Bài 10
Mô hình seq2seq và ứng dụng trong sinh văn bản
(seq2seq and application for text generation)

Lê Thanh Hương
Trường Công nghệ Thông tin và Truyền thông, ĐHBKHN

Context

- Machine translation
- The seq2seq model
- Attention mechanism
- Machine translation with seq2seq
- Google’s Neural Machine Translation

Machine Translation

<u>**Machine Translation (MT)**</u> 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*

$\downarrow$

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

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*

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

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

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

Translation Model

Models how words and phrases should be translated (*fidelity*).
Learned from parallel data.

Language Model

Models how to write good English (*fluency*).
Learned from monolingual data.

1990s–2010s: Statistical Machine Translation

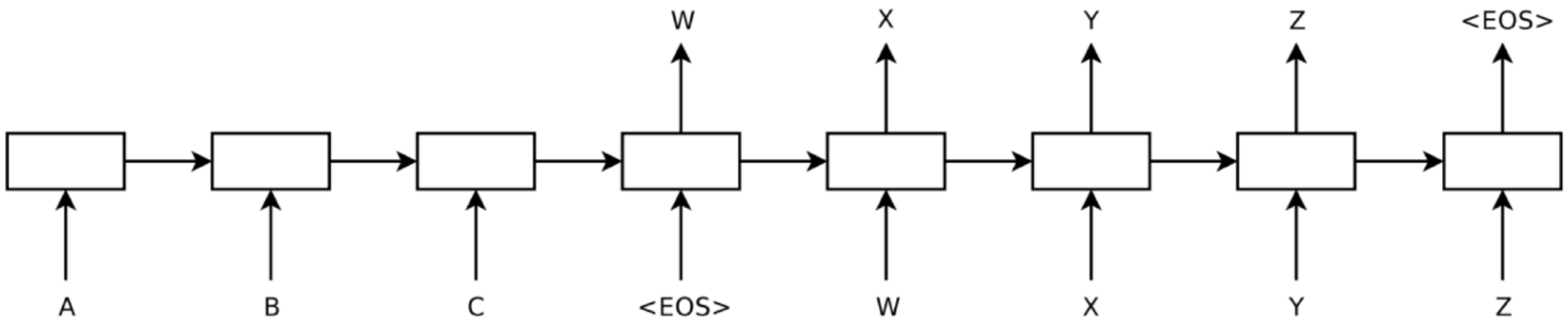
- SMT was a huge research field
- The best systems were extremely complex
  - Hundreds of important details
- Systems had many separately-designed subcomponents
  - Lots of feature engineering
    - Need to design features to capture particular language phenomena
  - Required compiling and maintaining extra resources
    - Like tables of equivalent phrases
  - Lots of human effort to maintain
    - Repeated effort for each language pair!

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.

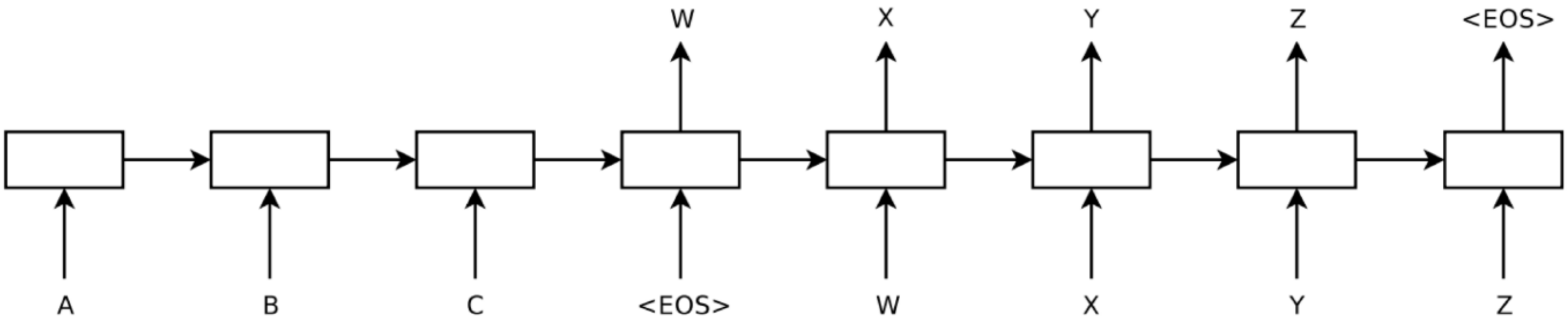
Encoder-decoder Framework



“Sequence to Sequence Learning with Neural Networks”, 2014

Encoder-decoder Framework

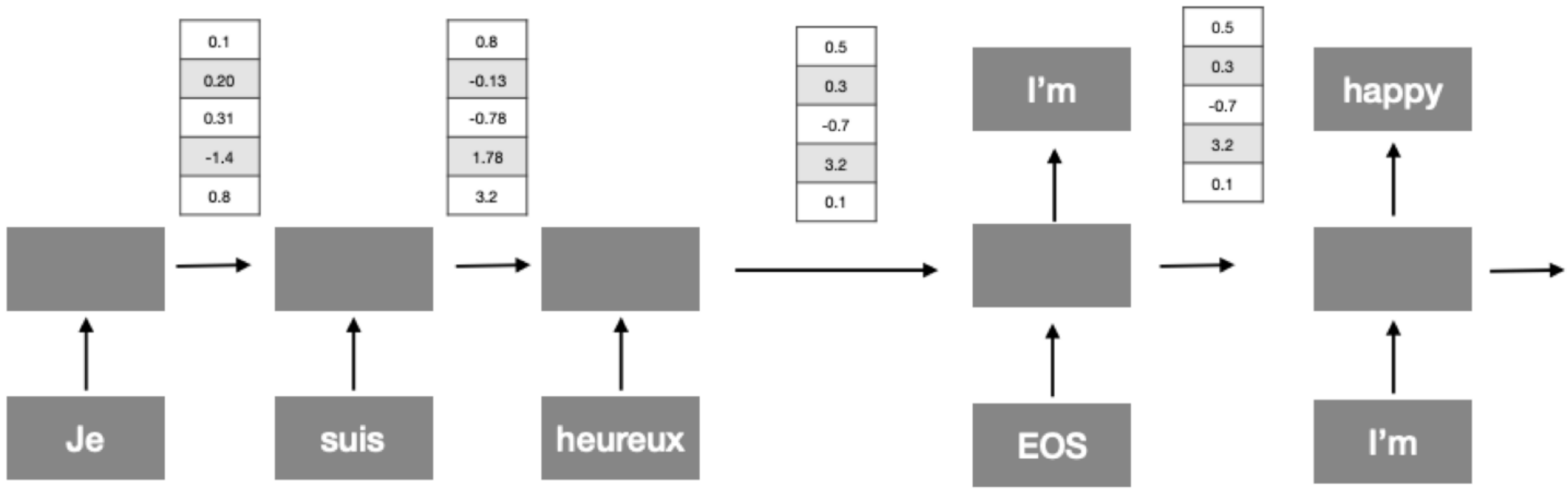
Vectơ K chiều của ngữ cảnh



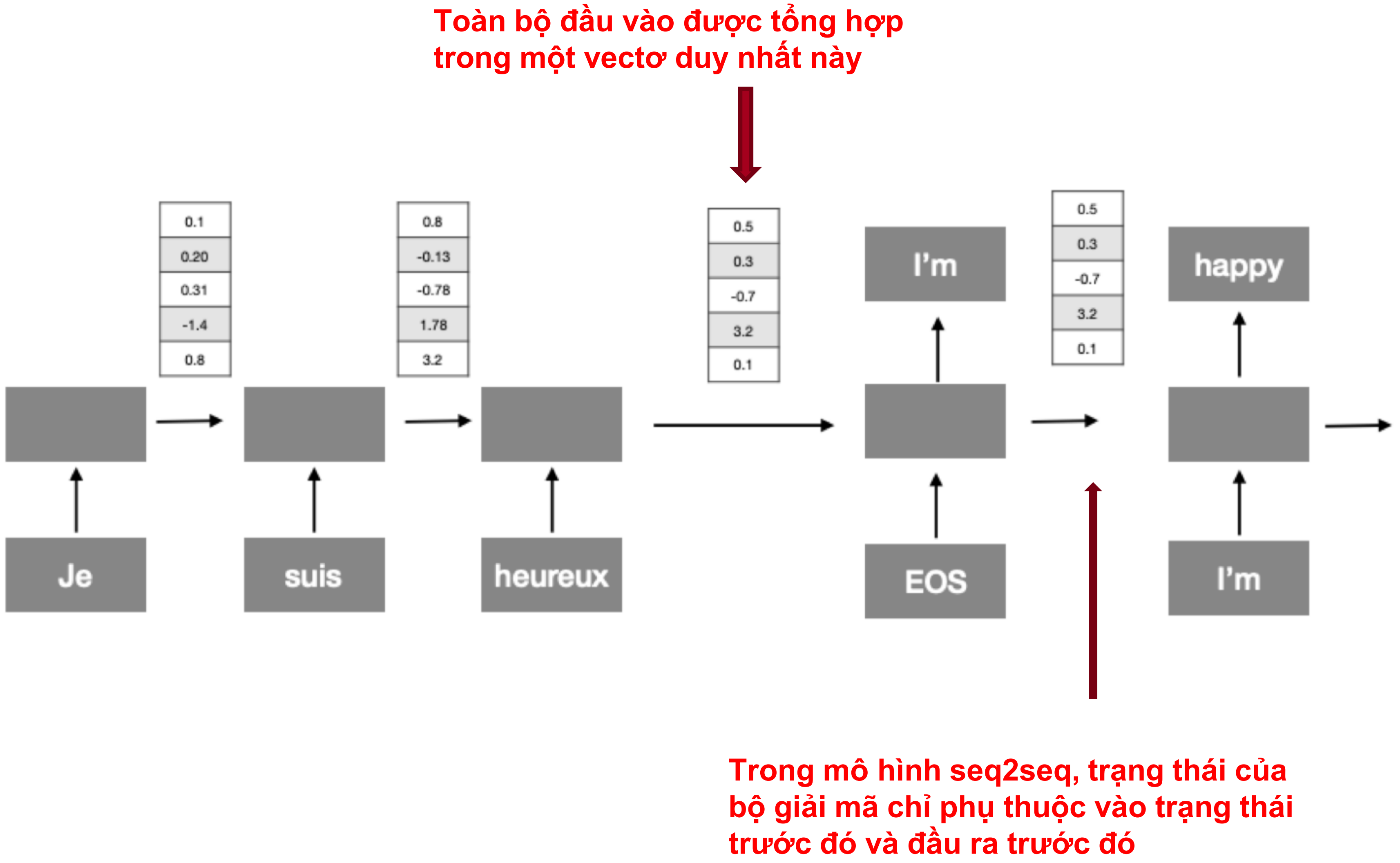
Điều kiện của từ được sinh ra trong bản dịch

"Sequence to Sequence Learning with Neural Networks", 2014

Encoder-decoder Framework

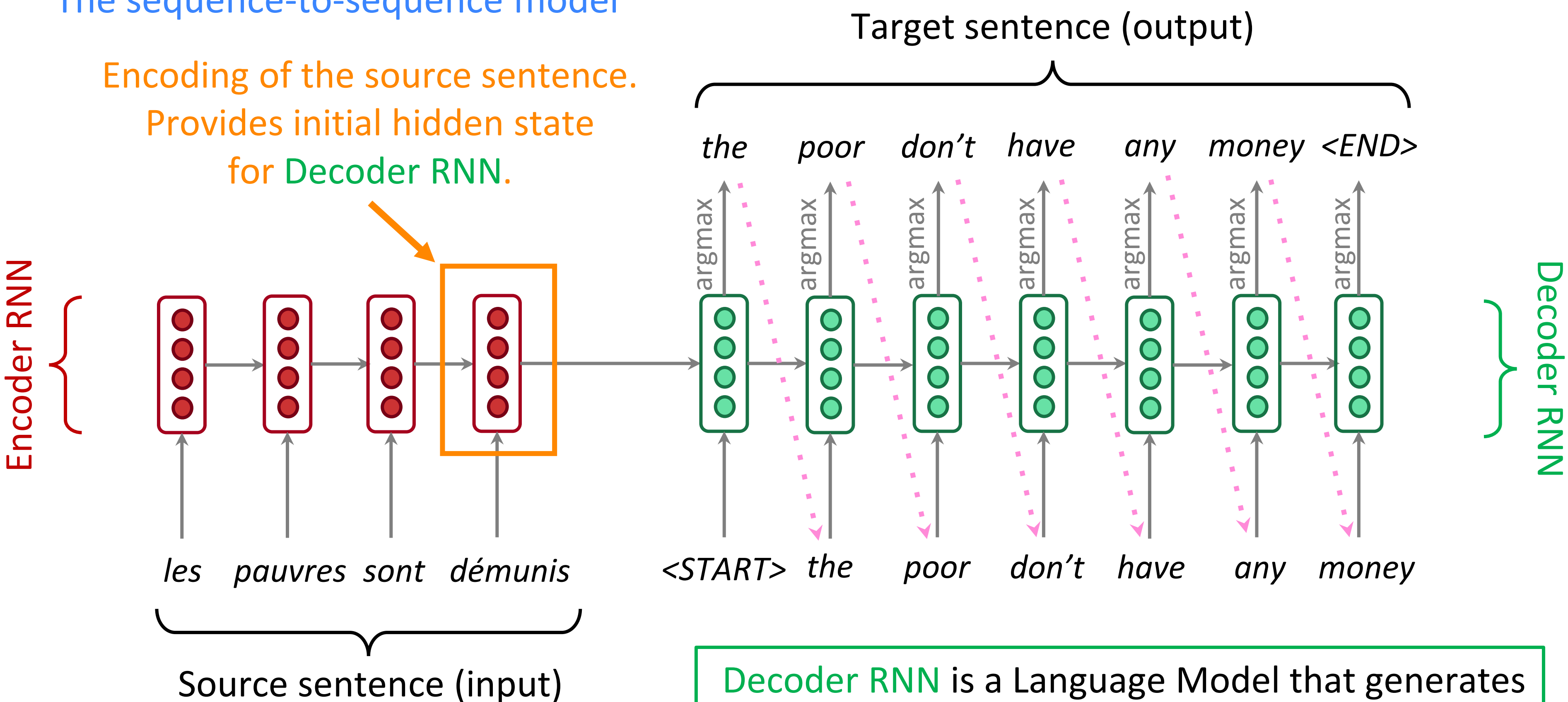


Encoder-decoder Framework



Neural Machine Translation (NMT)

The sequence-to-sequence model



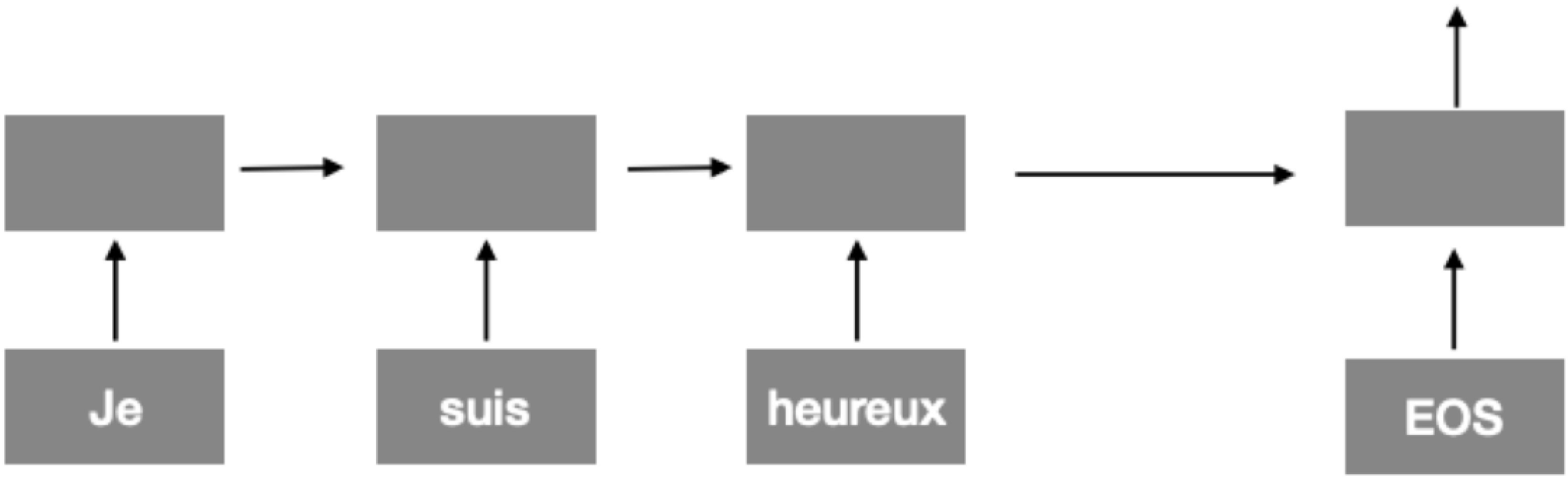
Decoder RNN is a Language Model that generates target sentence conditioned on encoding.

