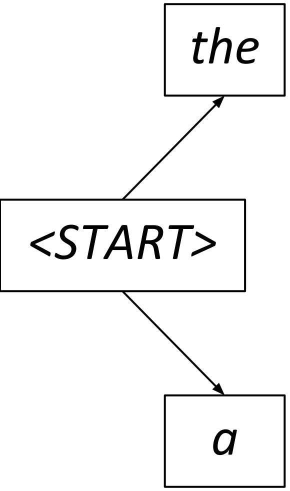
Beam search decoding: example

Beam size = 2

<START>

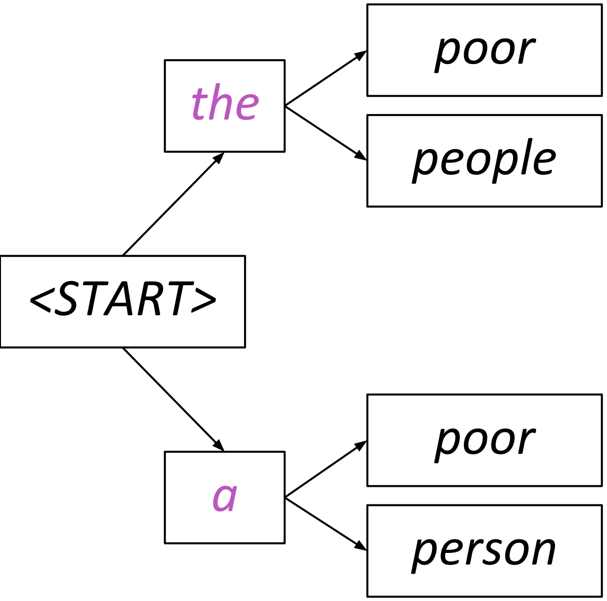
Beam search decoding: example

Beam size = 2



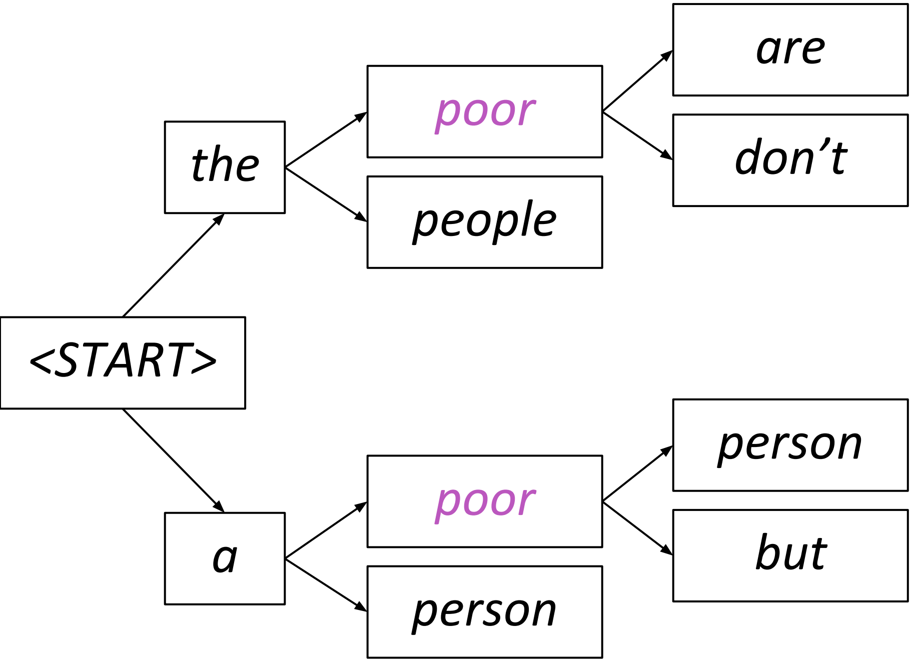
Beam search decoding: example

Beam size = 2



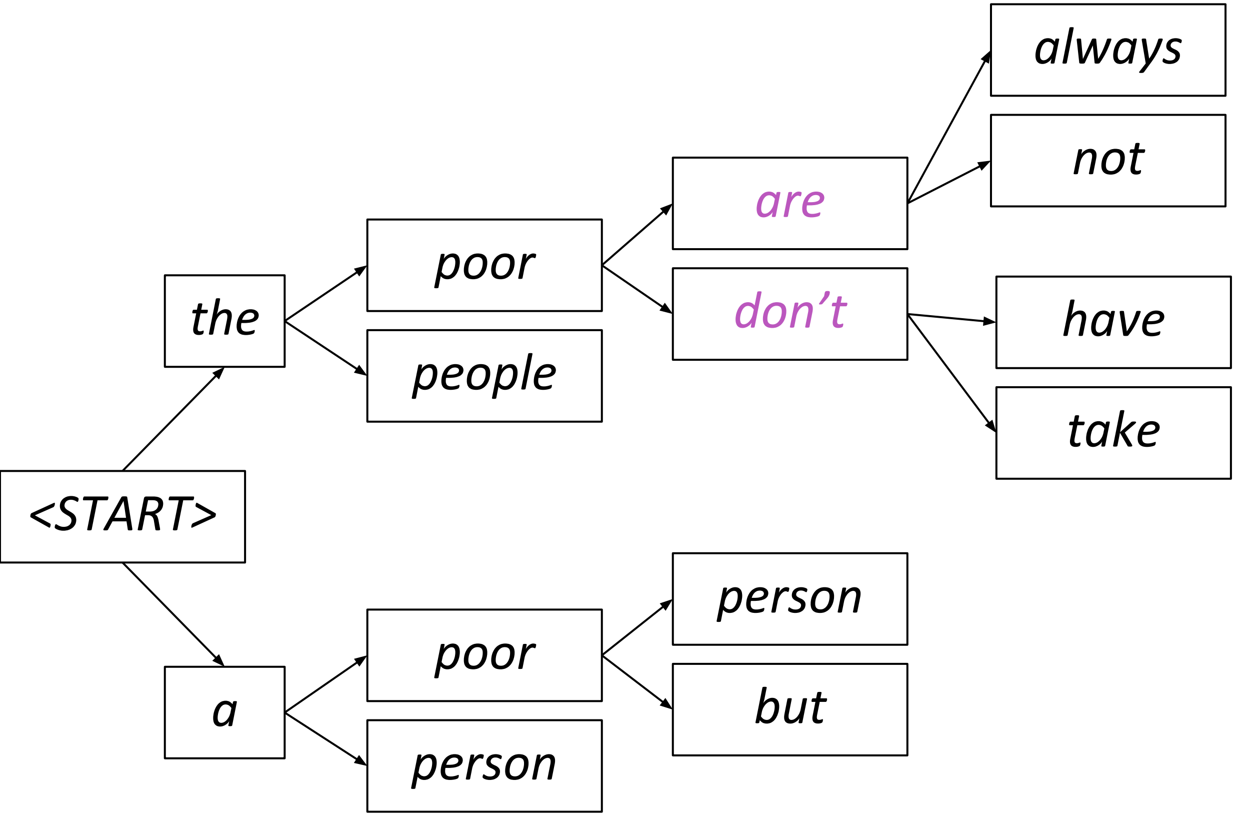
Beam search decoding: example

Beam size = 2



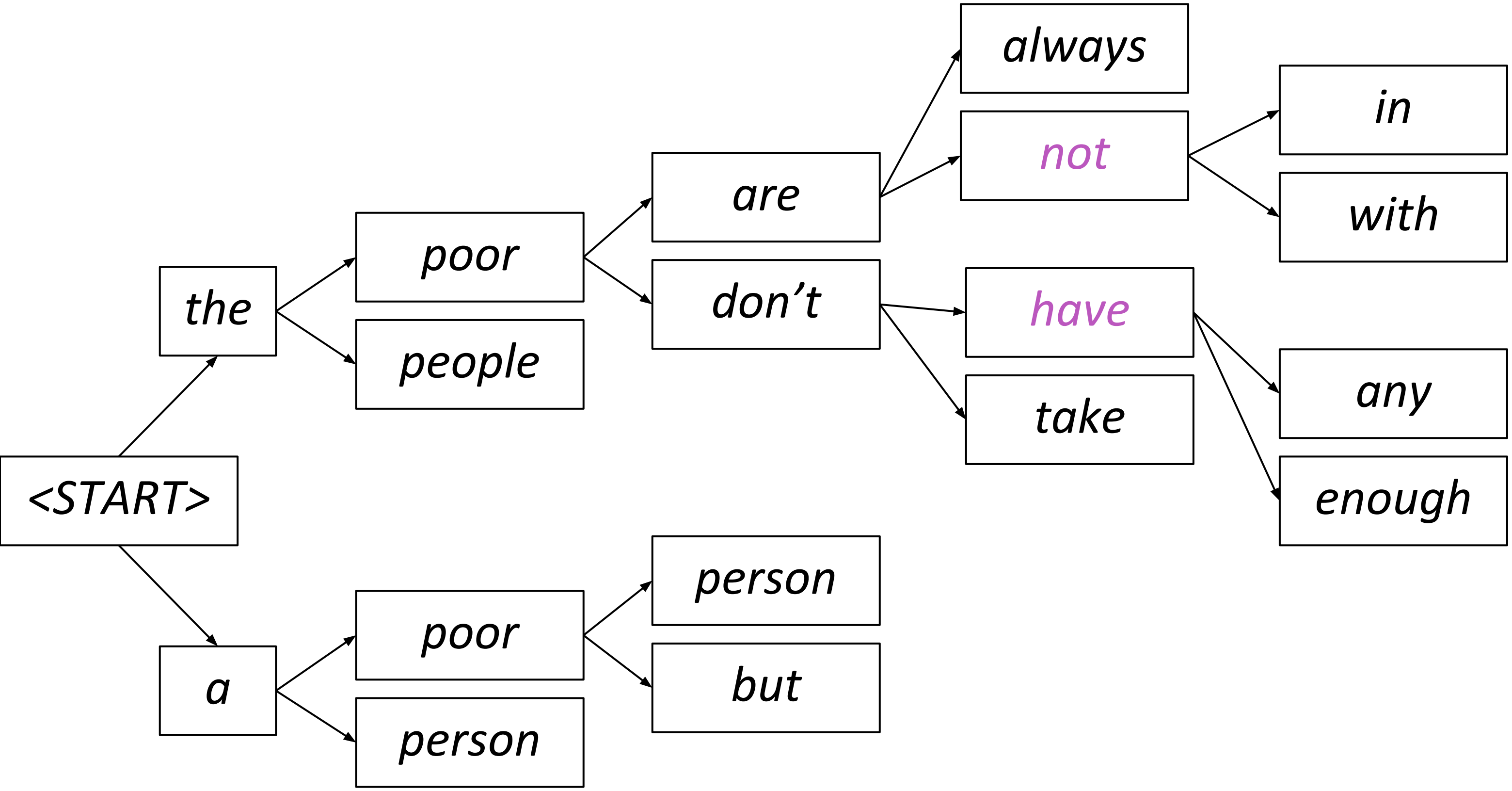
Beam search decoding: example

Beam size = 2



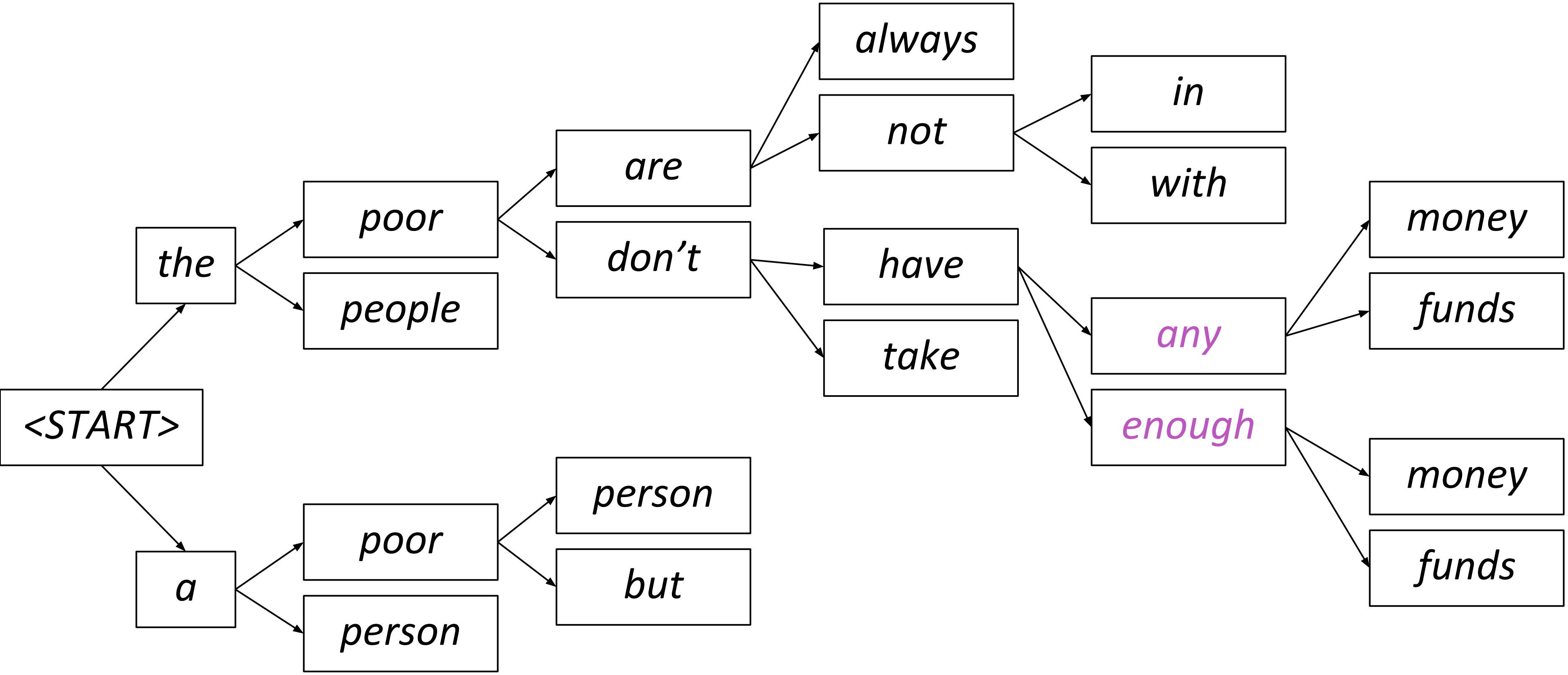
Beam search decoding: example

Beam size = 2



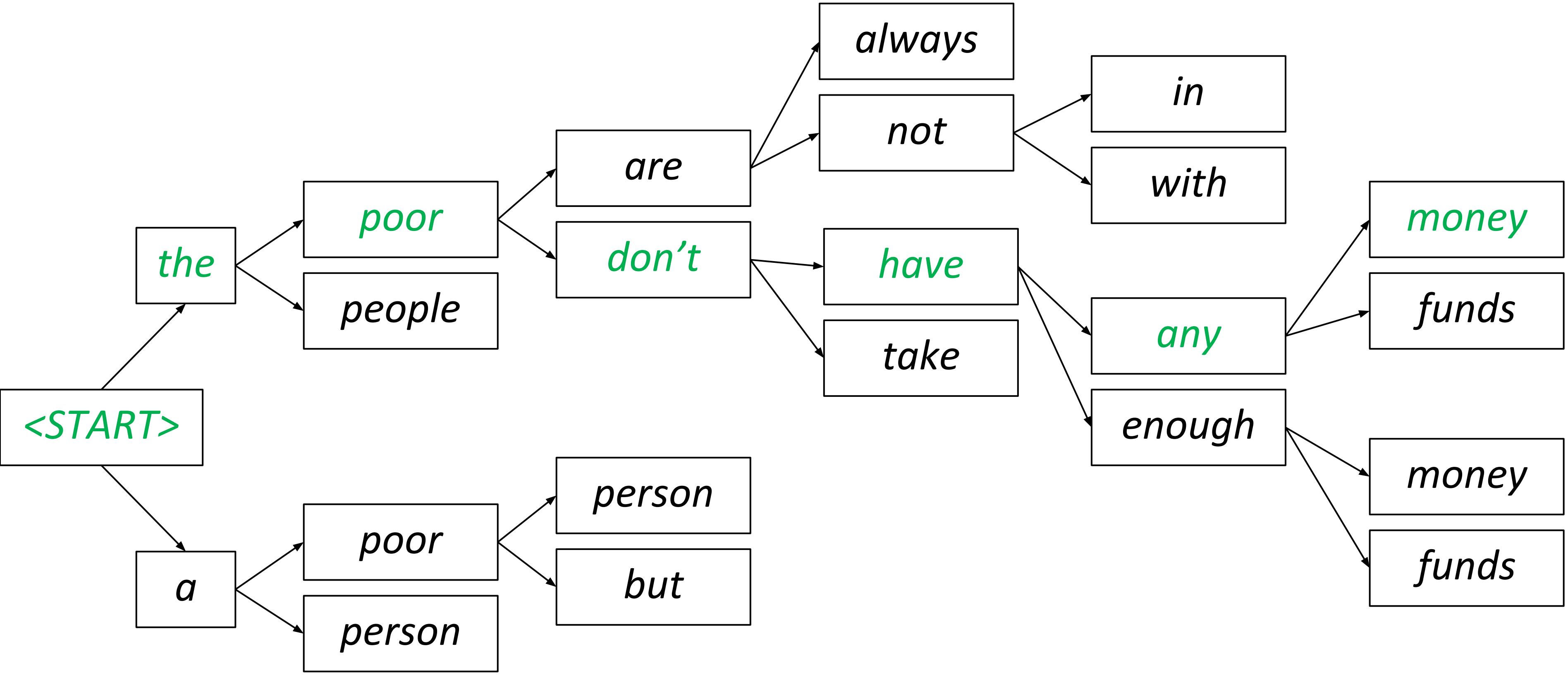
Beam search decoding: example

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!

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!

SMT is still very much in use!

How do we evaluate Machine Translation?

BLEU (**B**i**l**ingual **E**valuation **U**nderstudy)

- 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