Encoder RNN produces an encoding of the source sentence.

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

Training

- Như trong mô hình RNN khác, chúng ta có thể huấn luyện bằng cách minimizing hàm loss giữa những gì chúng ta dự đoán ở mỗi bước và giá trị đúng của nó.



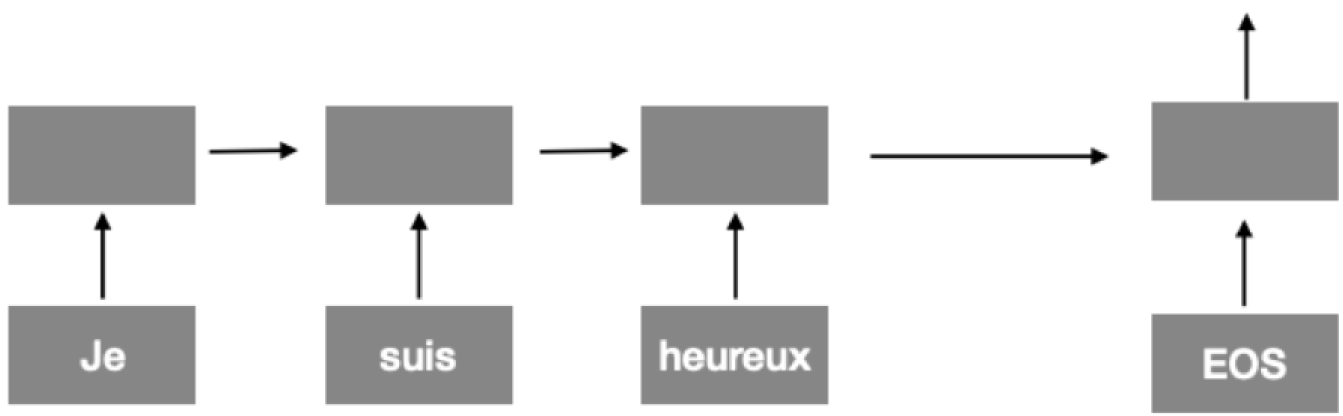
Training

Truth

I'm	you	are	the	…
1	0	0	0	0

Predicted

I'm	you	are	the	…
0.03	0.05	0.02	0.01	0.009

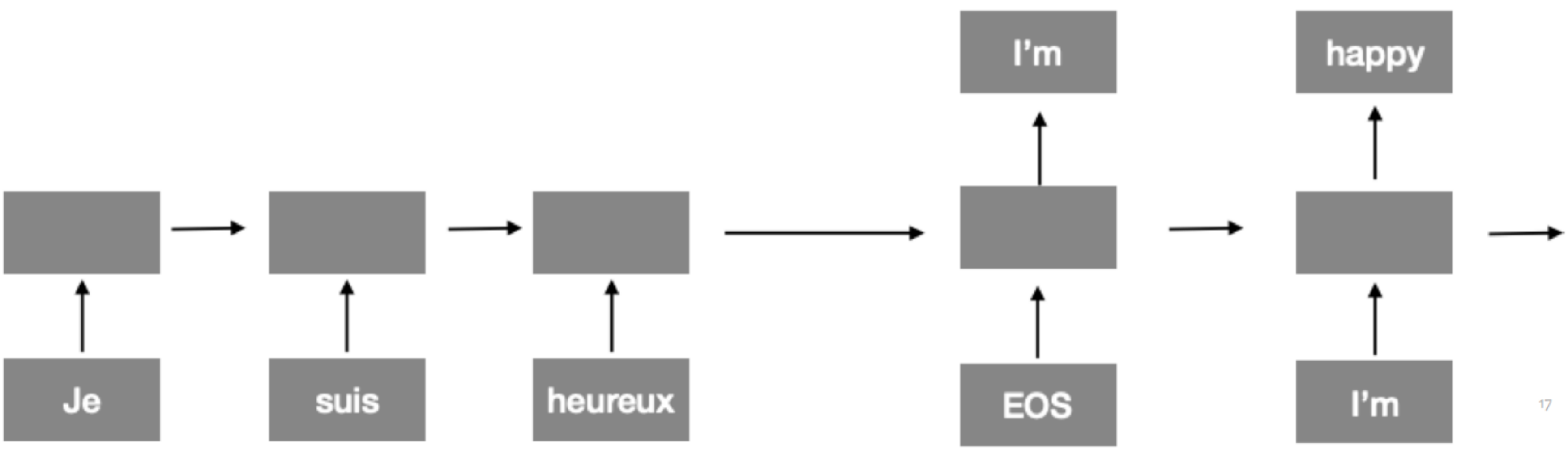


Truth

happy	great	bad	ok	...
1	0	0	0	0

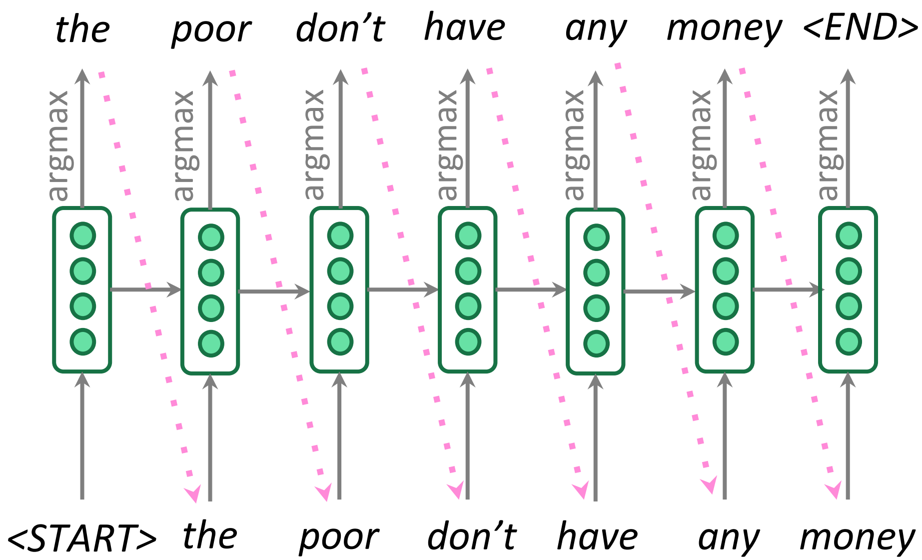
Predicted

happy	great	bad	ok	...
0.13	0.08	0.01	0.03	0.009



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

Giải mã dựa vào Beam search

- 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!

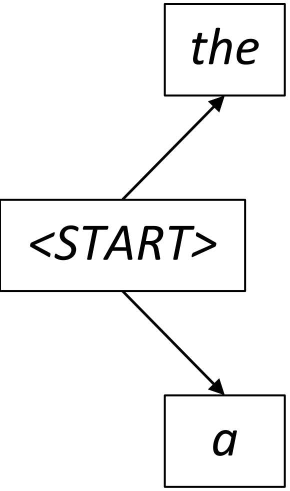
Giải mã dựa vào Beam search: ví dụ

Beam size = 2

<START>

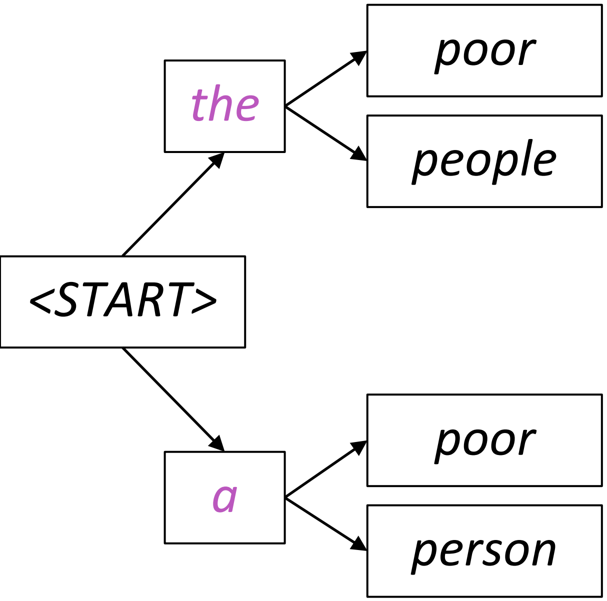
Giải mã dựa vào Beam search: ví dụ

Beam size = 2



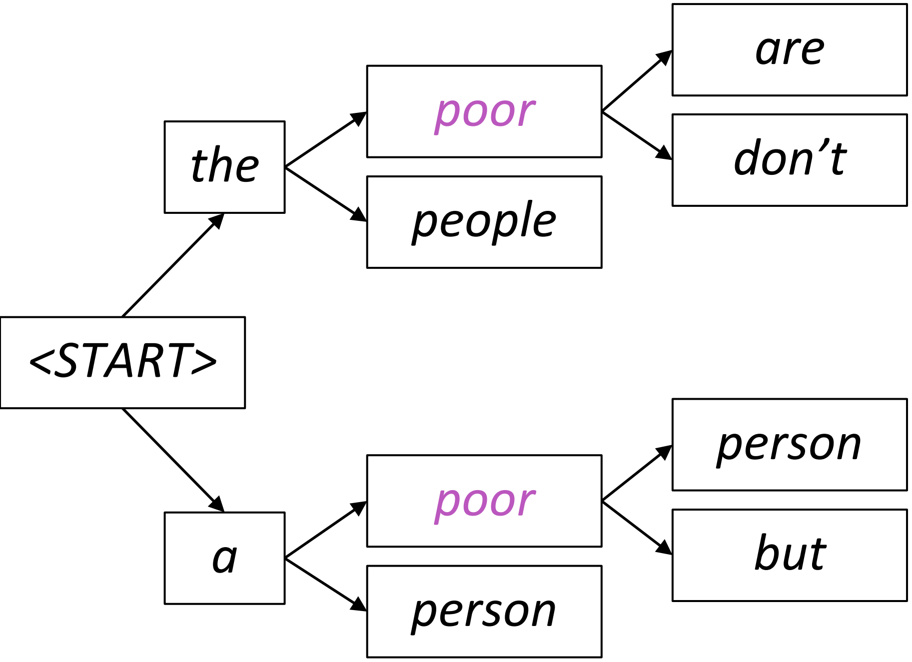
Giải mã dựa vào Beam search: ví dụ

Beam size = 2



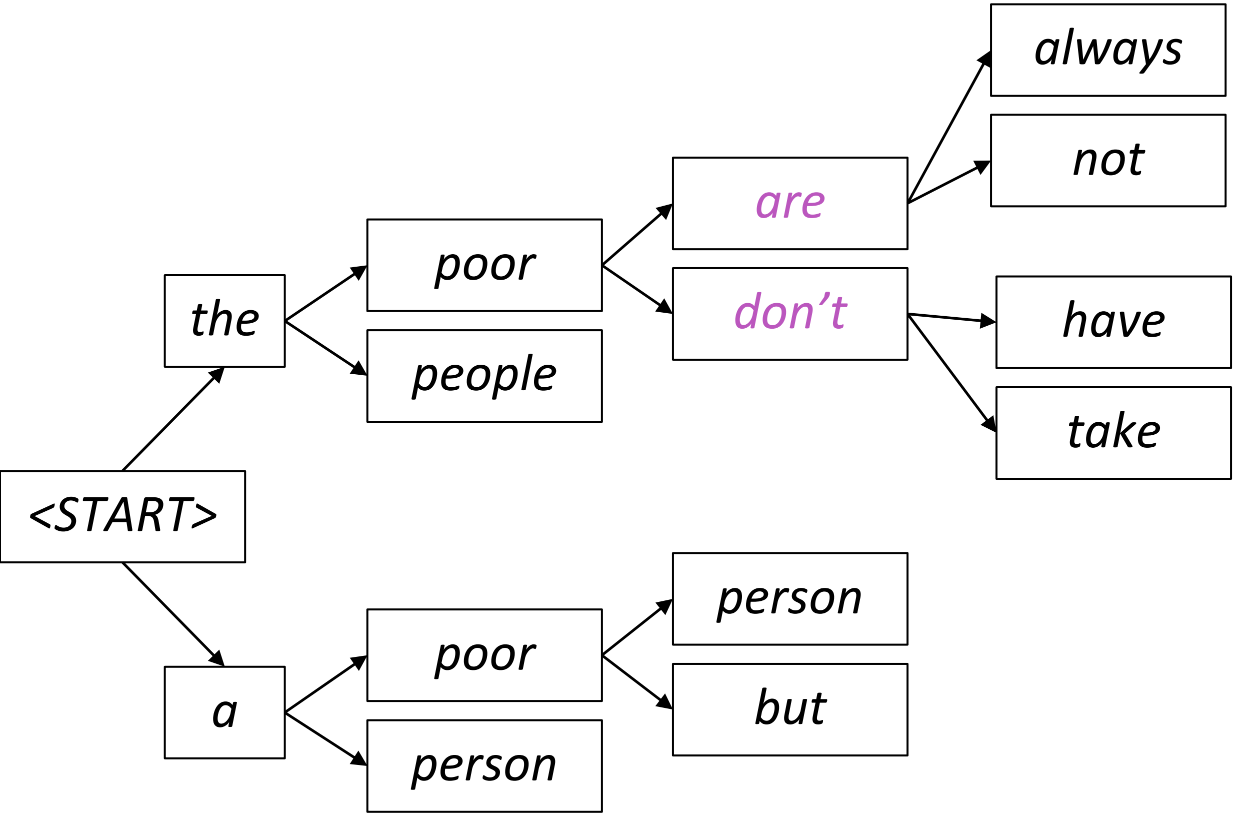
Giải mã dựa vào Beam search: ví dụ

Beam size = 2



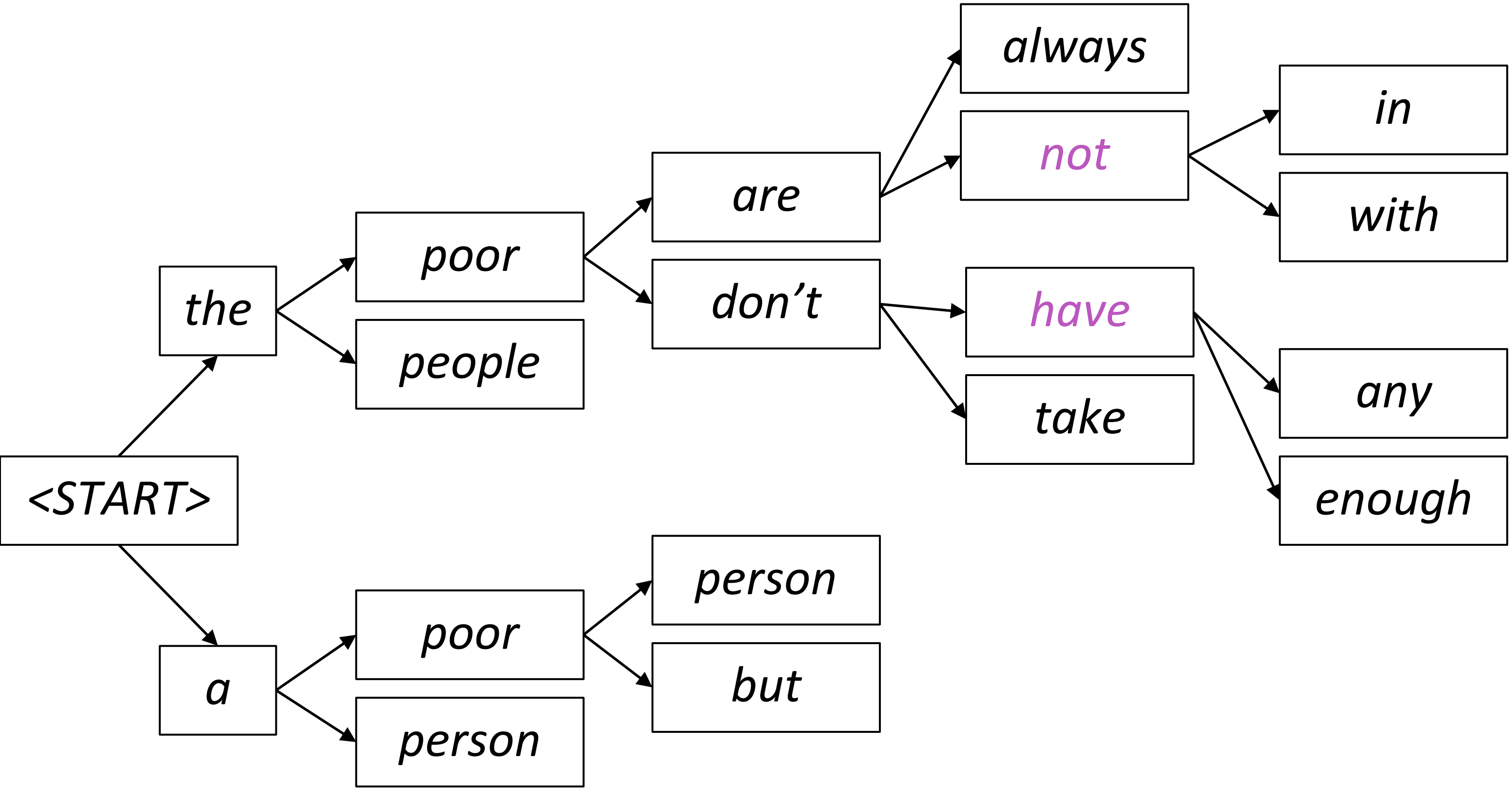
Giải mã dựa vào Beam search: ví dụ

Beam size = 2



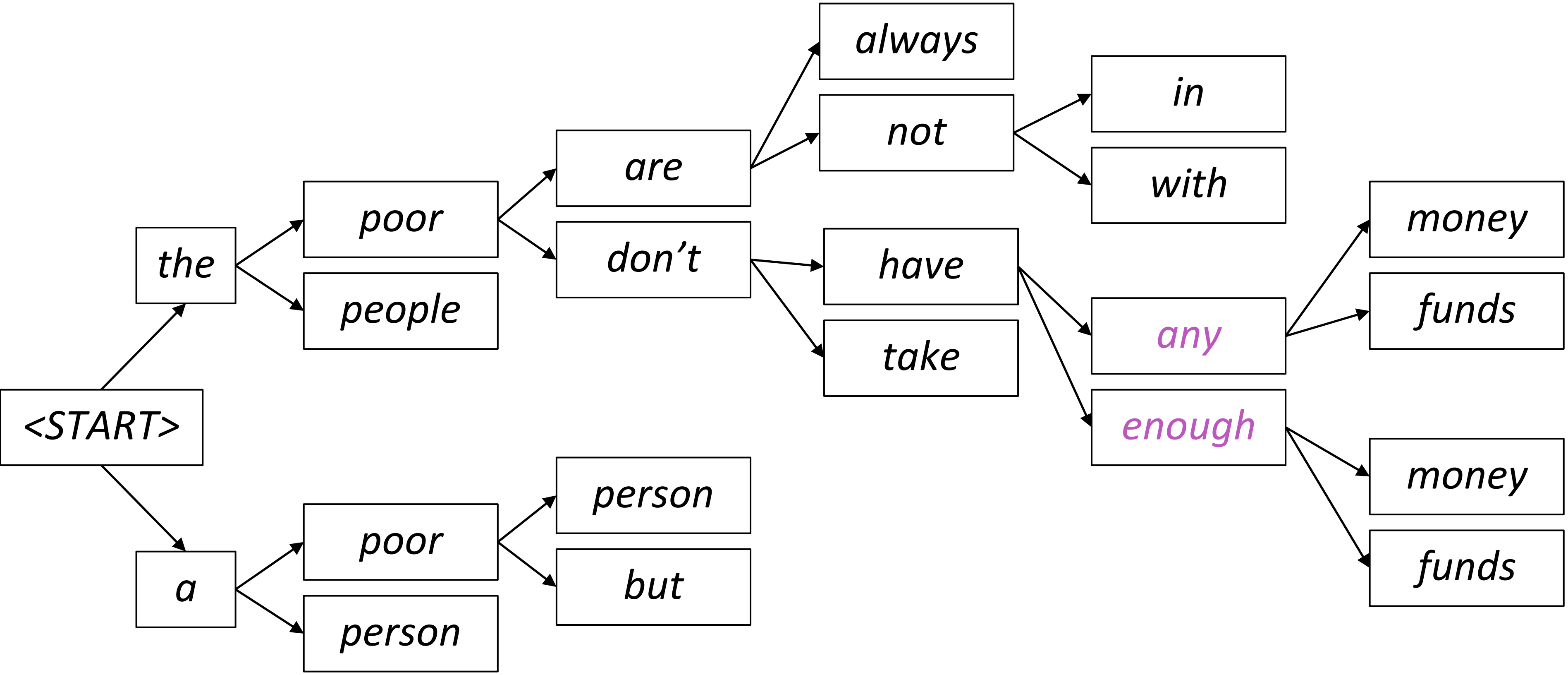
Giải mã dựa vào Beam search: ví dụ

Beam size = 2



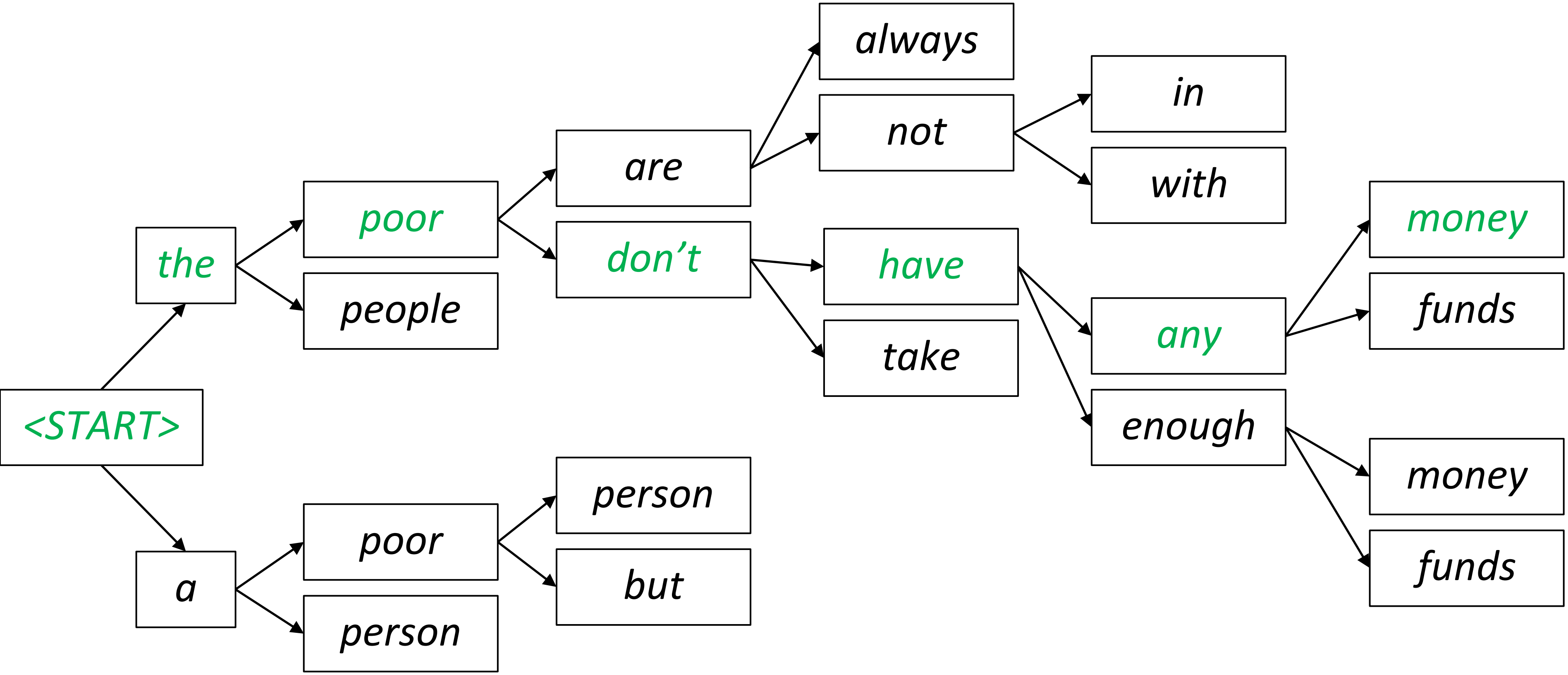
Giải mã dựa vào Beam search: ví dụ

Beam size = 2



Giải mã dựa vào Beam search: ví dụ

Beam size = 2



Beam search: Tiêu chí stopping

- Trong giải mã tham lam, chúng ta thường giải mã cho đến khi mô hình sinh ra token <END>

Ví dụ : *<START> he hit me with a pie <END>*

- Trong giải mã beam search, các giả thuyết khác nhau có thể tạo ra token <END> ở các bước thời gian khác nhau.
  - Khi một giả thuyết sinh ra <END>, thì giả thuyết đó đã hoàn thành.
  - Đặt nó sang một bên và tiếp tục khám phá các giả thuyết khác thông qua beam Search.
- Thông thường, chúng tôi tiếp tục beam search cho đến khi:
  - Chúng ta đạt đến bước thời gian T (T là một số ngưỡng đã được xác định trước) hoặc
  - Chúng ta có ít nhất n giả thuyết đã hoàn thành (trong đó n là ngưỡng được xác định trước)

Beam search: Kết thúc

- We have our list of completed hypotheses.
- How to select top one with highest score?

- Each hypothesis $y_1, \ldots, y_t$ on our list has a score

$$\text{score}(y_1, \ldots, y_t) = \log P_{\text{LM}}(y_1, \ldots, y_t | x) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x$$

- <u>Problem with this:</u> longer hypotheses have lower scores

- <u>Fix:</u> Normalize by length. Use this to select top one instead:

$$\frac{1}{t} \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$$

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!

How do we evaluate Machine Translation?

BLEU (**Bil**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 ☹

How do we evaluate Machine Translation?

- Precision n-gram = Number of correct predicted n-grams /
  Number of total predicted n-grams

- **Correct sentence:** *The guard arrived late because it was raining*
- **Predicted sentence:** *The guard arrived late because of the rain*

Precision 1-gram ($p1$) = 5 / 8

Precision 2-gram ($p2$)= 4 / 7

recision 3-gram ($p3$) = 3 / 6

Precision 4-gram ($p4$) = 2 / 5

How do we evaluate Machine Translation?

- **Geometric Average Precision Scores**

$$
\begin{aligned}
Geometric\ Average\ Precision\ (N) &= exp(\sum_{n=1}^{N} w_n \ log \ p_n) \\
&= \prod_{n=1}^{N} {p_n}^{w_n} \\
&= (p_1)^{\frac{1}{4}} \cdot (p_2)^{\frac{1}{4}} \cdot (p_3)^{\frac{1}{4}} \cdot (p_4)^{\frac{1}{4}}
\end{aligned}
$$

How do we evaluate Machine Translation?

- **Correct sentence:** *The guard arrived late because it was raining*
- **Predicted sentence:** *The late*

  Precision 1-gram (*p*1) = 2 / 2

  Precision 2-gram (*p*2)= 1 / 1

  ➢ khuyến khích model sinh đầu ra ngắn hơn và điểm cao hơn.

- **Brevity Penalty:** phạt những câu quá ngắn

$$Brevity\ Penalty = \begin{cases} 1, & \text{if } c > r \\ e^{(1-r/c)}, & \text{if } c <= r \end{cases}$$

- c là *predicted length* = số lượng từ có trong predicted sentence
- r là *target length* = số lượng từ có trong target sentence

How do we evaluate Machine Translation?