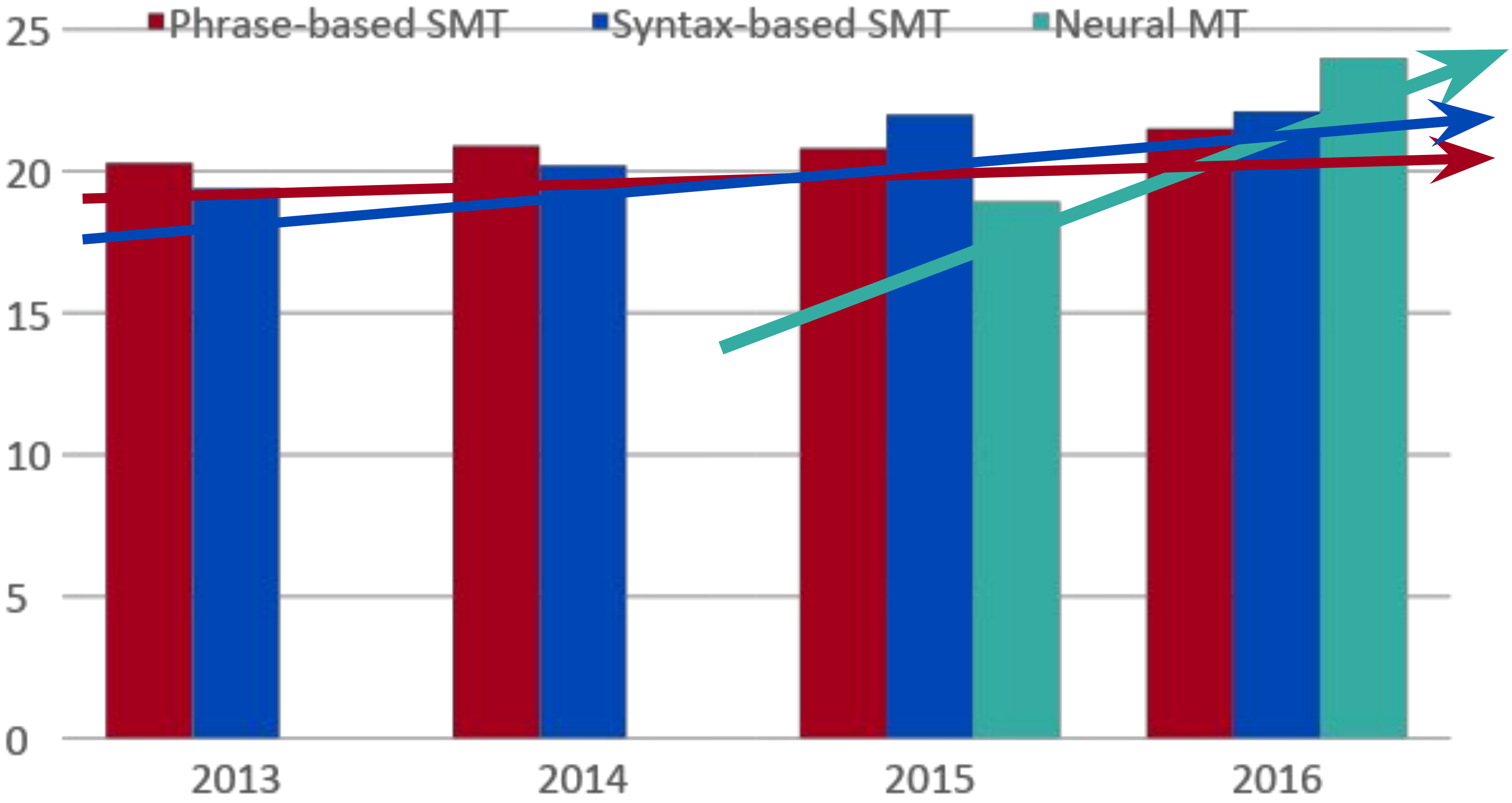
- BLEU is useful but imperfect
  - There are many valid ways to translate a sentence
  - So a good translation can get a poor BLEU score because it has low *n*-gram overlap with the human translation ☹

Beyond BLEU

- Its own area of research
- Thought: metric without reference texts

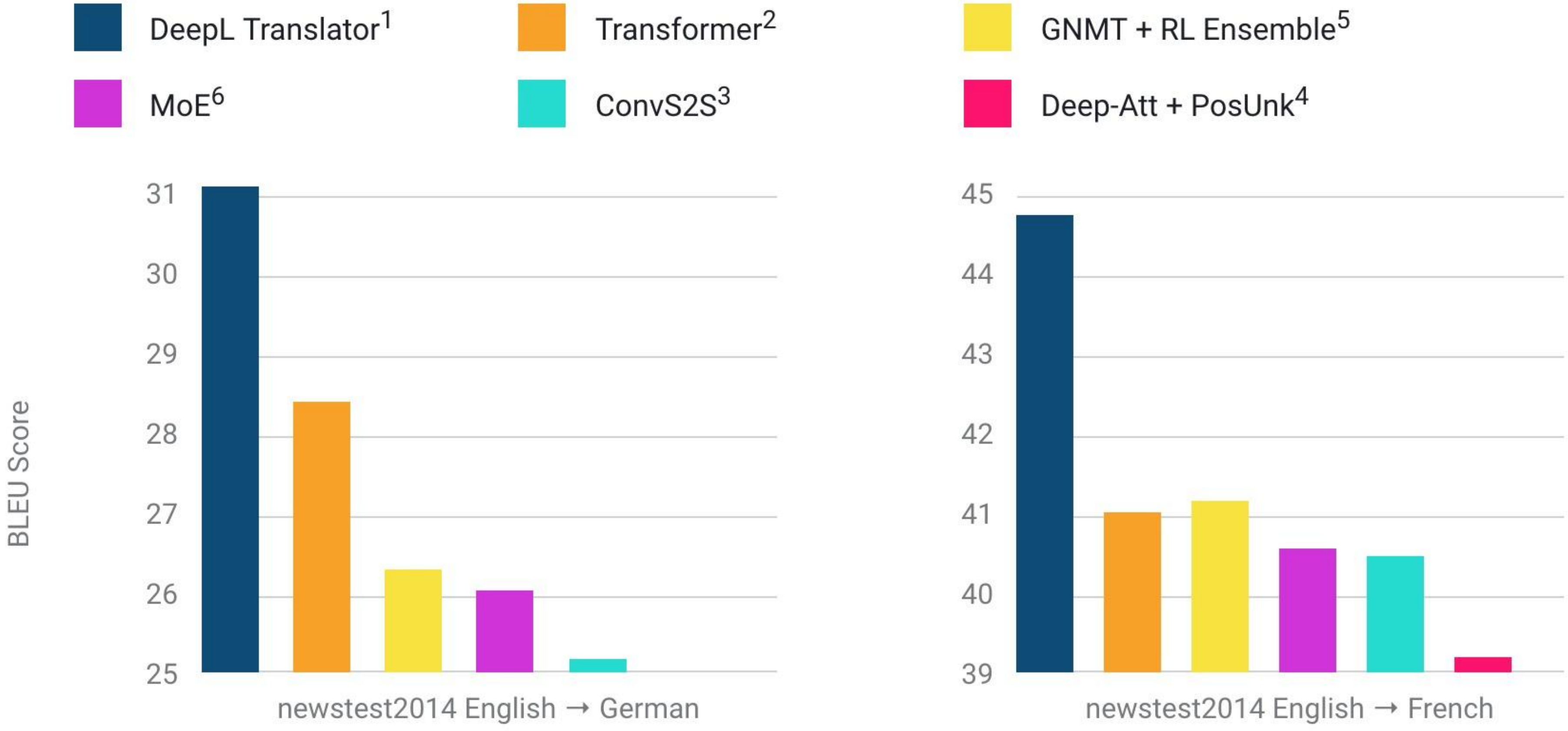
MT progress over time

[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

Data data data



Source: DeepL's press release (Aug 2017)

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**

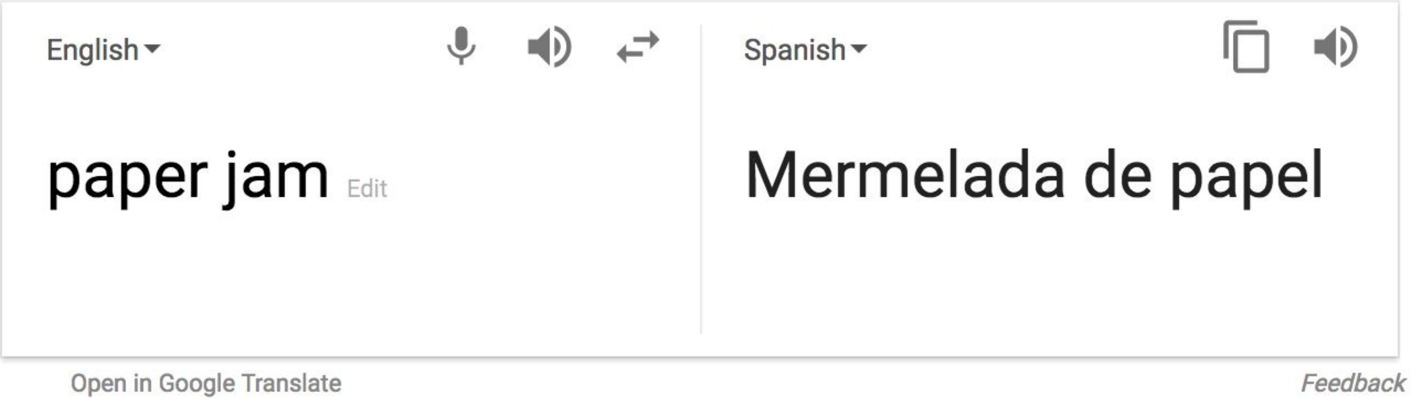
- **2014**: First seq2seq paper published
- **2016**: Google Translate switches from SMT to NMT
- This is amazing!
  - **SMT** systems, built by hundreds of engineers over many years, outperformed by NMT systems trained by a handful of engineers in a few months

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



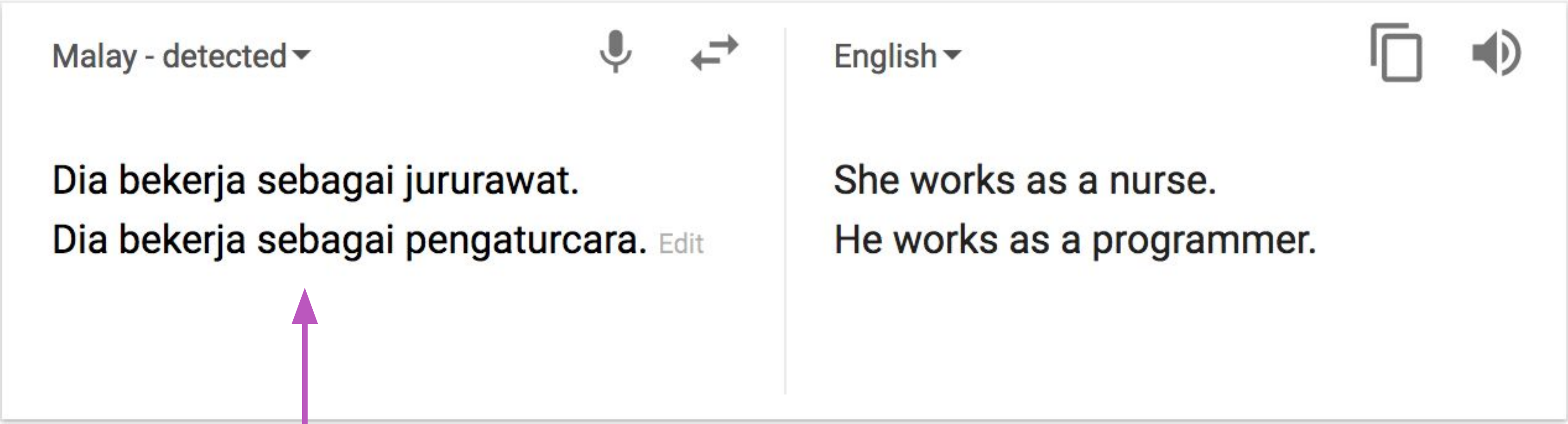
paper jam

Mermelada de papel

?

So is Machine Translation solved?

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



Dia bekerja sebagai jururawat.
Dia bekerja sebagai pengaturcara.

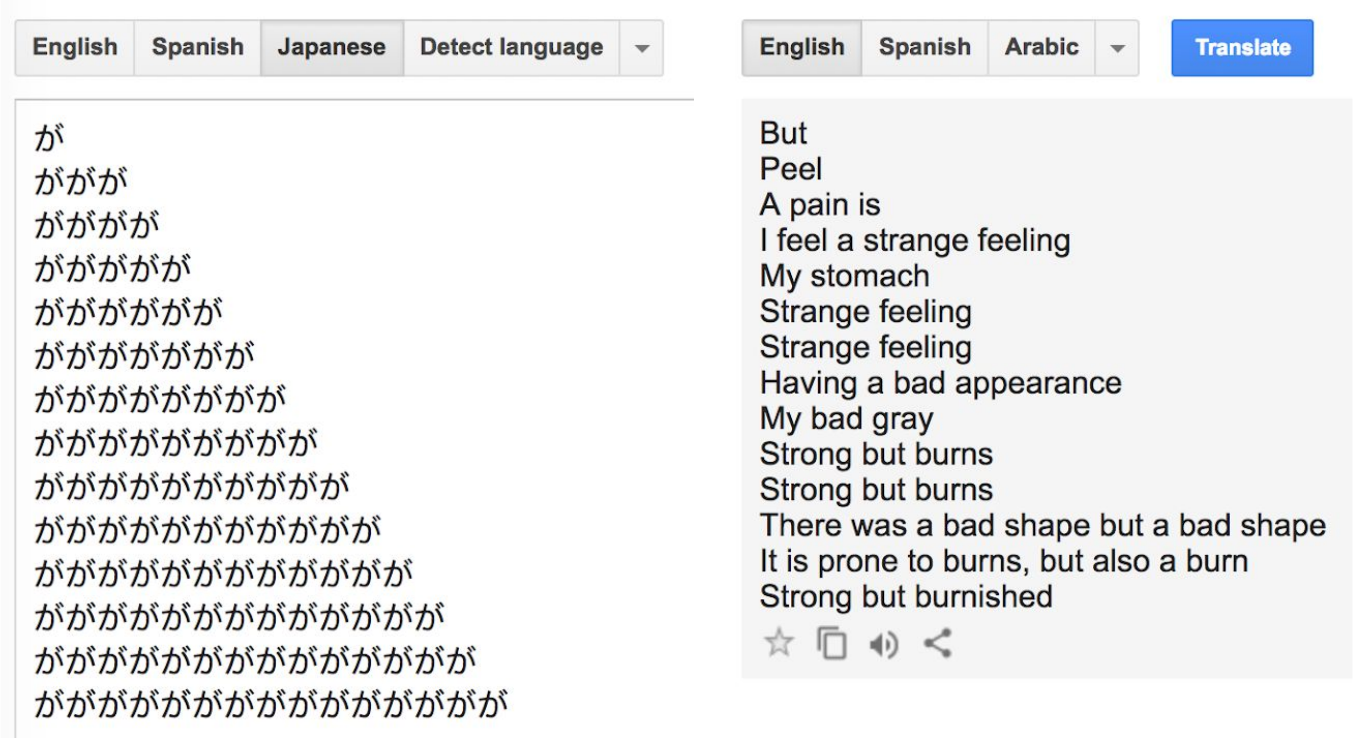
She works as a nurse.
He works as a programmer.

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



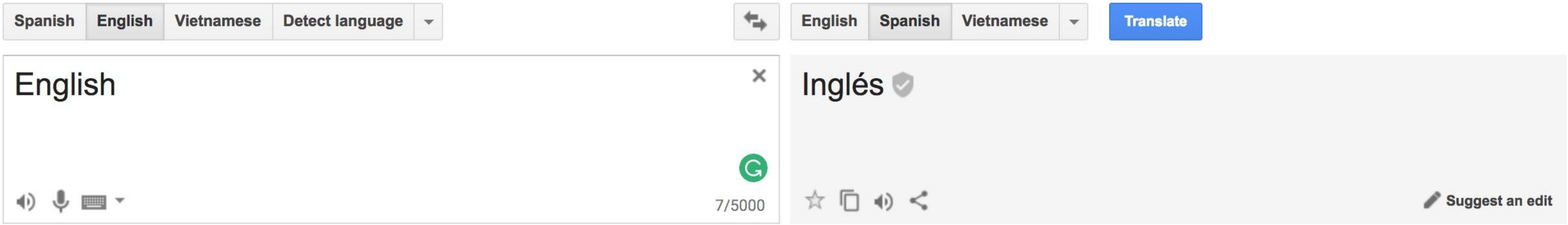
が
ががが
がががが
ががががが
がががががが
ががががががが
がががががががが
ががががががががが
がががががががががが
ががががががががががが
がががががががががががが
ががががががががががががが
がががががががががががががが
ががががががががががががががが

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
Strong but burnished

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

Google Translate vs DeepL (2/23/2018)

Google Translate: 0

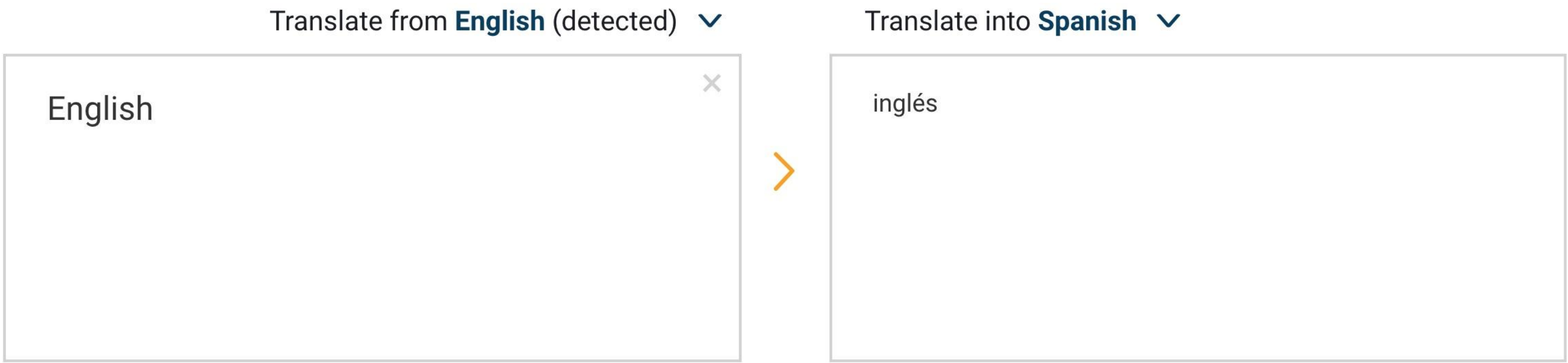


English

7/5000

Inglés

DeepL: 0



Translate from **English** (detected)

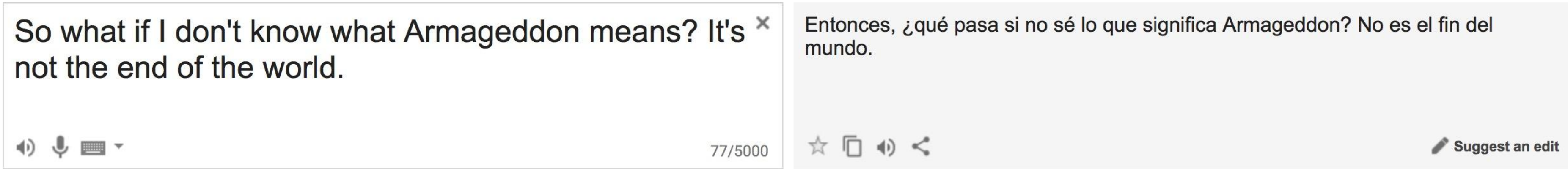
Translate into **Spanish**

English

inglés

Google Translate vs DeepL (2/23/2018)

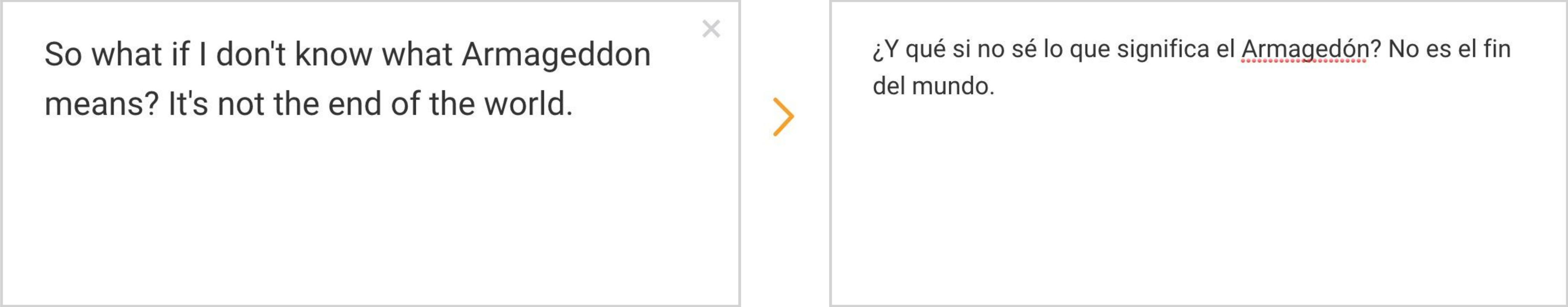
Google Translate: 0



So what if I don't know what Armageddon means? It's not the end of the world.

Entonces, ¿qué pasa si no sé lo que significa Armageddon? No es el fin del mundo.

DeepL: 0

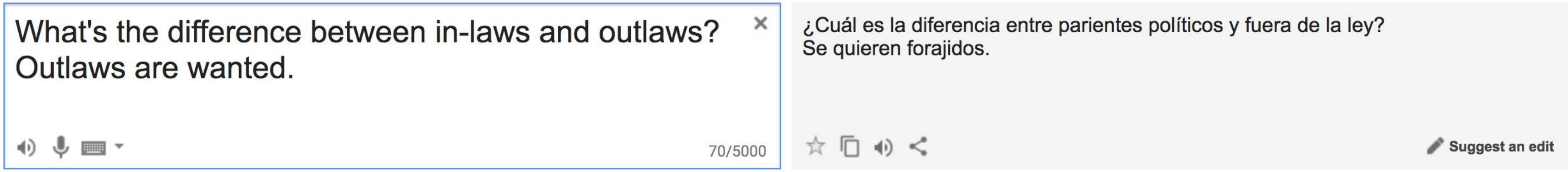


So what if I don't know what Armageddon means? It's not the end of the world.