- **BLEU score**

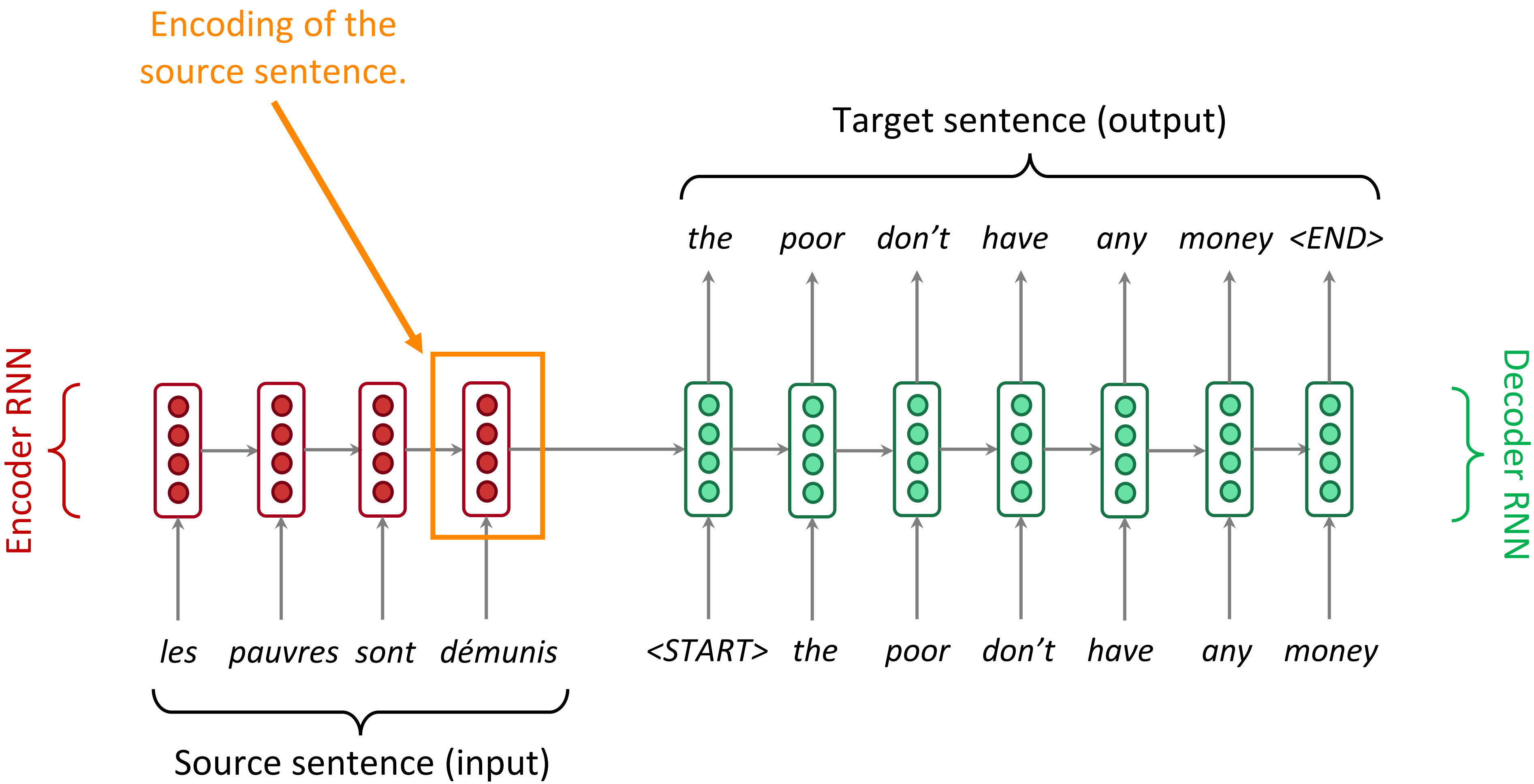
$$Bleu\ (N) = Brevity\ Penalty \cdot Geometric\ Average\ Precision\ Scores\ (N)$$

với N = số gram

- **Công thức BLEU khác**

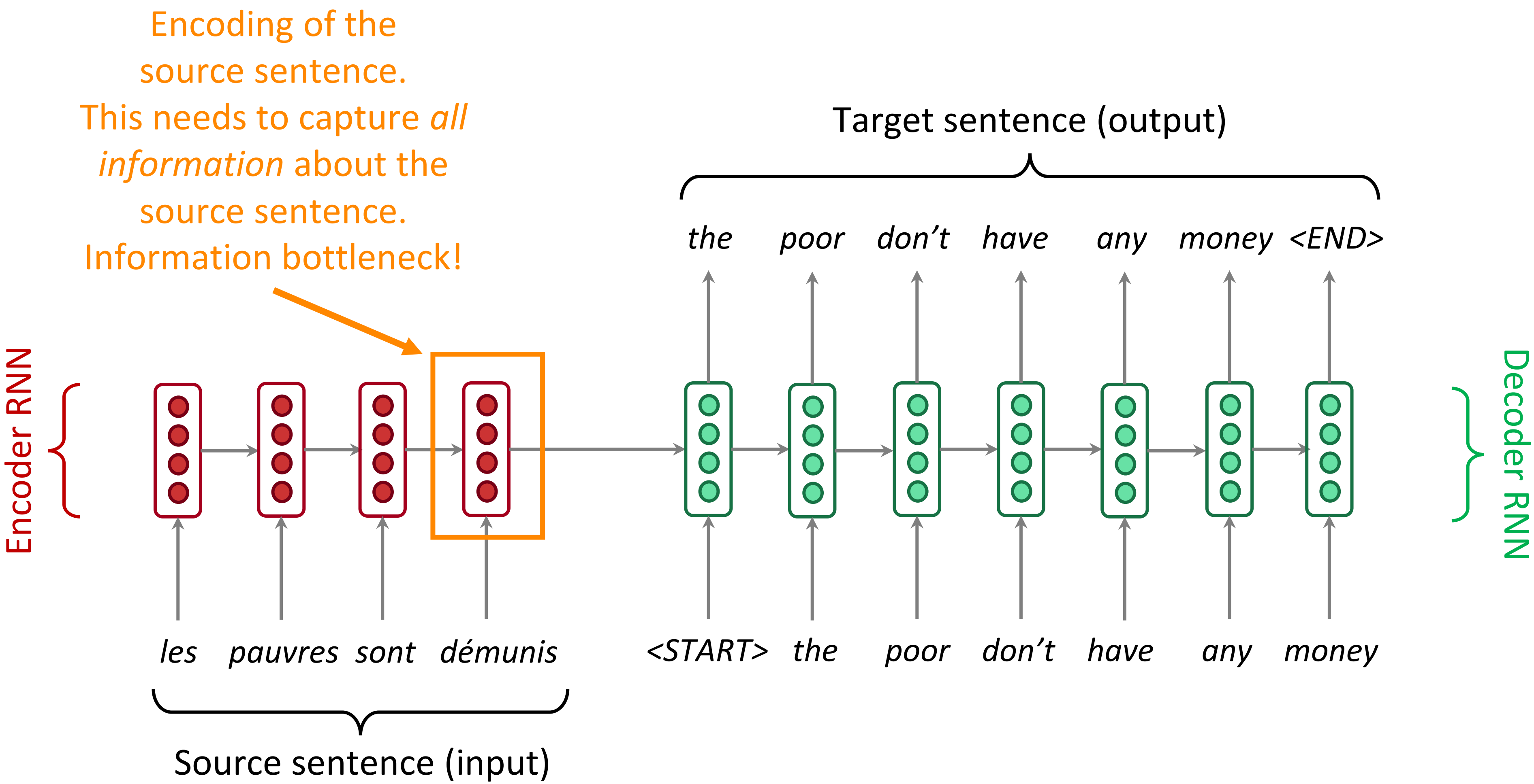
$$\begin{aligned} log\ Bleu &= min(1 - \frac{r}{c}, 0) + \sum_{n=1}^{4} \frac{log\ p_n}{4} \\ &= min(1 - \frac{r}{c}, 0) + \frac{log\ p_1 + log\ p_2 + log\ p_3 + log\ p_4}{4} \end{aligned}$$

Sequence-to-sequence: Vấn đế nút cổ chai



Problems with this architecture?

Sequence-to-sequence: the bottleneck problem

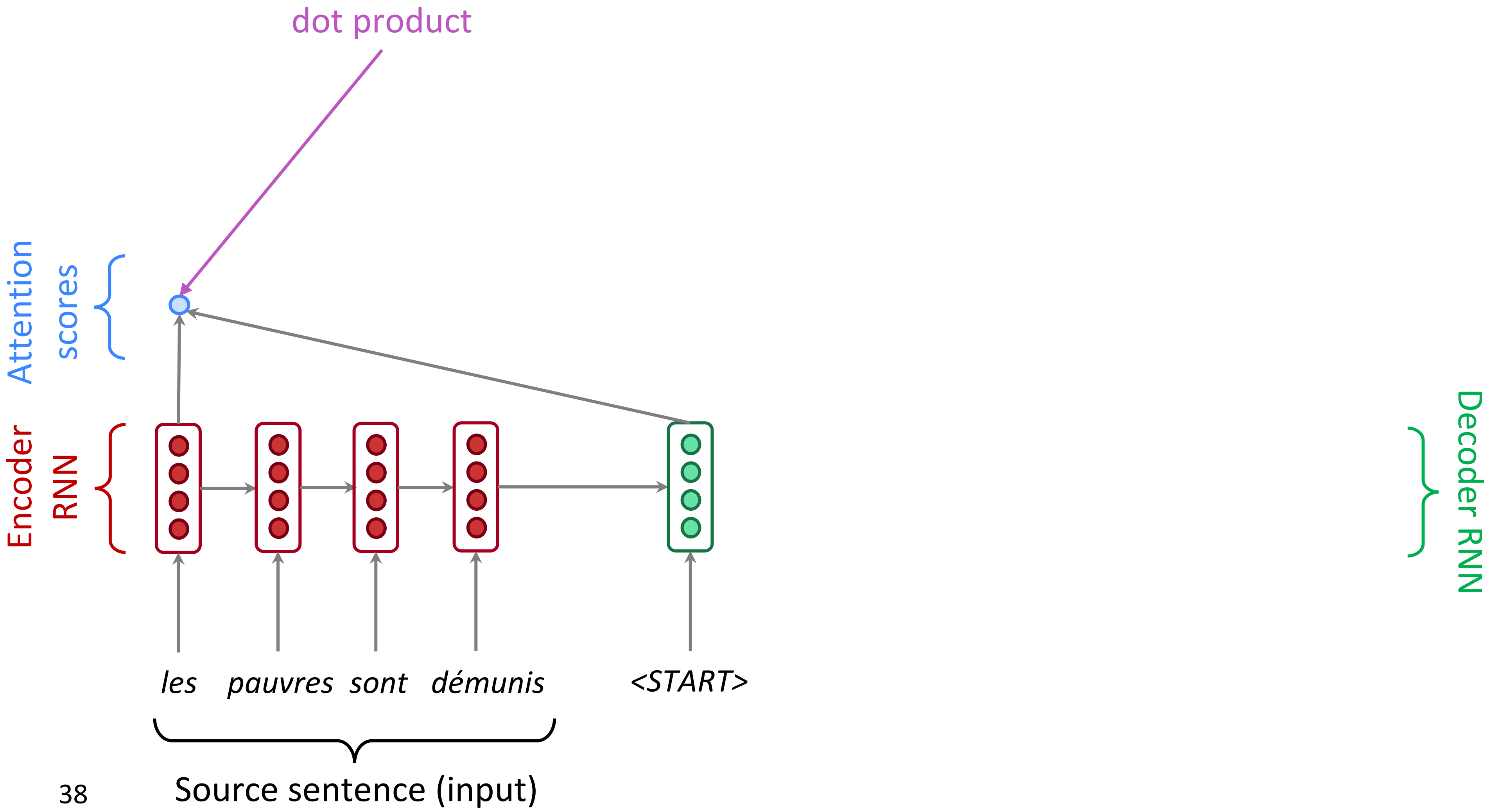


Attention

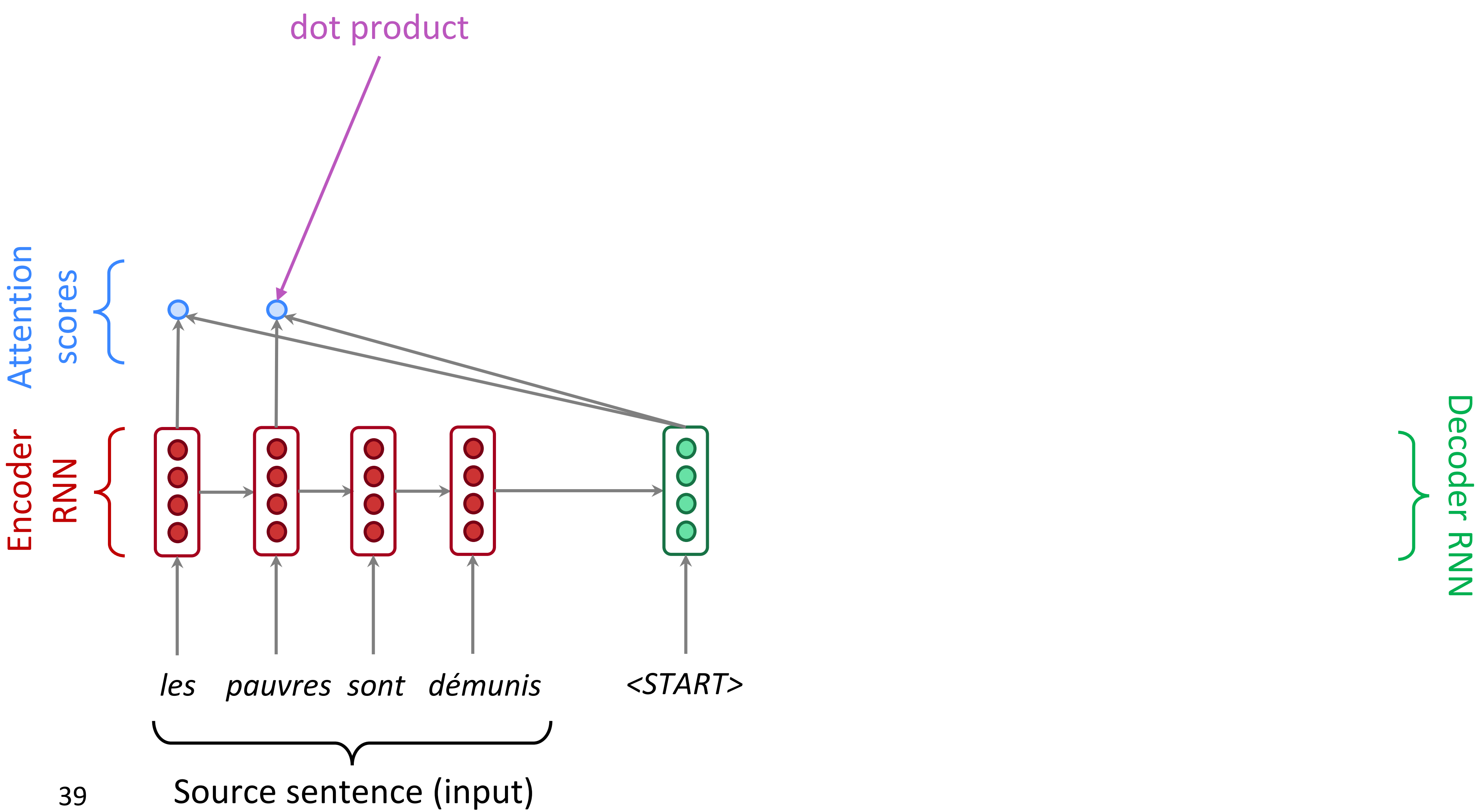
- **Attention** provides a solution to the bottleneck problem.