¿Y qué si no sé lo que significa el Armagedón? No es el fin del mundo.

Google Translate vs DeepL (2/23/2018)

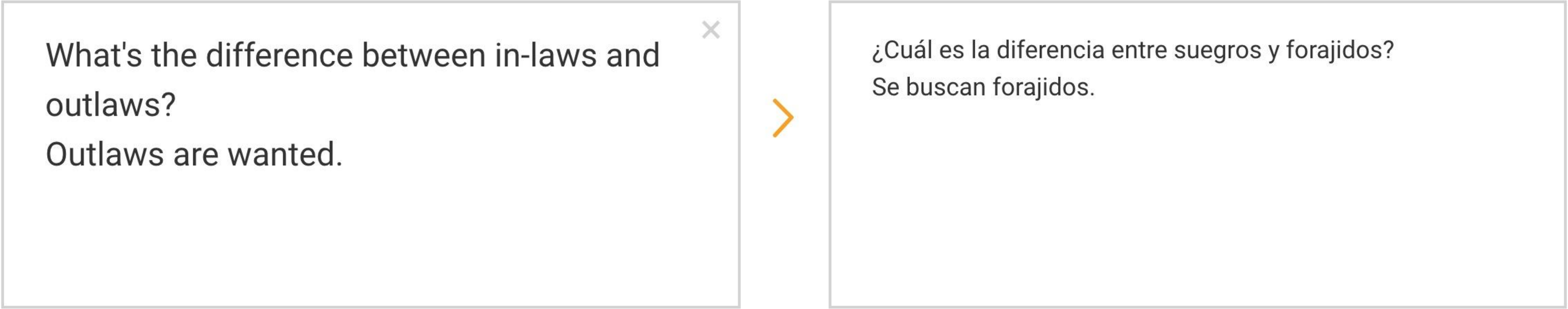
Google Translate: 0



What's the difference between in-laws and outlaws?
Outlaws are wanted.

¿Cuál es la diferencia entre parientes políticos y fuera de la ley?
Se quieren forajidos.

DeepL: 0

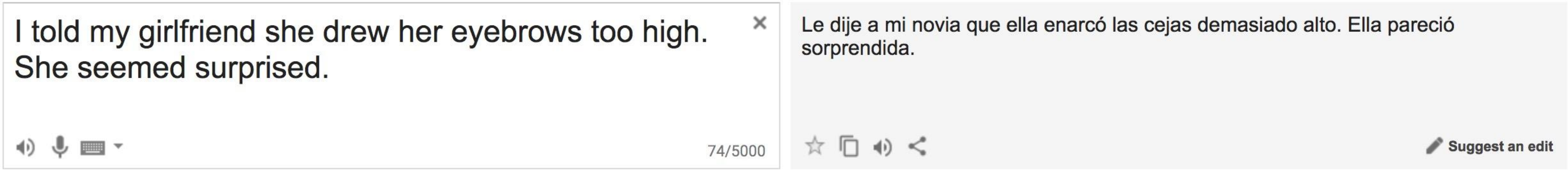


What's the difference between in-laws and outlaws?
Outlaws are wanted.

¿Cuál es la diferencia entre suegros y forajidos?
Se buscan forajidos.

Google Translate vs DeepL (2/23/2018)

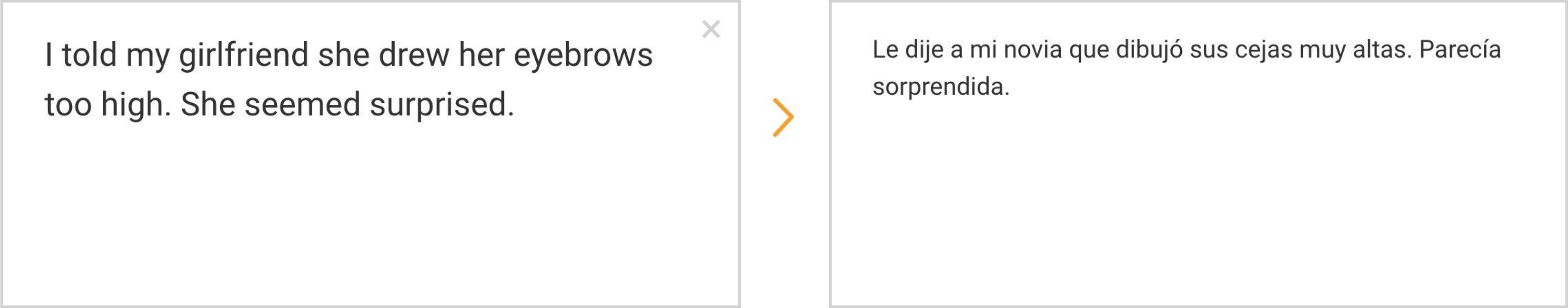
Google Translate: 0



I told my girlfriend she drew her eyebrows too high. She seemed surprised.

Le dije a mi novia que ella enarcó las cejas demasiado alto. Ella pareció sorprendida.

DeepL: 0

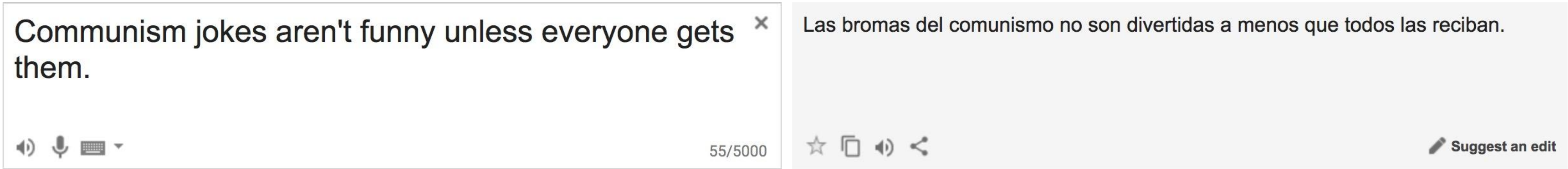


I told my girlfriend she drew her eyebrows too high. She seemed surprised.

Le dije a mi novia que dibujó sus cejas muy altas. Parecía sorprendida.

Google Translate vs DeepL (2/23/2018)

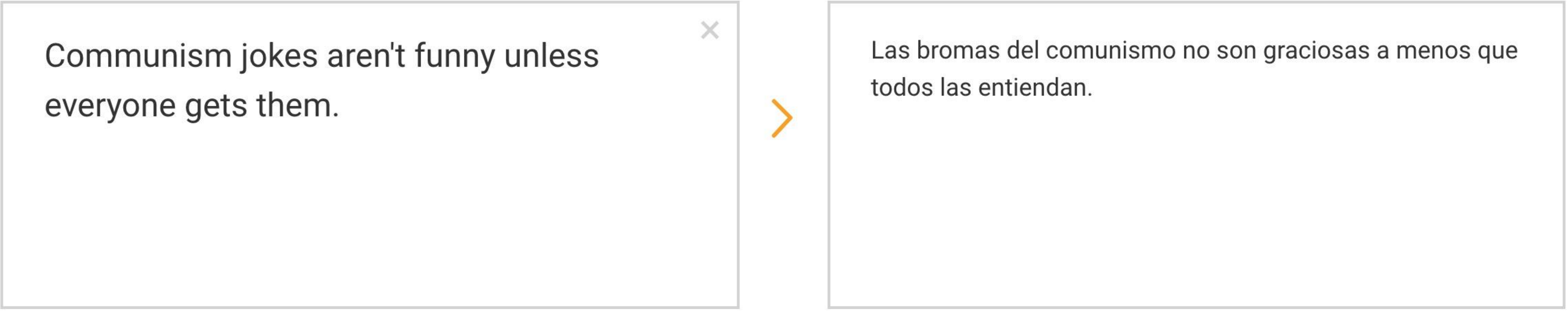
Google Translate: 0



Communism jokes aren't funny unless everyone gets them.

Las bromas del comunismo no son divertidas a menos que todos las reciban.

DeepL: 0

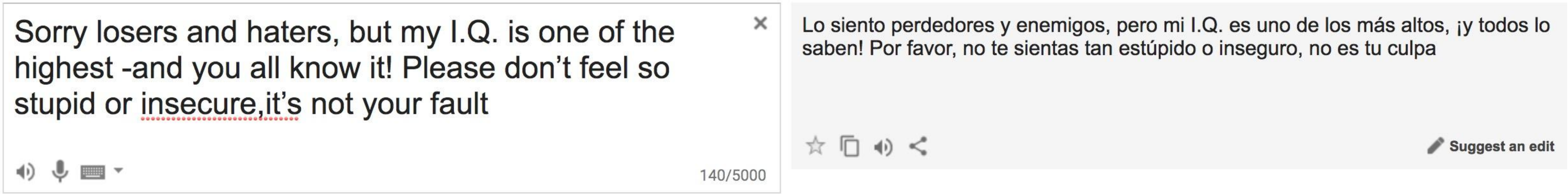


Communism jokes aren't funny unless everyone gets them.

Las bromas del comunismo no son graciosas a menos que todos las entiendan.

Google Translate vs DeepL (2/23/2018)

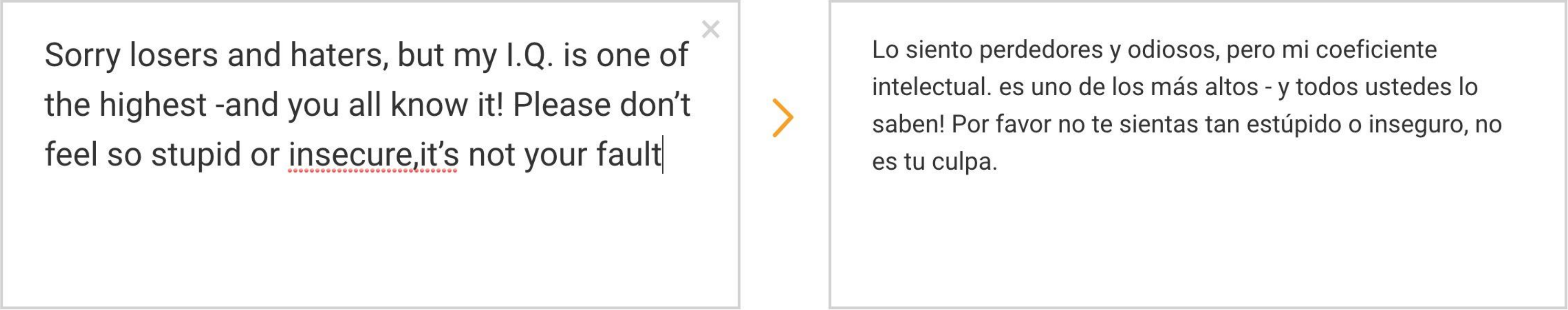
Google Translate: 0



Sorry losers and haters, but my I.Q. is one of the highest -and you all know it! Please don't feel so stupid or insecure,it's not your fault

Lo siento perdedores y enemigos, pero mi I.Q. es uno de los más altos, ¡y todos lo saben! Por favor, no te sientas tan estúpido o inseguro, no es tu culpa

DeepL: 0



Sorry losers and haters, but my I.Q. is one of the highest -and you all know it! Please don't feel so stupid or insecure,it's not your fault

Lo siento perdedores y odiosos, pero mi coeficiente intelectual. es uno de los más altos - y todos ustedes lo saben! Por favor no te sientas tan estúpido o inseguro, no es tu culpa.

NMT research continues

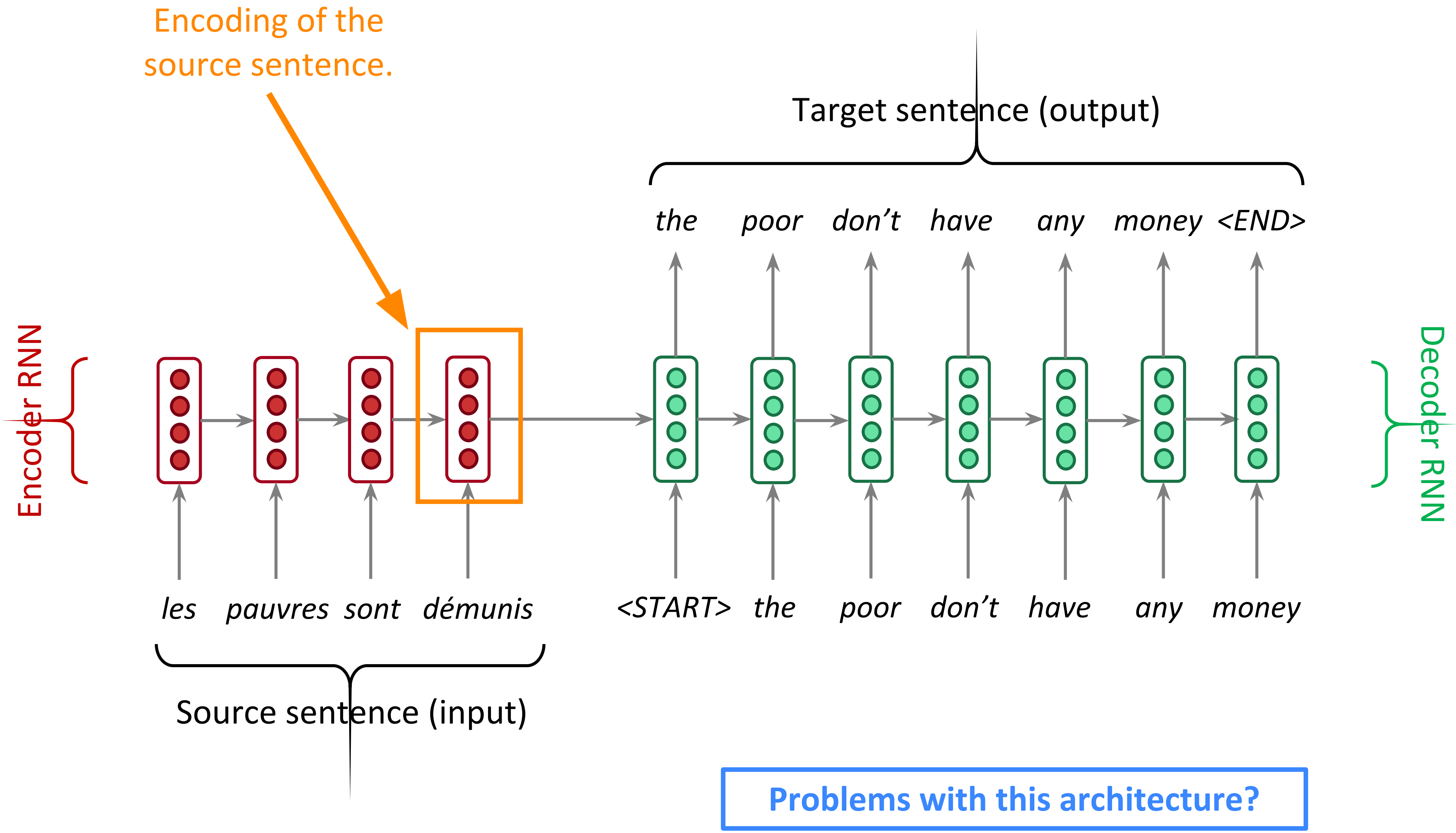
NMT is the **flagship task** for NLP Deep Learning