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

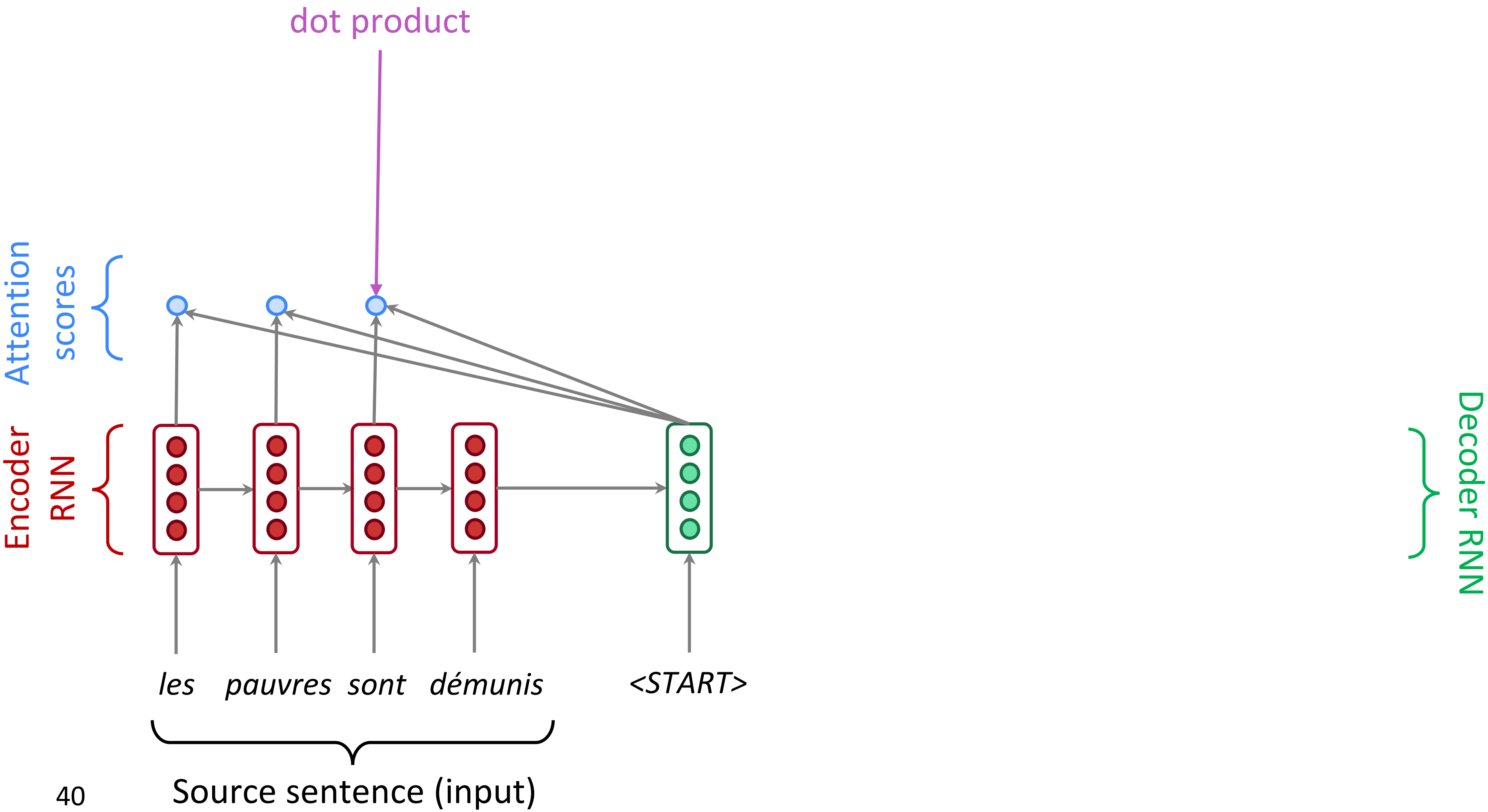
Encoder-Decoder với Attention



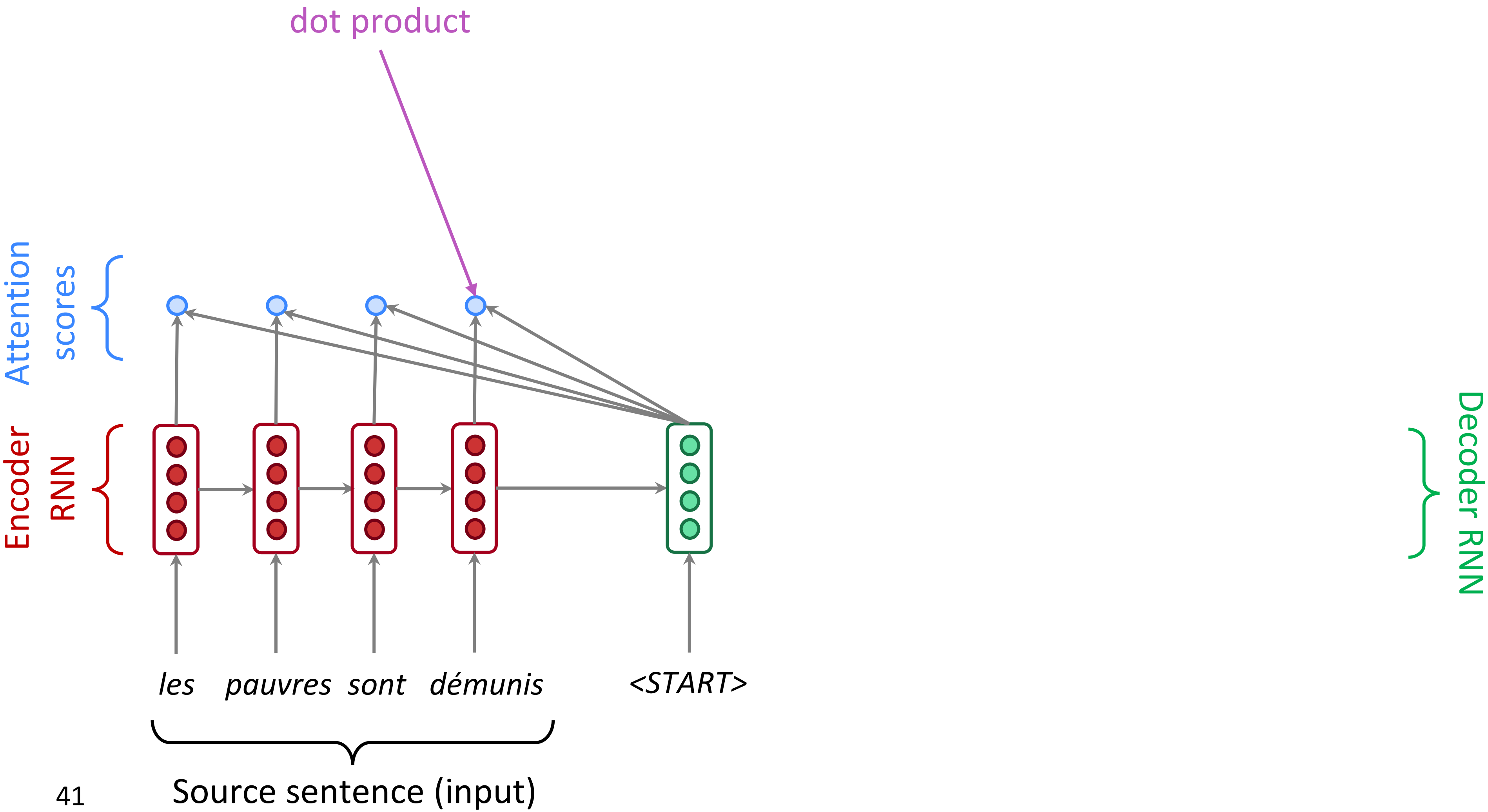
Seq2Seq với attention



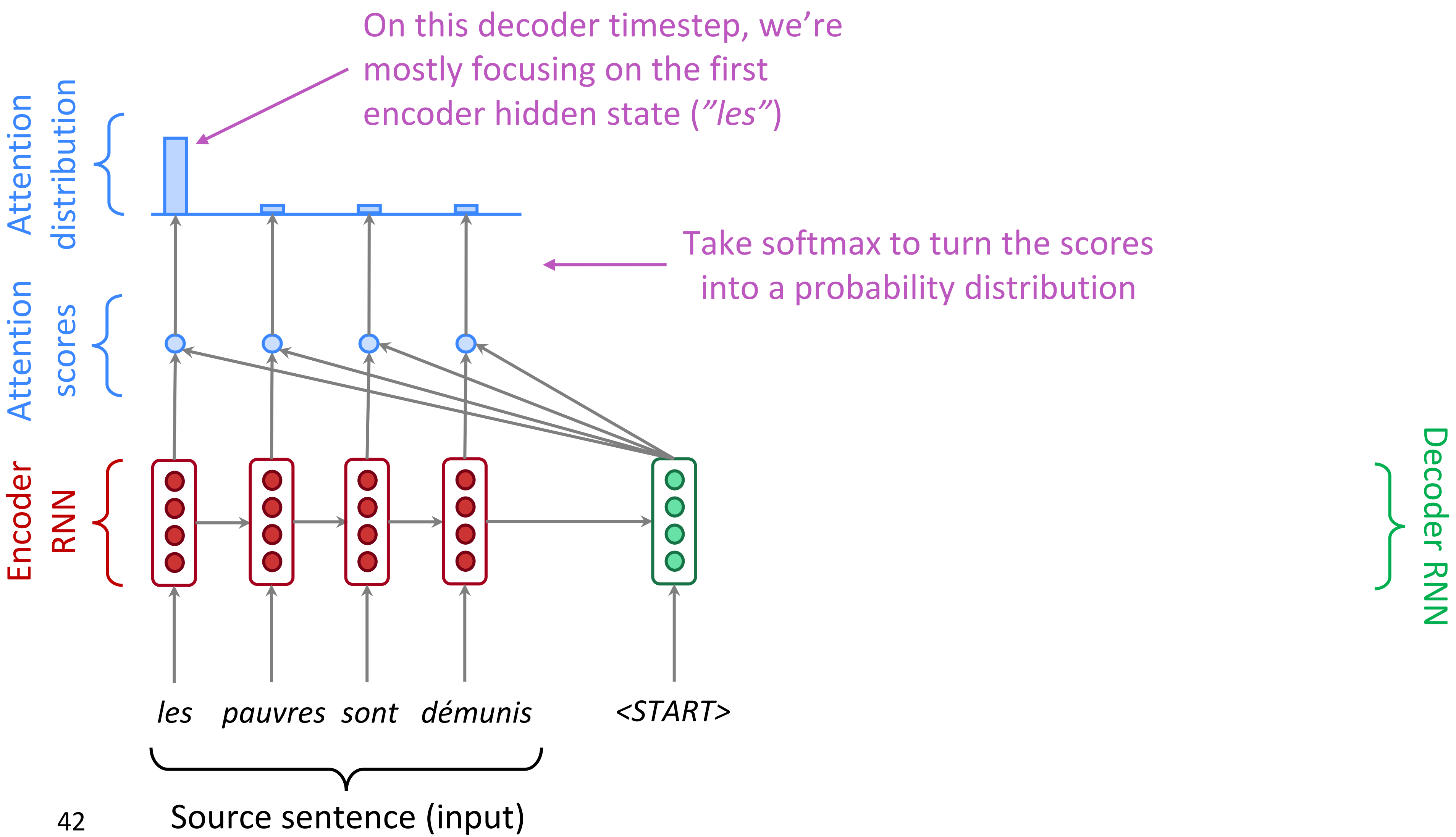
Sequence-to-sequence with attention



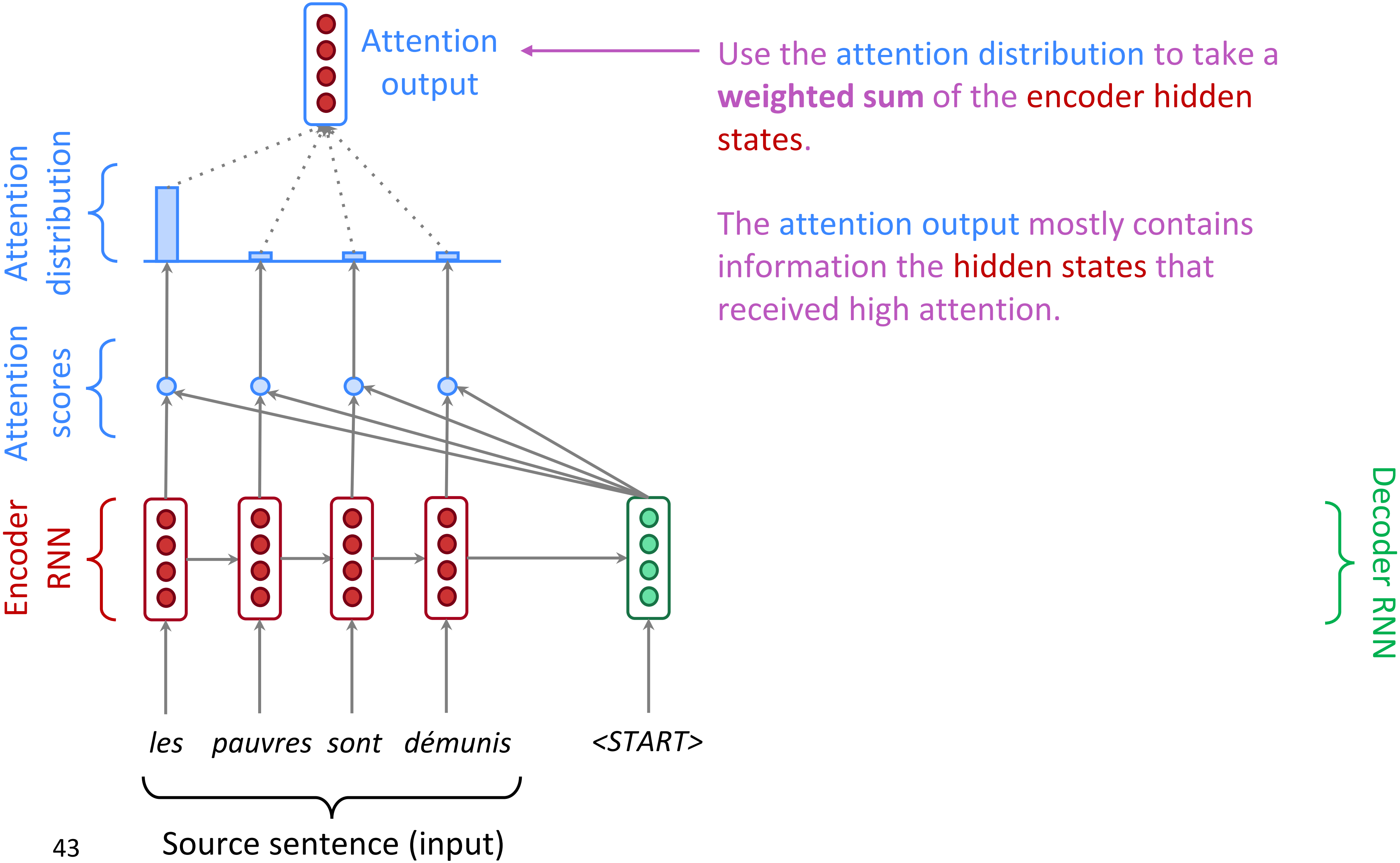