- 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…

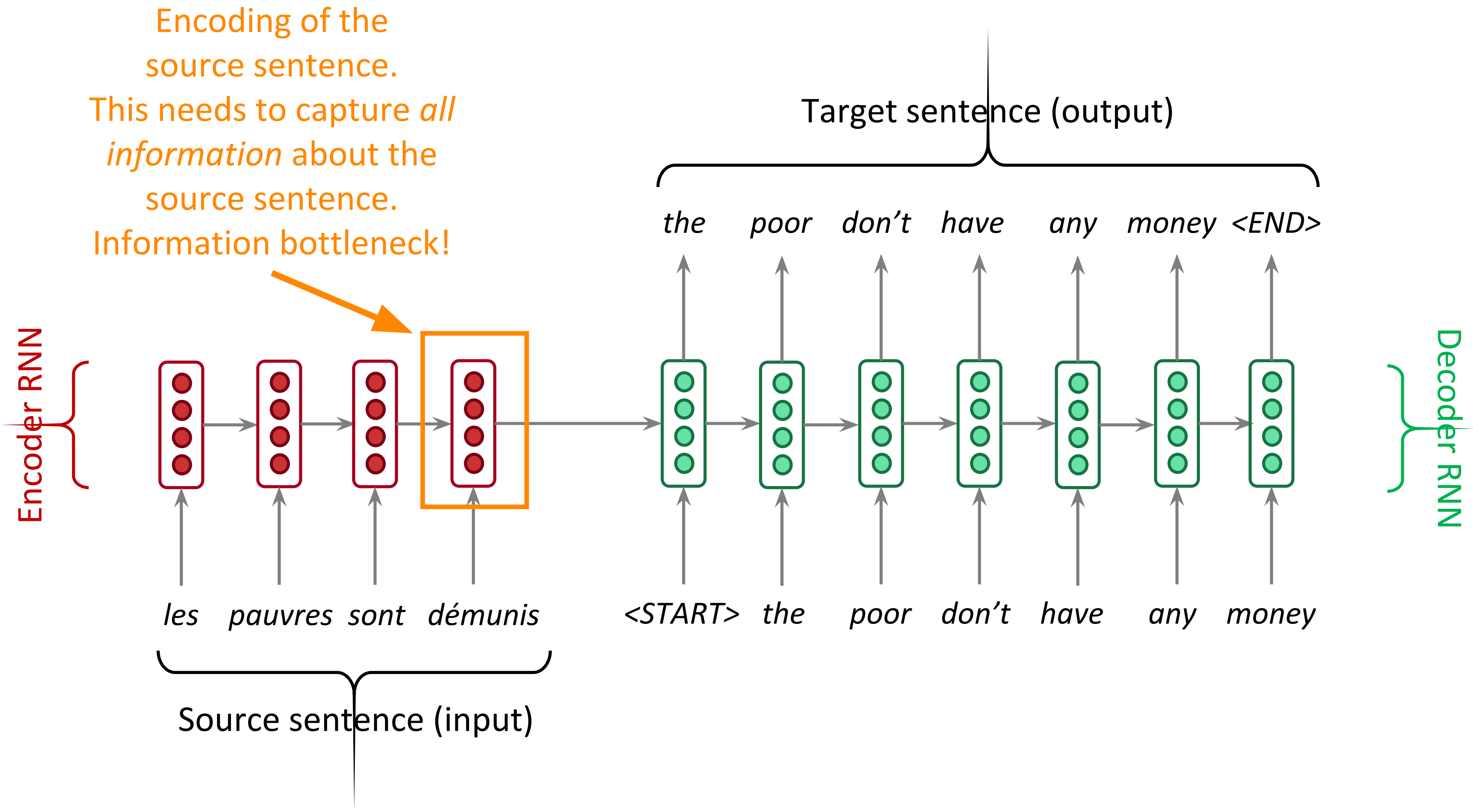
ATTENTION

Sequence-to-sequence: the bottleneck problem


Encoding of the source sentence.
Target sentence (output)
the poor don’t have any money <END>
Encoder RNN
Decoder RNN
les pauvres sont démunis
<START> the poor don’t have any money
Source sentence (input)

Problems with this architecture?

Sequence-to-sequence: the bottleneck problem

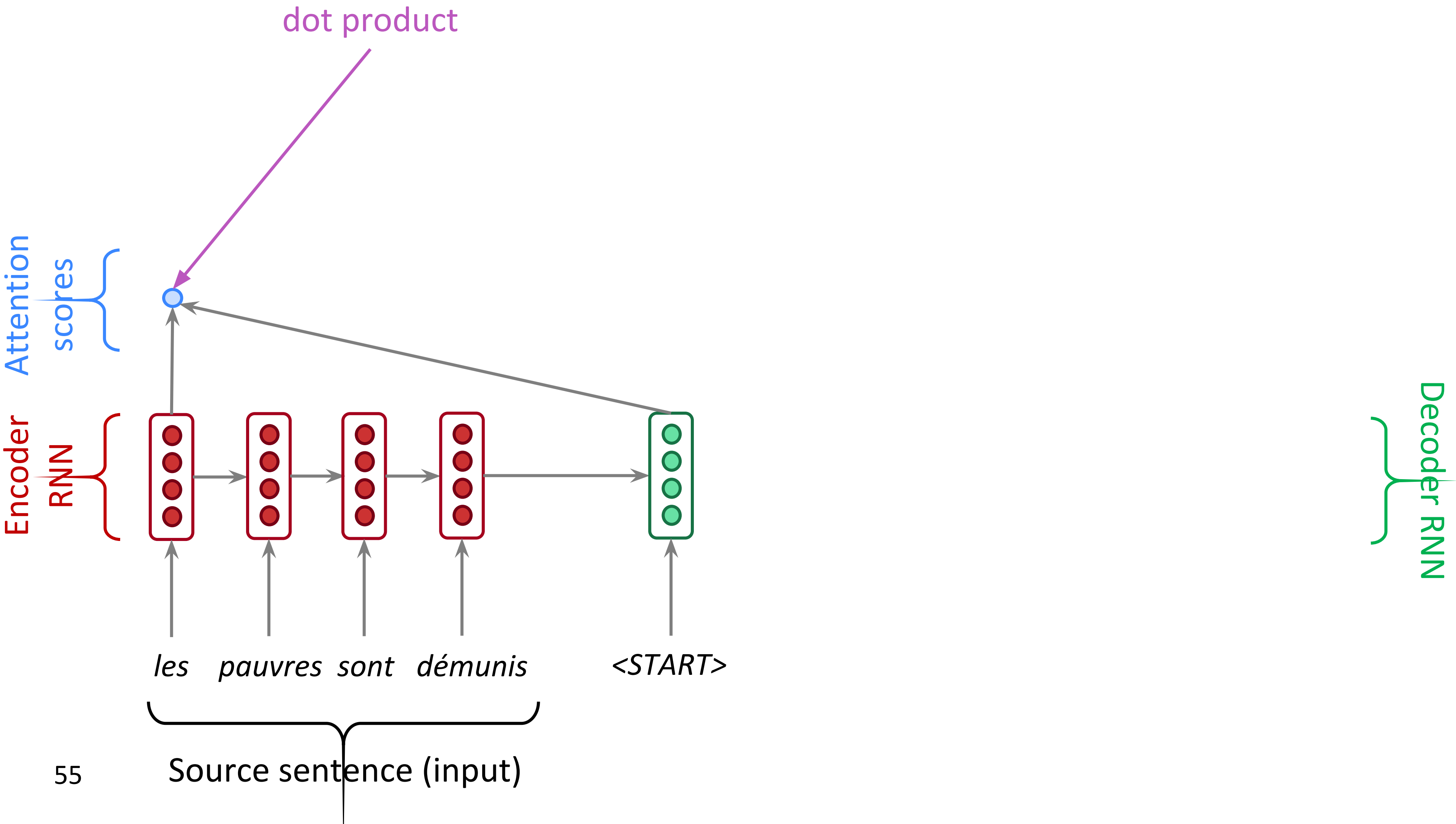

Encoding of the source sentence.
This needs to capture *all information* about the source sentence.
Information bottleneck!
Target sentence (output)
the *poor* *don't* *have* *any* *money* *<END>*
Encoder RNN
Decoder RNN
les *pauvres* *sont* *démunis*
<START> *the* *poor* *don't* *have* *any* *money*
Source sentence (input)

Attention

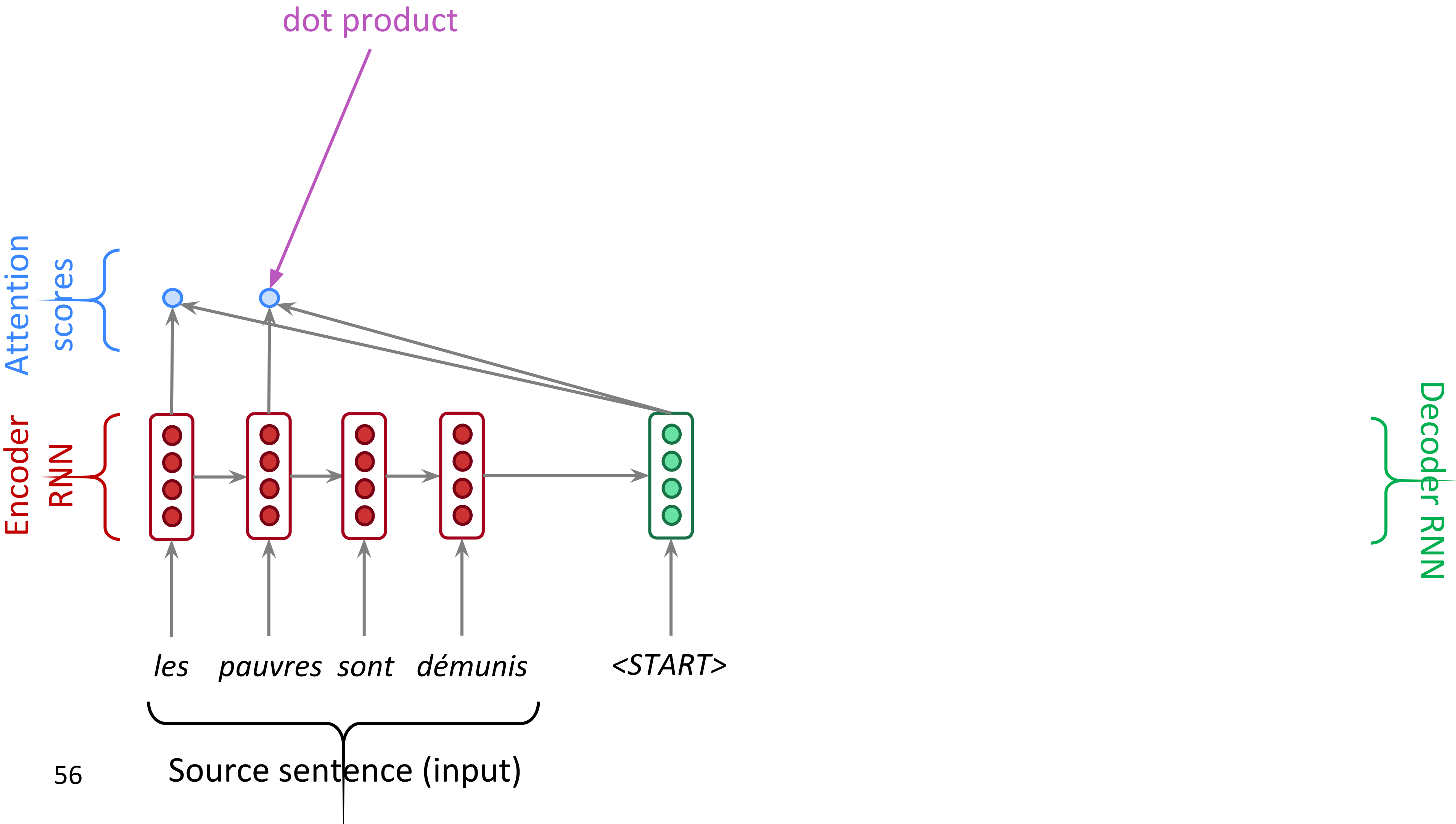
- **Attention** provides a solution to the bottleneck problem.

- Core idea: on each step of the decoder, *focus on a particular part* of the source sequence

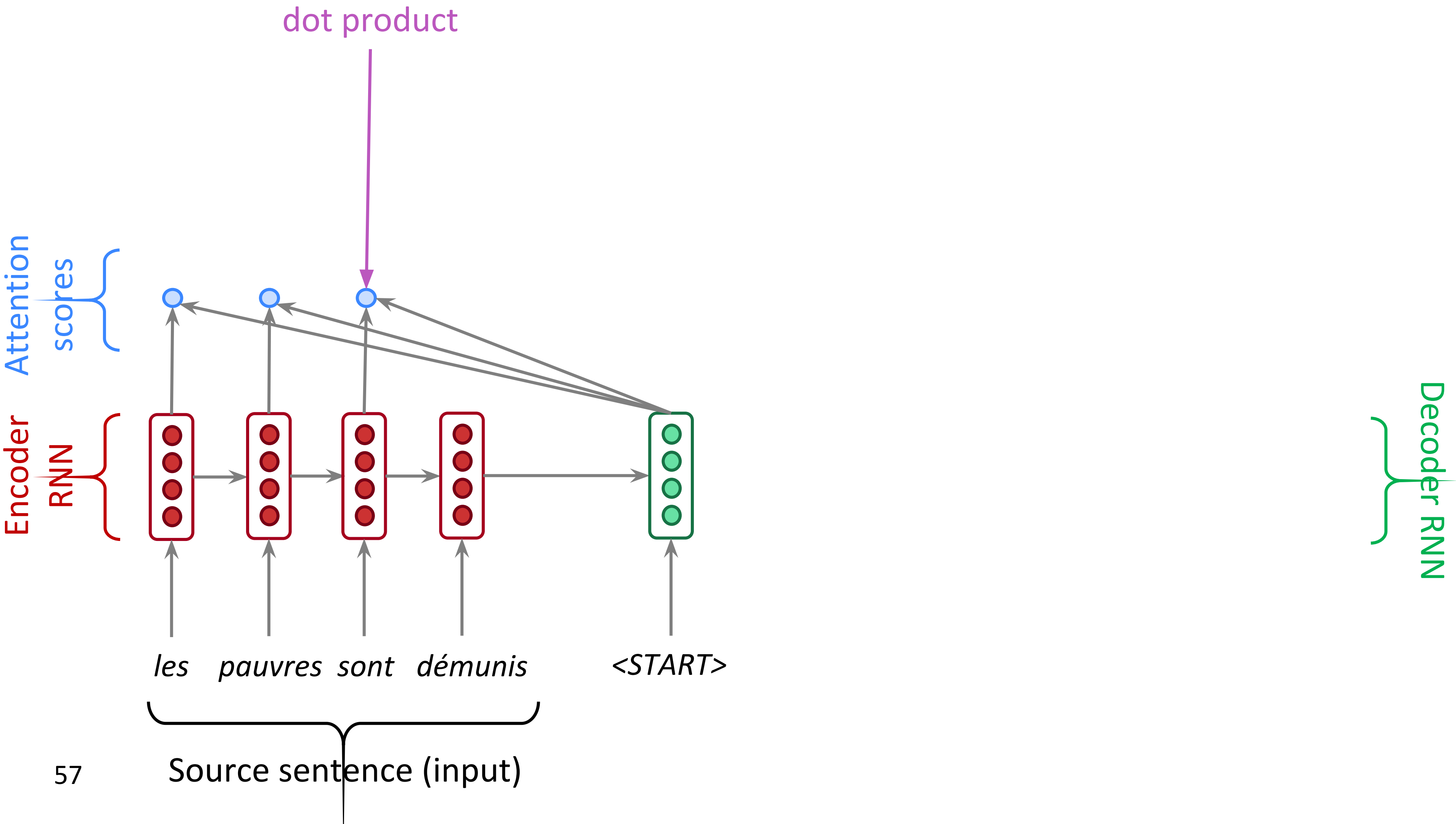
Sequence-to-sequence with attention



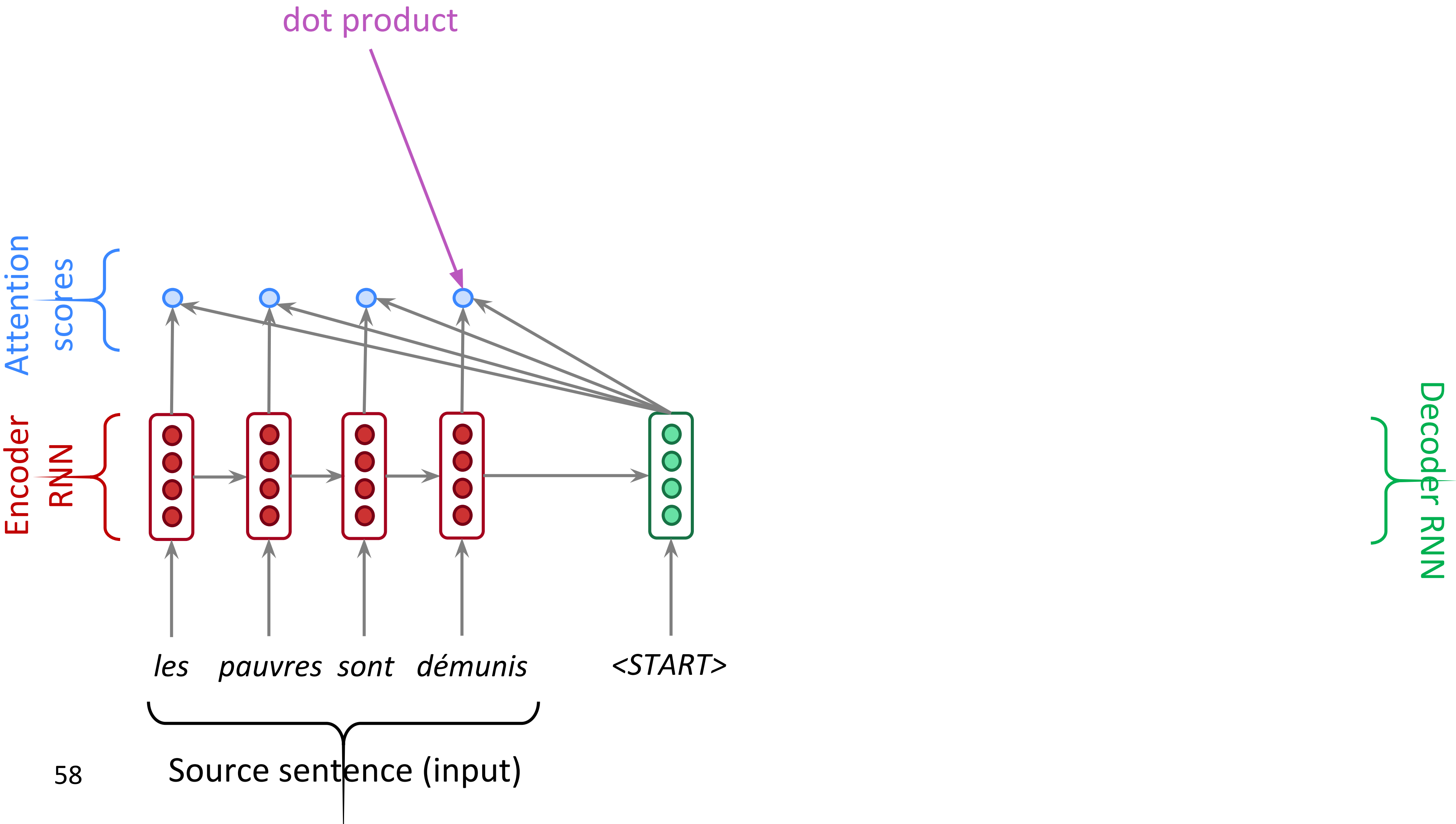
Sequence-to-sequence with attention



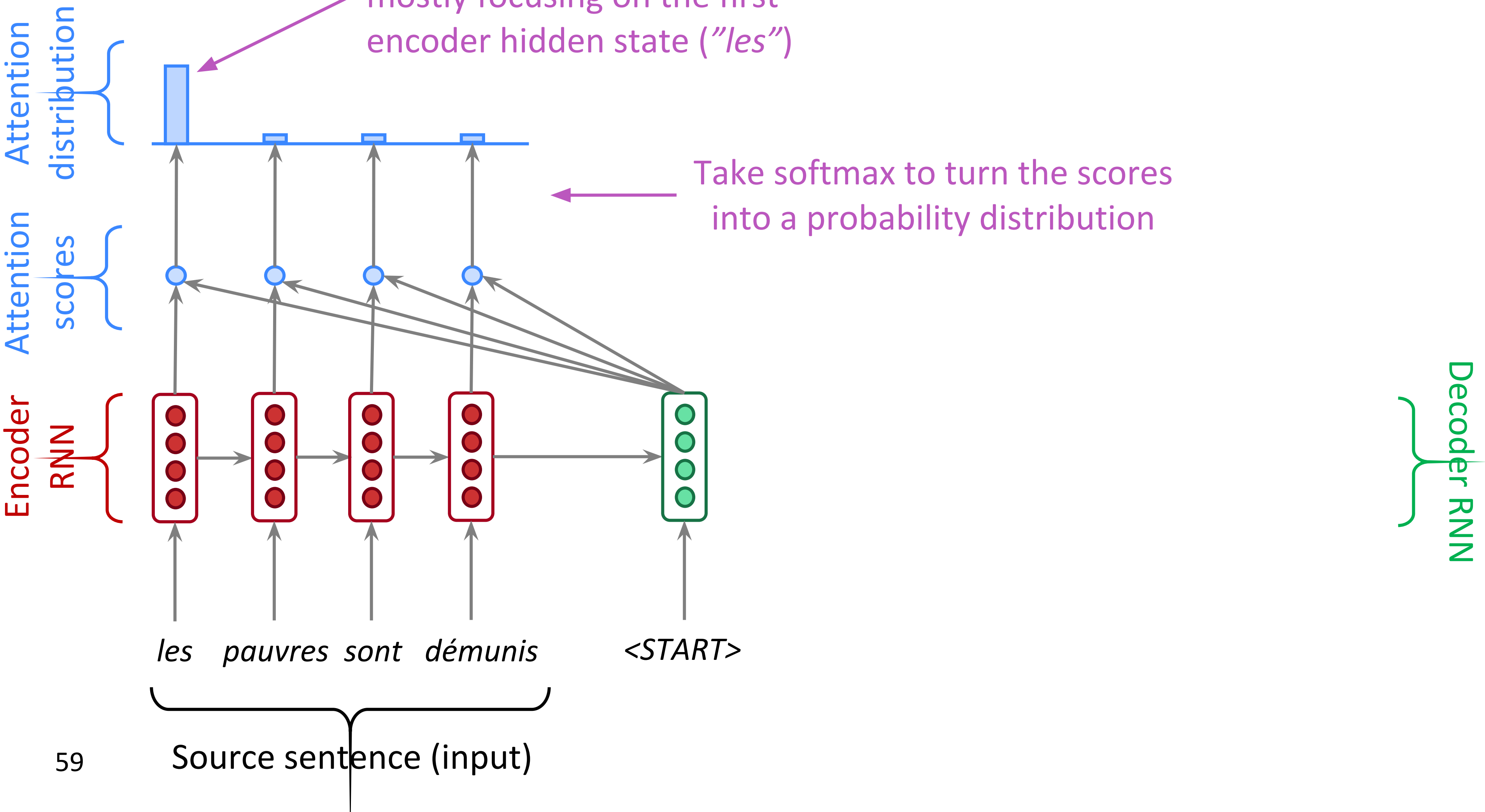
Sequence-to-sequence with attention



Sequence-to-sequence with attention



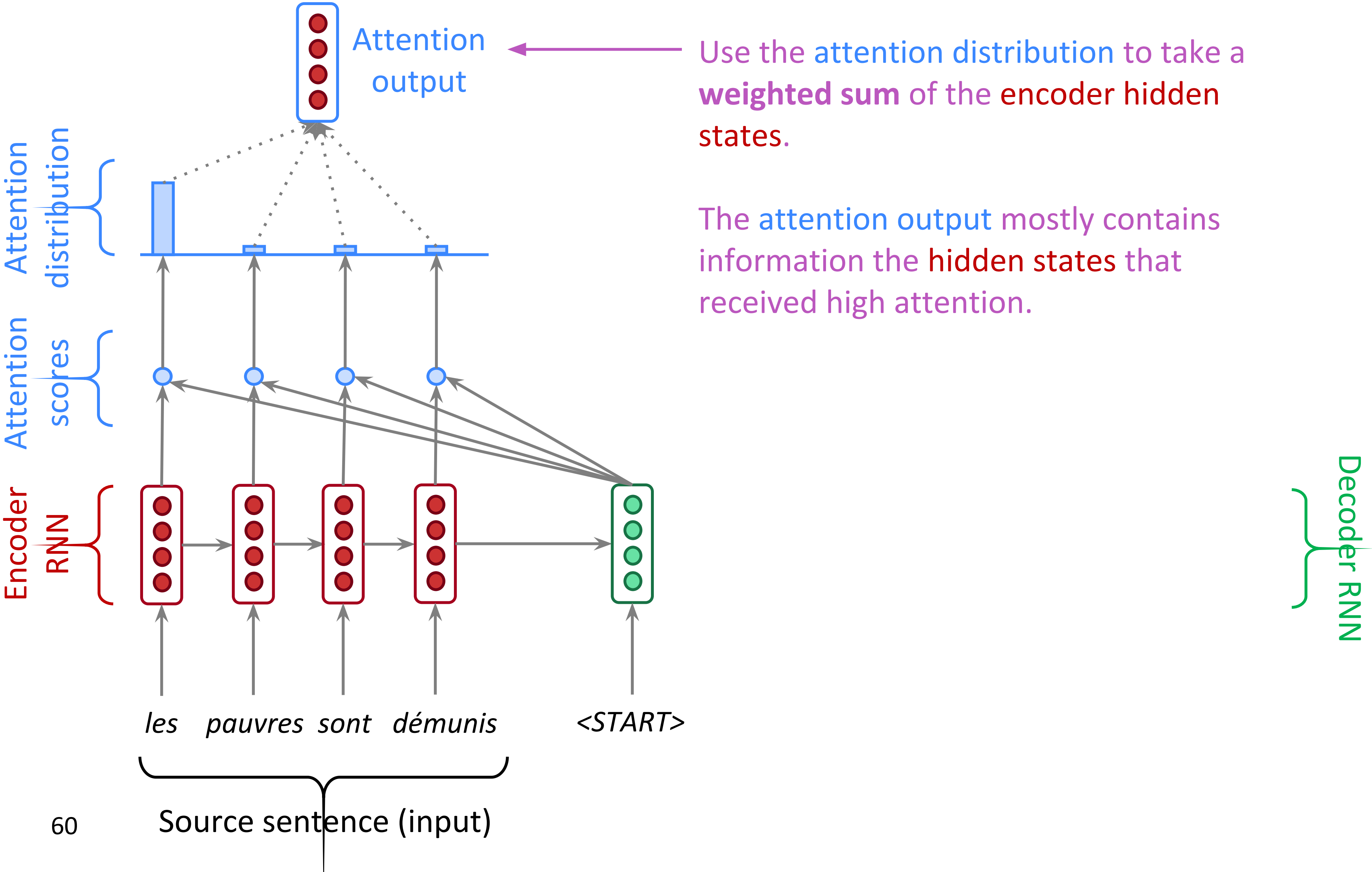
Sequence-to-sequence with attention



On this decoder timestep, we're mostly focusing on the first encoder hidden state (*"les"*)