Sequence-to-sequence with attention



Sequence-to-sequence with attention



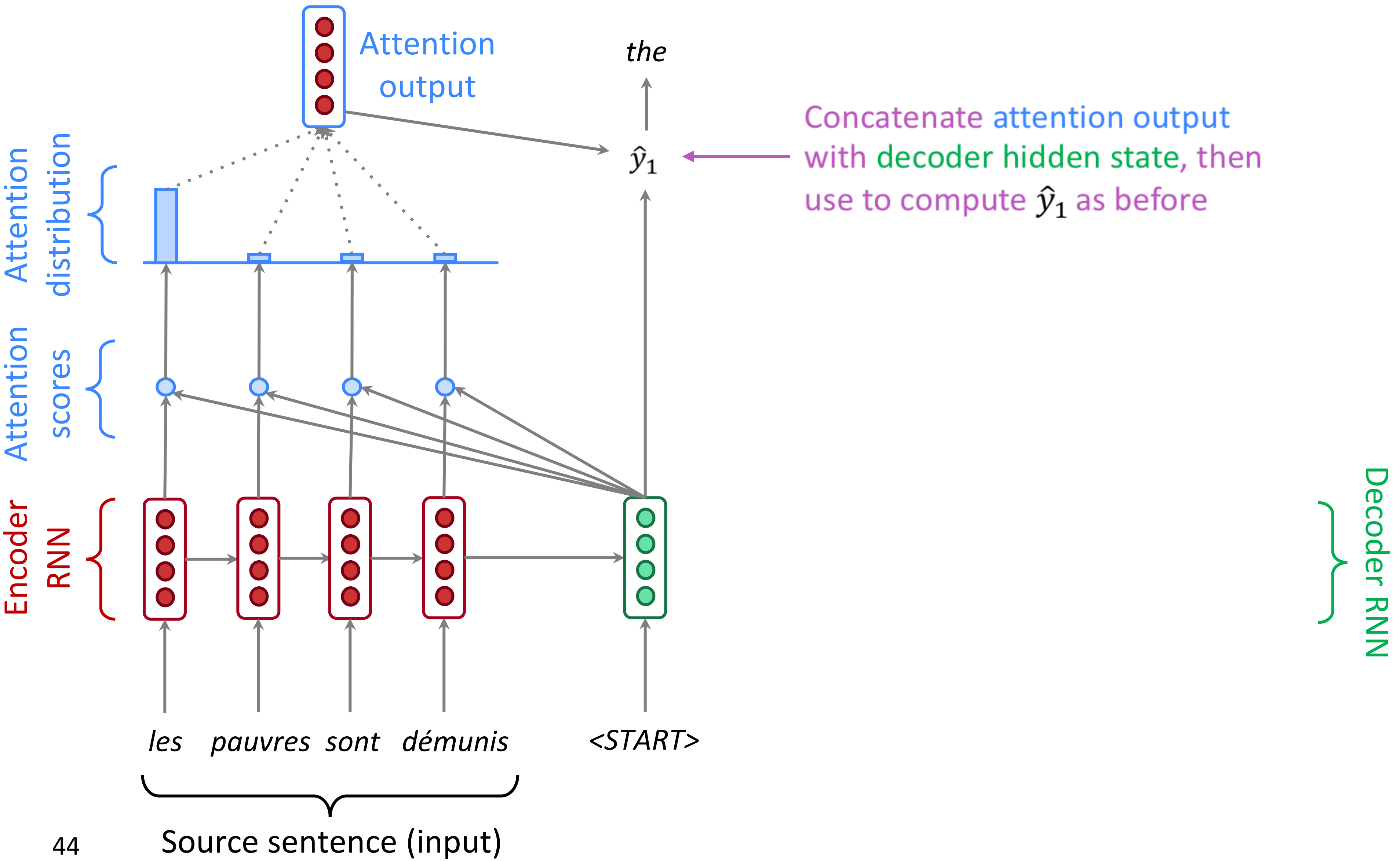
Sequence-to-sequence with attention



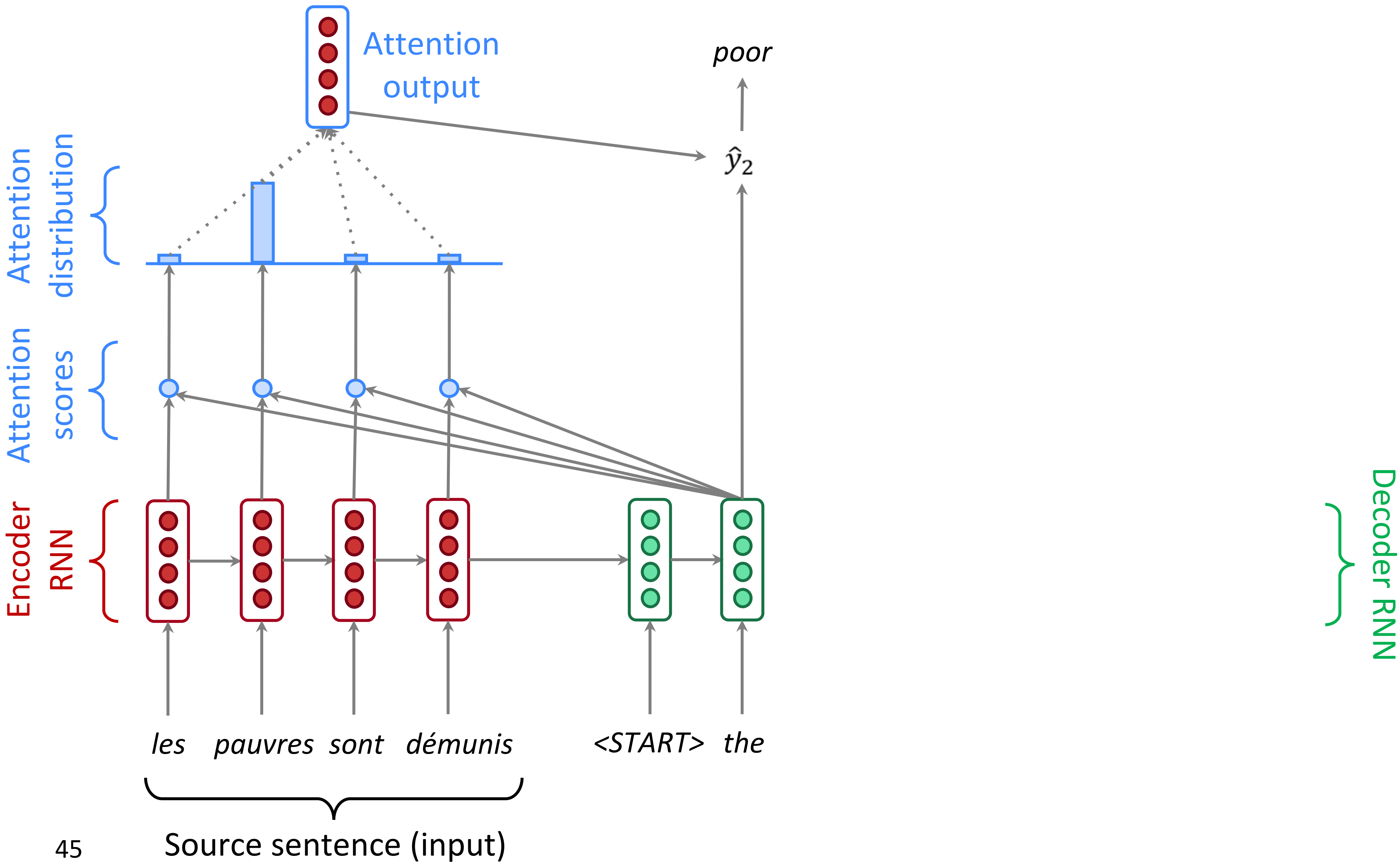
Use the attention distribution to take a **weighted sum** of the encoder hidden states.