Take softmax to turn the scores into a probability distribution

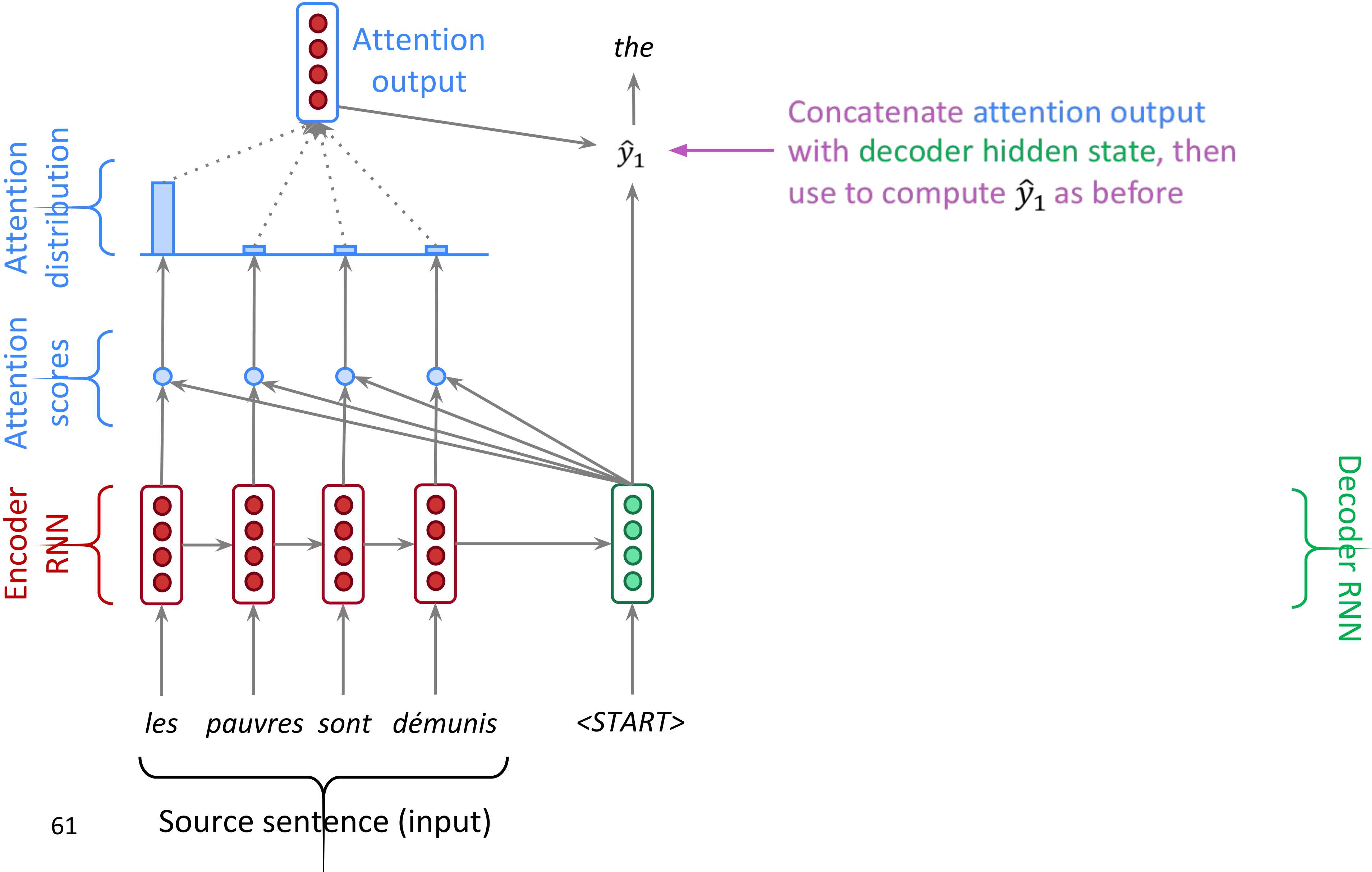
Sequence-to-sequence with attention



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

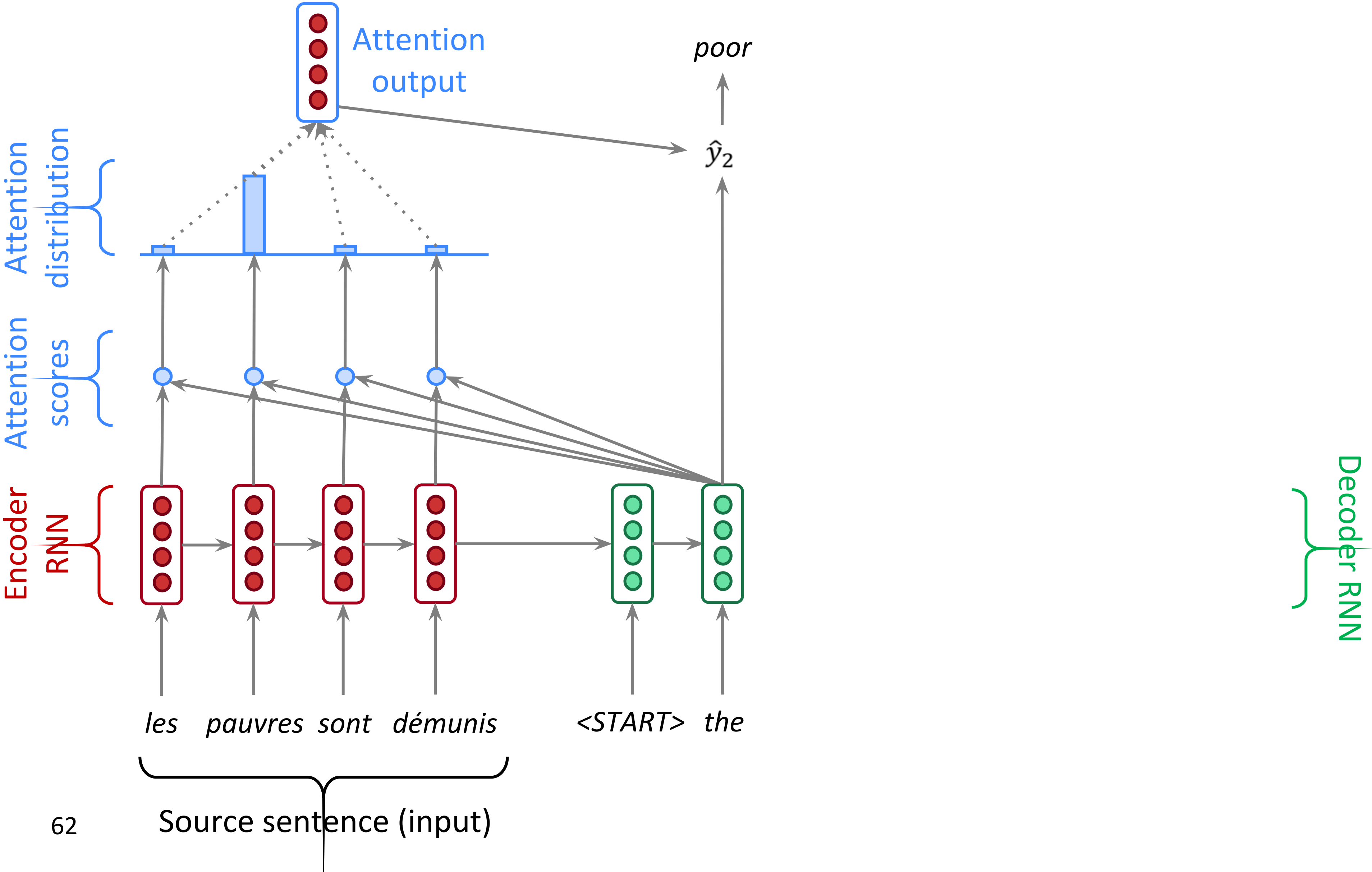
The attention output mostly contains information the hidden states that received high attention.

Sequence-to-sequence with attention

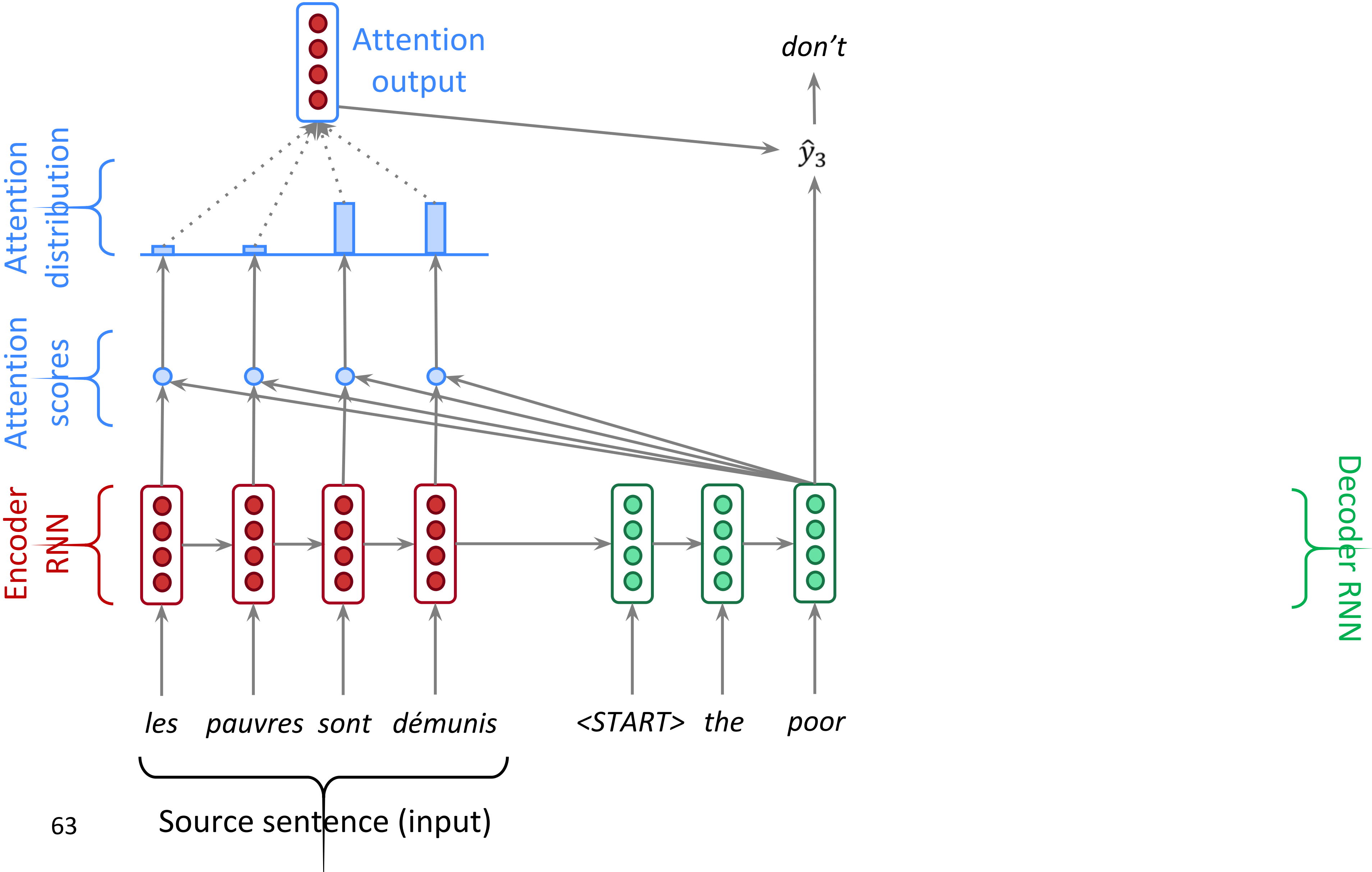


Concatenate attention output with decoder hidden state, then use to compute $\hat{y}_1$ as before