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

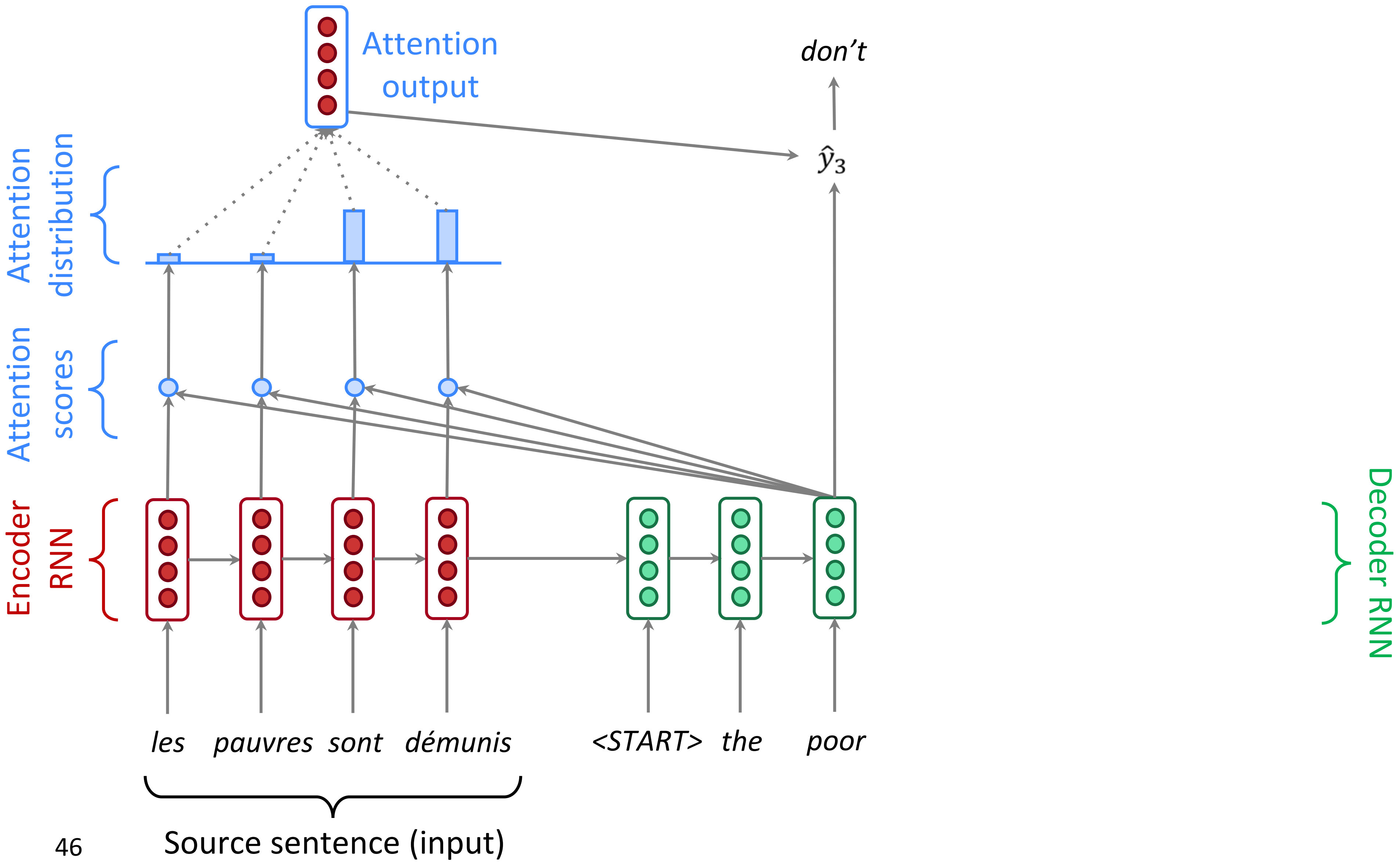
Sequence-to-sequence with attention



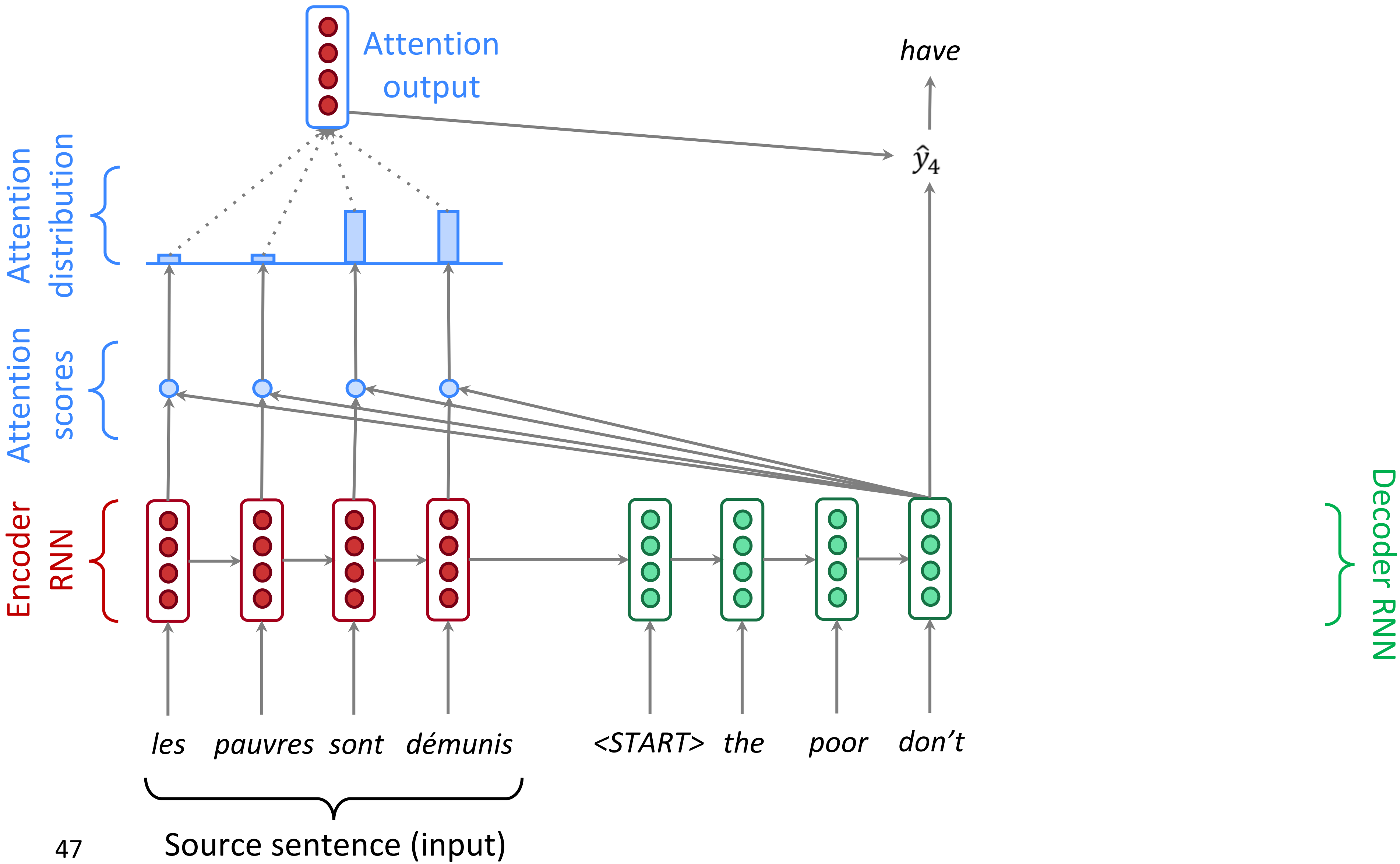
Sequence-to-sequence with attention



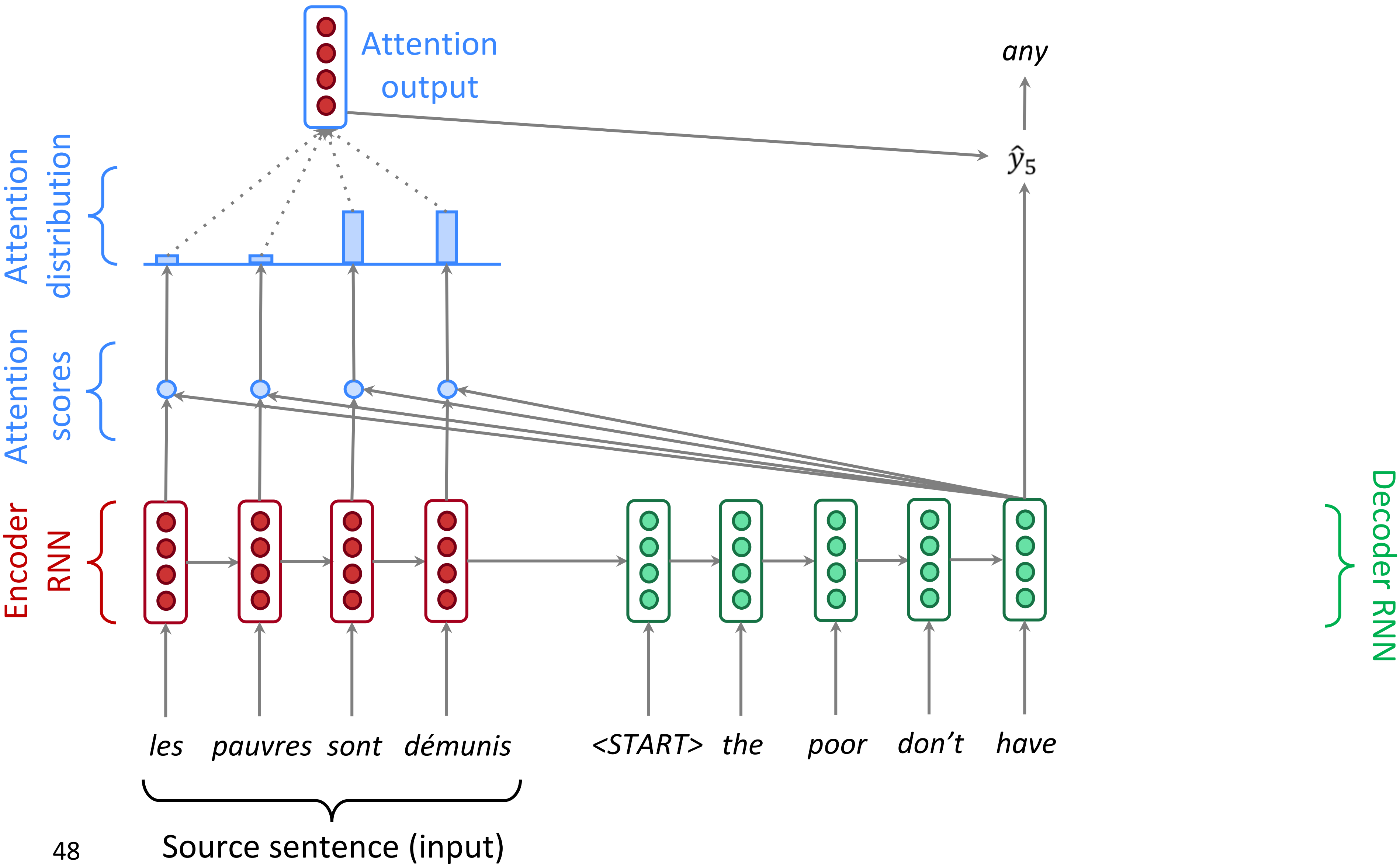
Sequence-to-sequence with attention



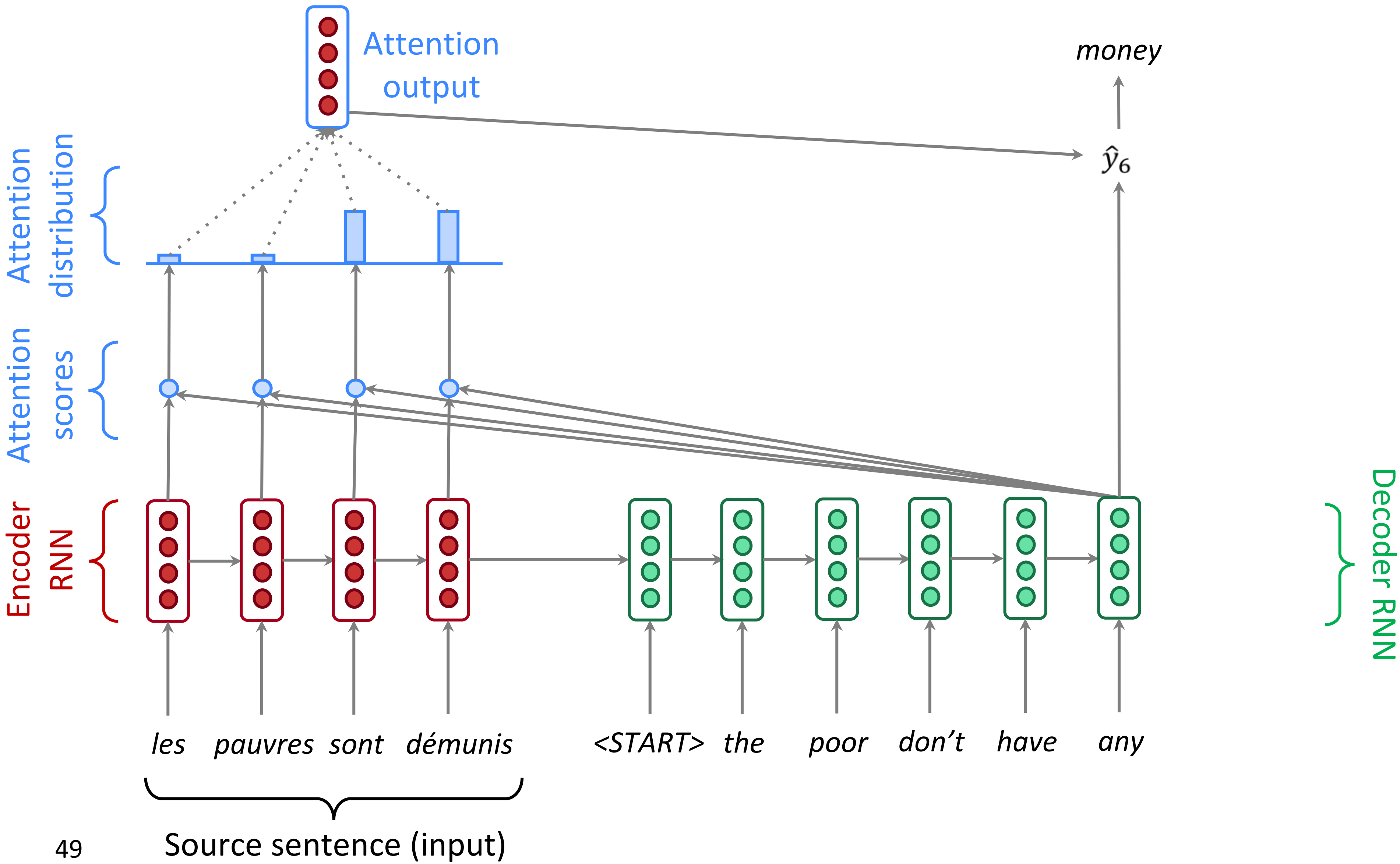
Sequence-to-sequence with attention



Sequence-to-sequence with attention



Sequence-to-sequence with attention



Attention: Công thức

- 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)

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

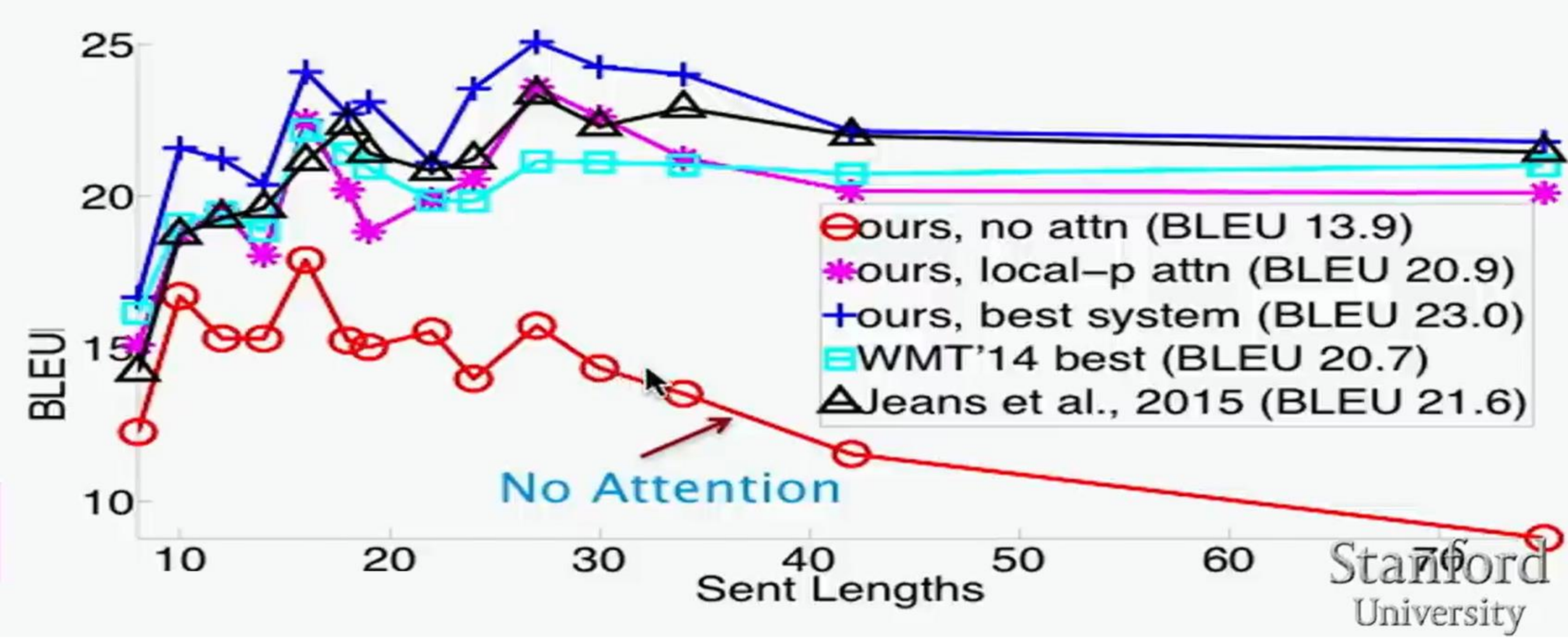
- We use α^t to take a weighted sum of the encoder hidden states to get the attention output $\boldsymbol{a}_t$

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

- 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 dịch tốt các câu dài

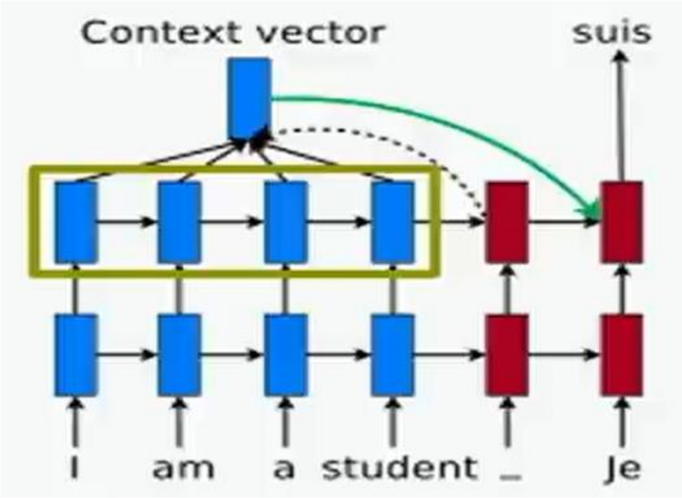


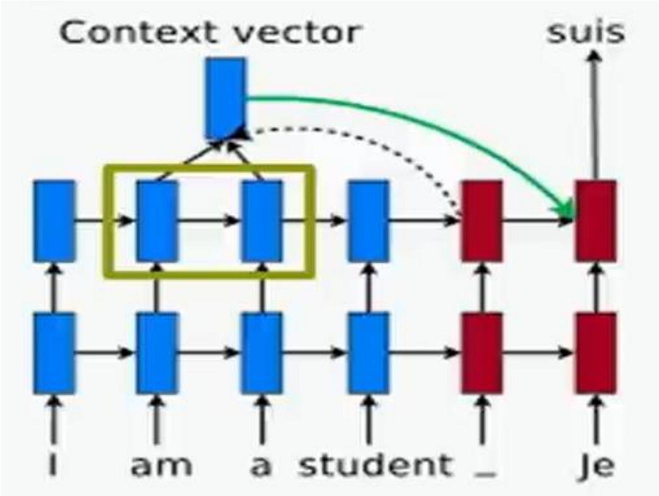
Dịch từ tiếng Anh sang tiếng Đức

source	Orlando Bloom and *Miranda Kerr* still love each other
human	Orlando Bloom und **Miranda Kerr** lieben sich noch immer
+attn	Orlando Bloom und $\boxed{\textbf{Miranda Kerr}}$ lieben einander noch immer .
base	Orlando Bloom und $\boxed{\textbf{Lucas Miranda}}$ lieben einander noch immer .