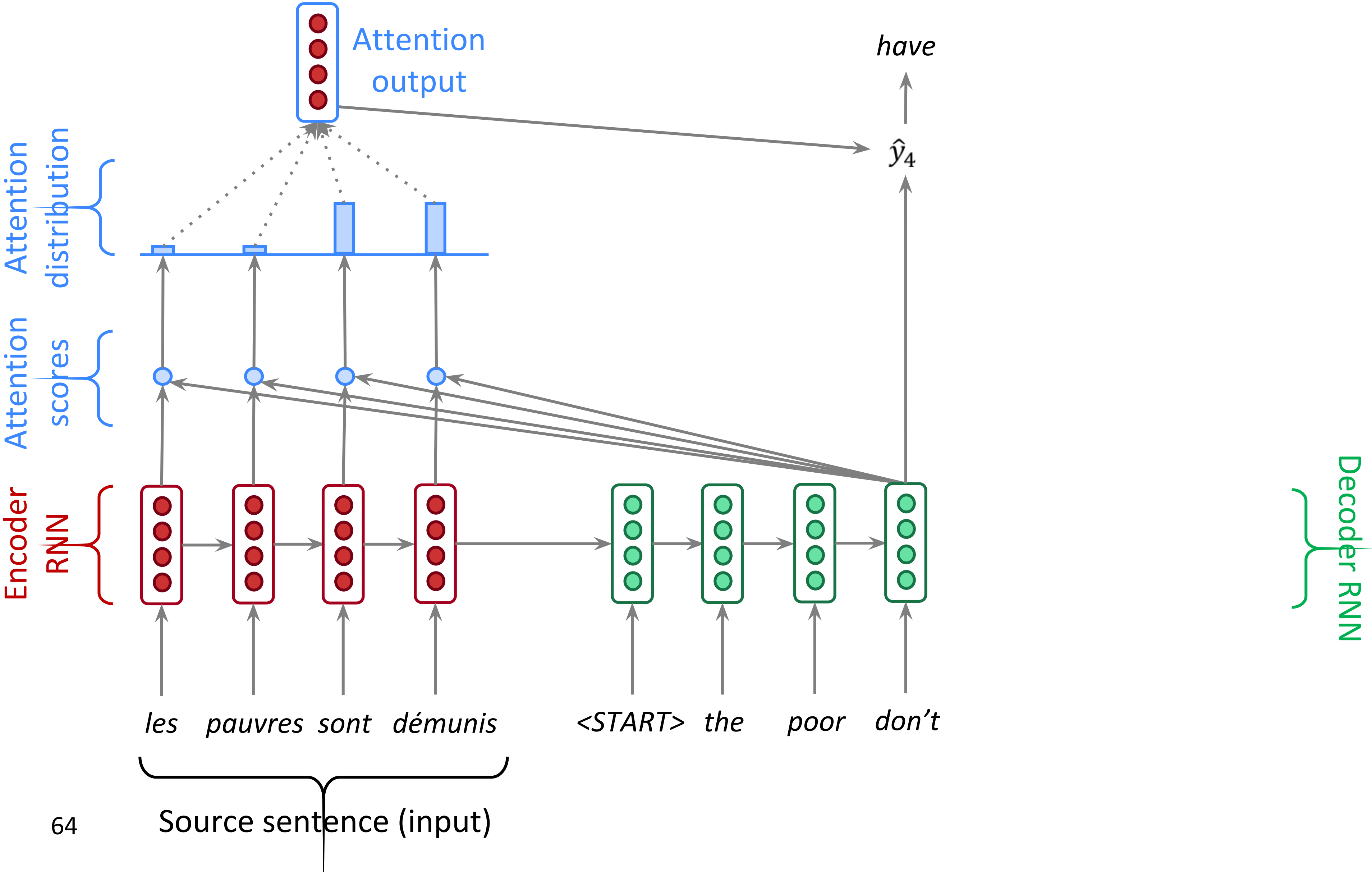
Sequence-to-sequence with attention



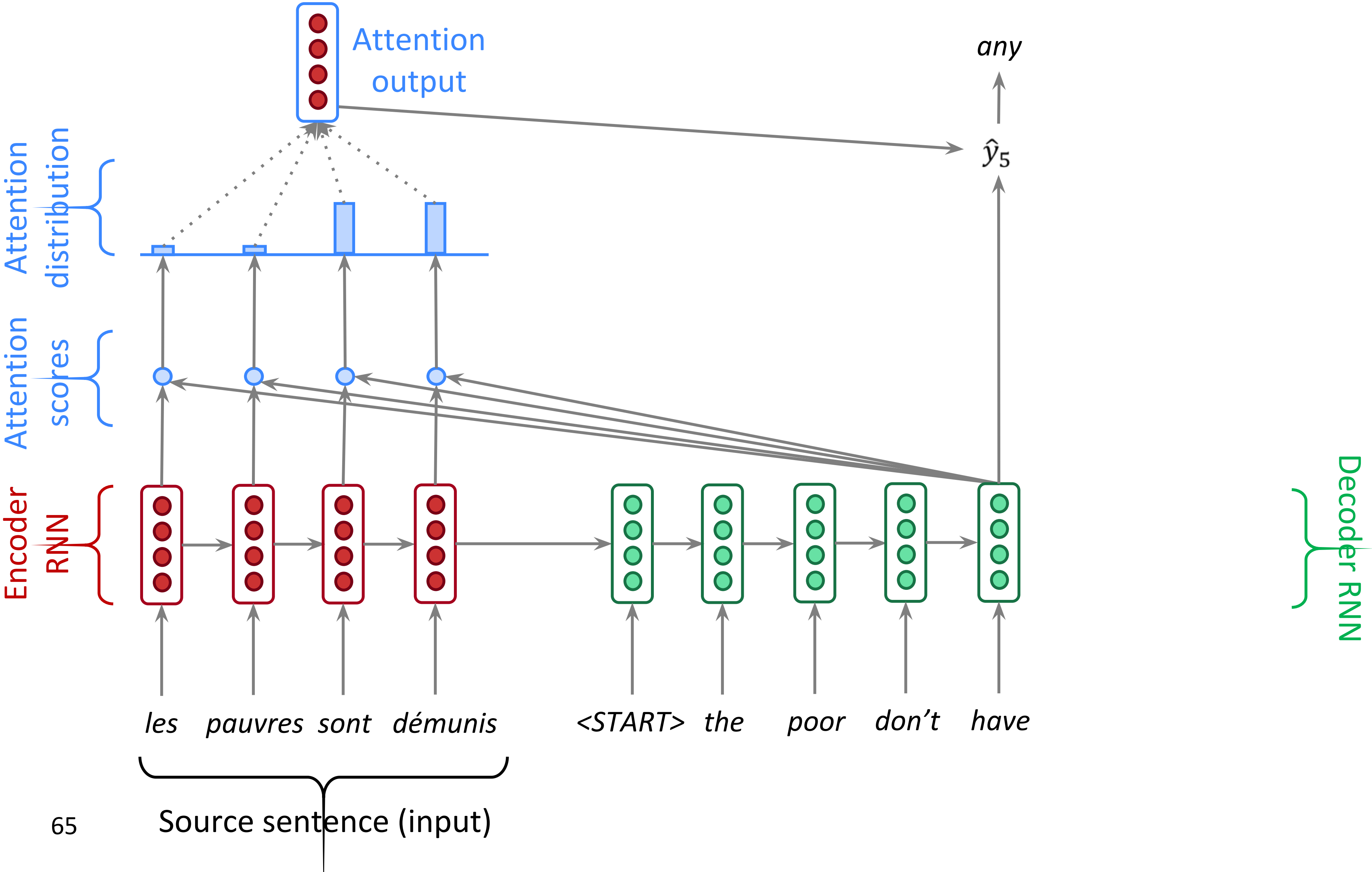
Sequence-to-sequence with attention



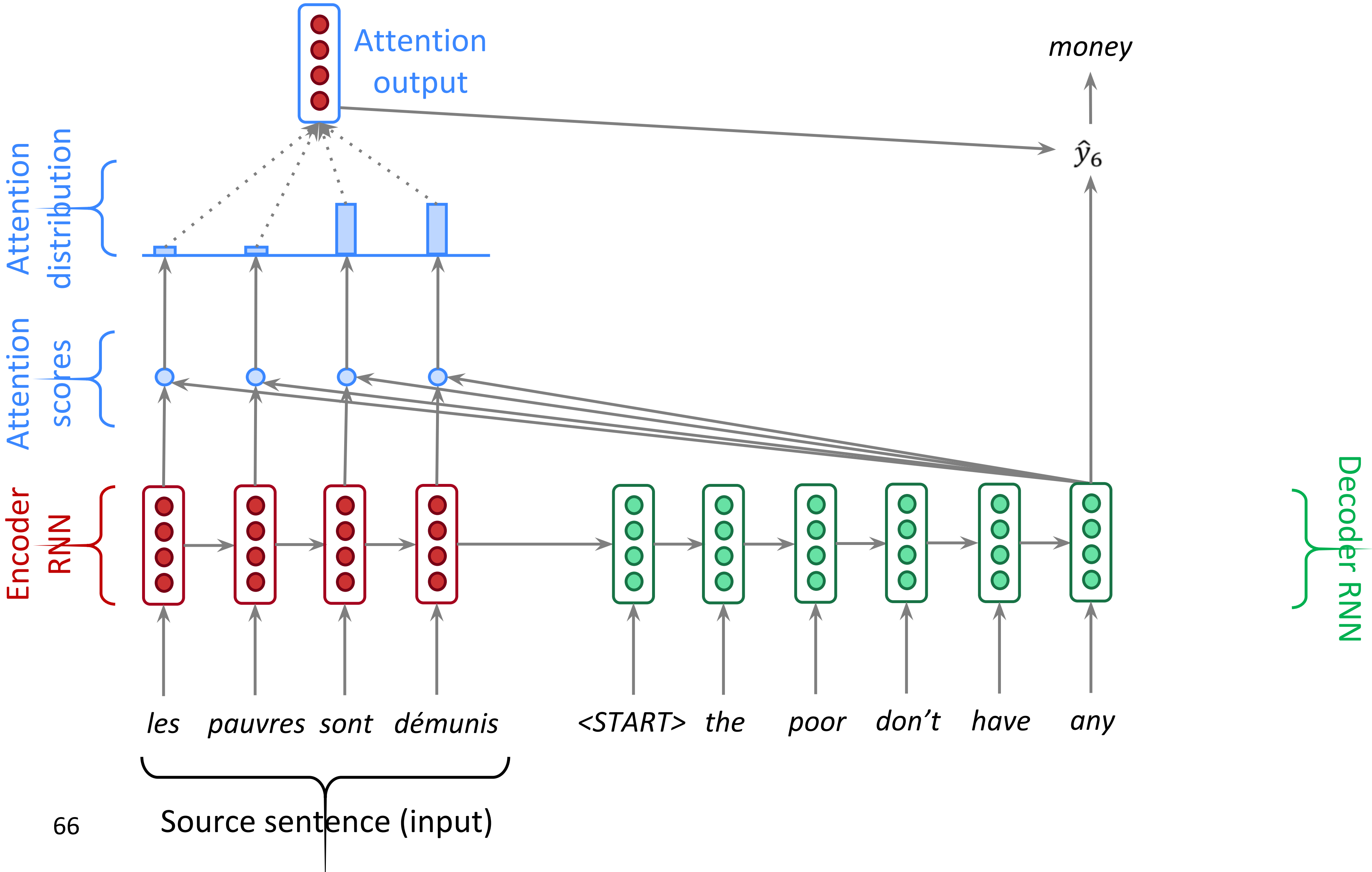
Sequence-to-sequence with attention



Sequence-to-sequence with attention



Sequence-to-sequence with attention



Attention: in equations

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

$$\boldsymbol{e}^t = [\boldsymbol{s}_t^T \boldsymbol{h}_1, \ldots, \boldsymbol{s}_t^T \boldsymbol{h}_N] \in \mathbb{R}^N$$

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

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

- We use α^t to take a weighted sum of the encoder hidden states to get the attention output $\boldsymbol{a}_t$

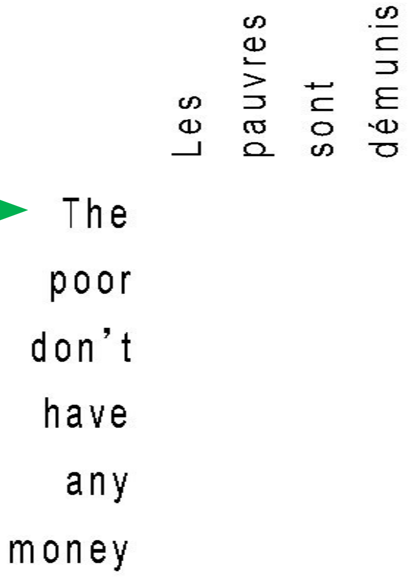
$$\boldsymbol{a}_t = \sum_{i=1}^{N} \alpha_i^t \boldsymbol{h}_i \in \mathbb{R}^h$$

- Finally we concatenate the attention output $\boldsymbol{a}_t$ with the decoder hidden state s_t and proceed as in the non-attention seq2seq model

$$[\boldsymbol{a}_t; \boldsymbol{s}_t] \in \mathbb{R}^{2h}$$

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 class

- Transformers (guest lecture by Lukasz Kaiser)