Global vs. Local Attention

- Tránh chú ý đến mọi thứ



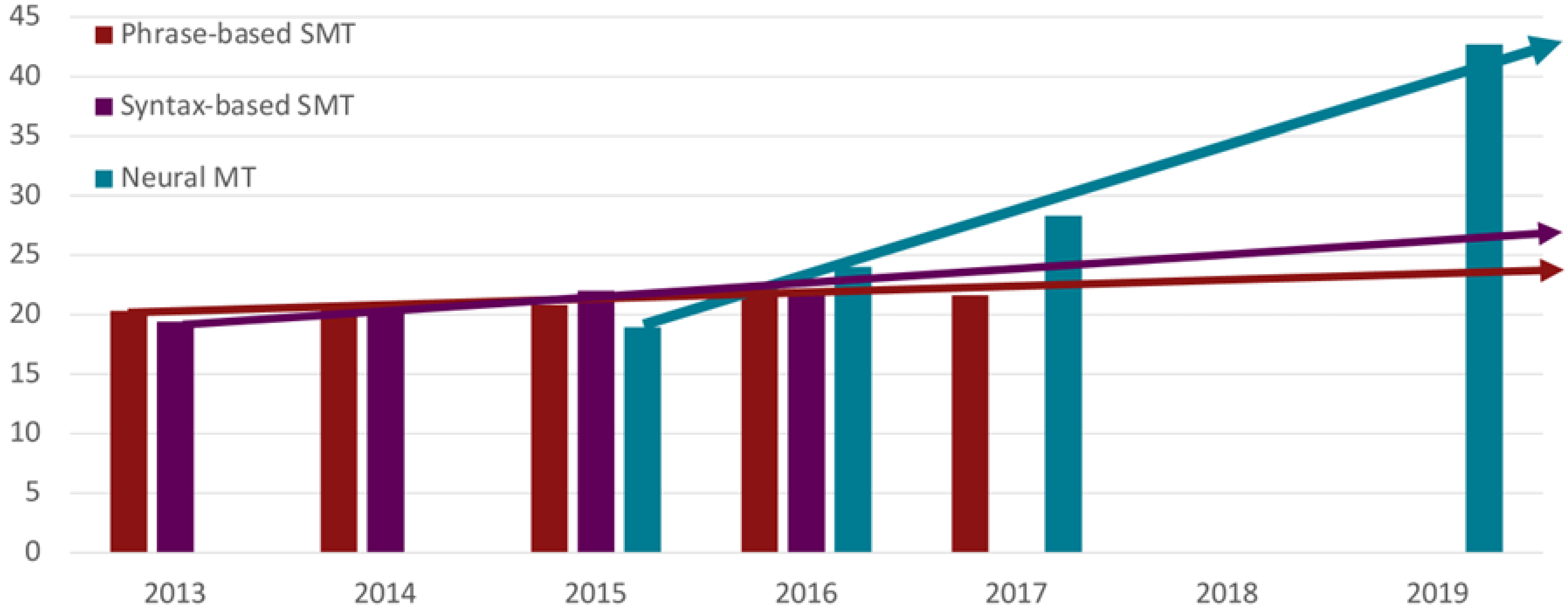


Global: all source states

Local: subset of source states

Tiến triển của hệ thống MT theo thời gian

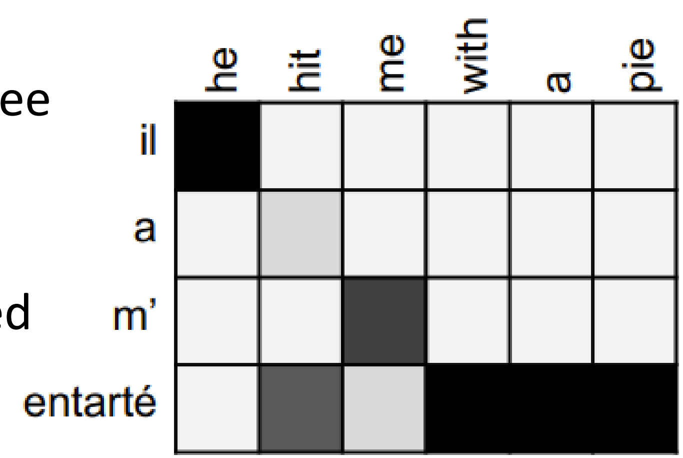
[Edinburgh En-De WMT newstest2013 Cased BLEU; NMT 2015 from U. Montréal; NMT 2019 FAIR on newstest2019]



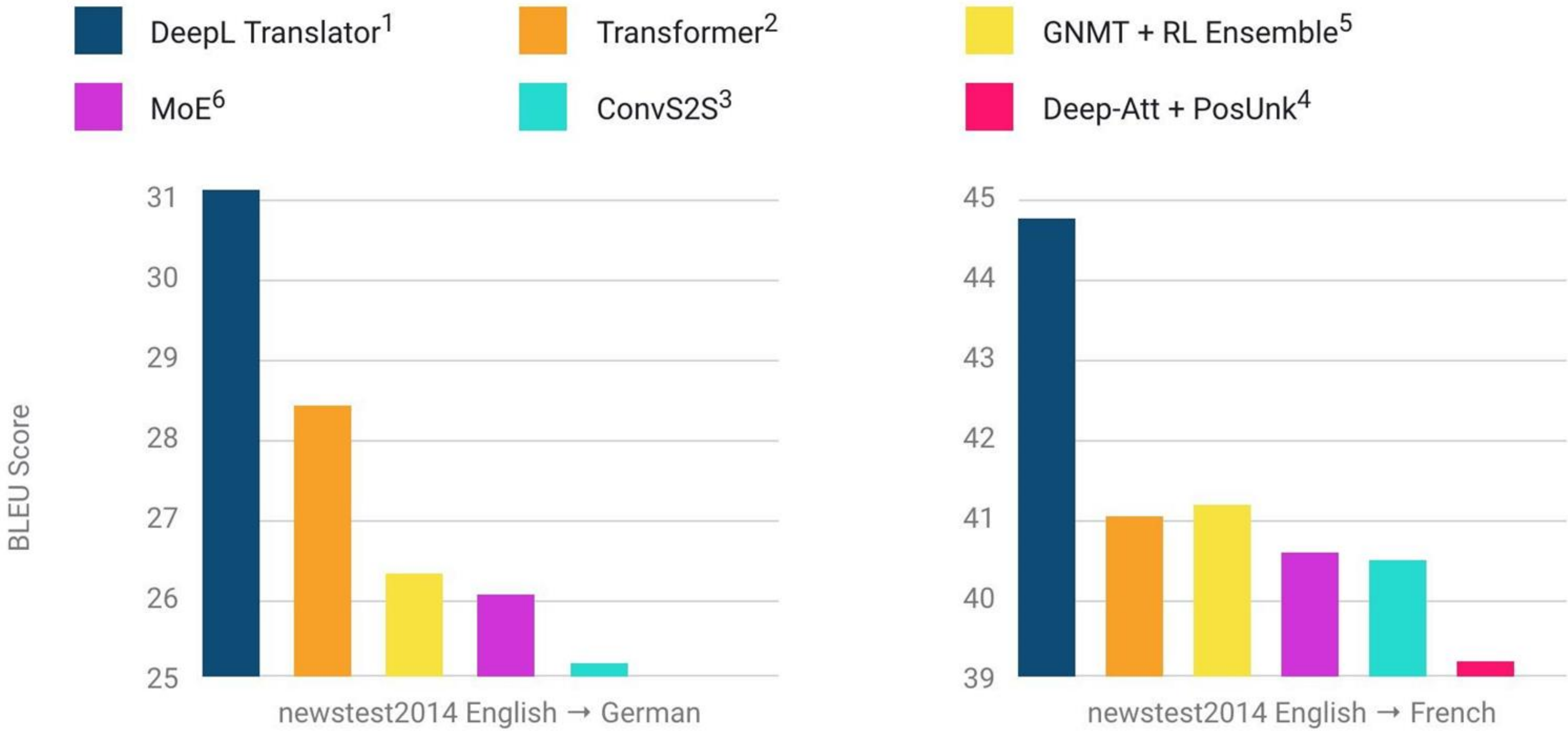
Source: Neural Machine Translation: Breaking the Performance Plateau (meta-net.eu)

Attention tuyệt vời (Bahdanau at al., 14164 citations)

- 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



Data data data



Source: DeepL's press release (Aug 2017)

NMT: Câu chuyện thành công lớn nhất của NLP Deep Learning

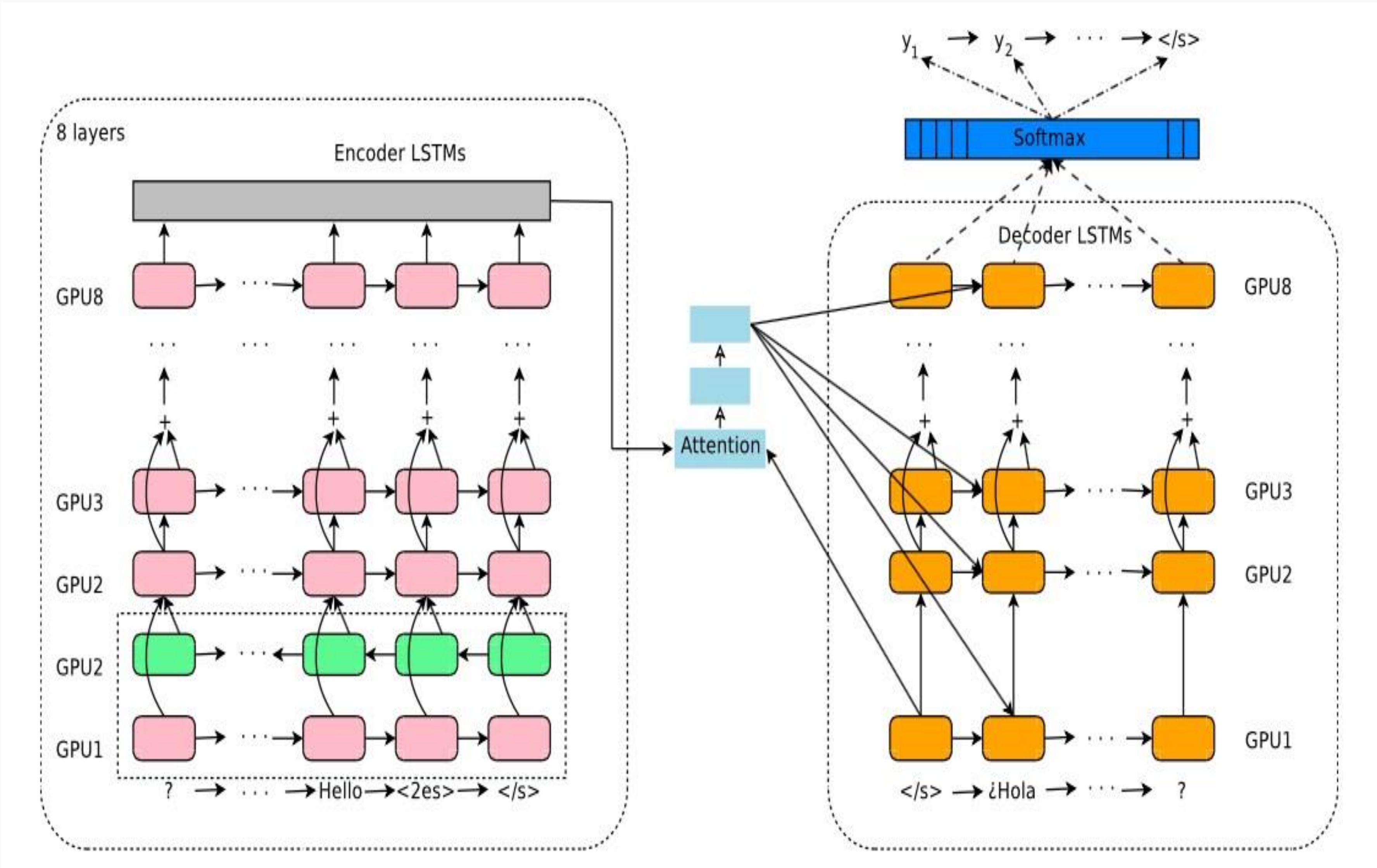
Neural Machine Translation went from a fringe research activity in **2014** to the leading standard method in **2016**

- **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

Vậy là MT đã được giải quyết chưa?

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

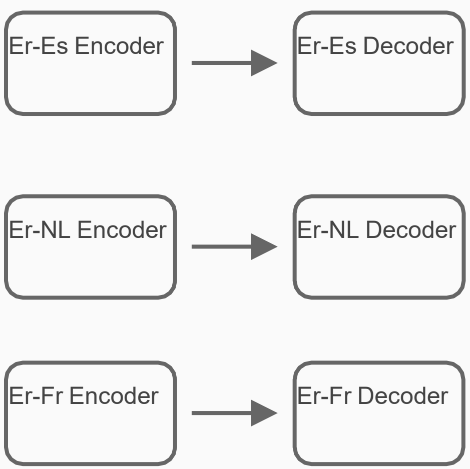
Yonghui Wu et al. (2016). Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation



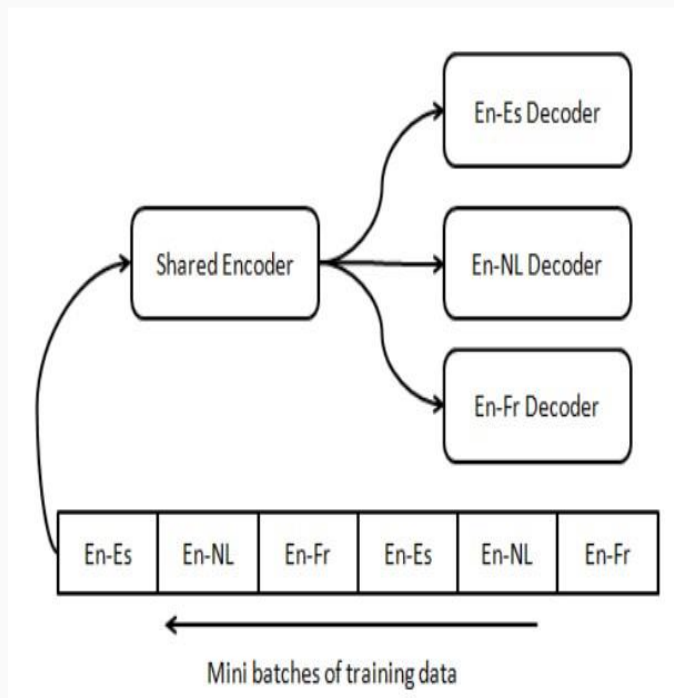
Google’s Multilingual NMT System: Enabling Zero-Shot Translation.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat,Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, Jeffrey Dean

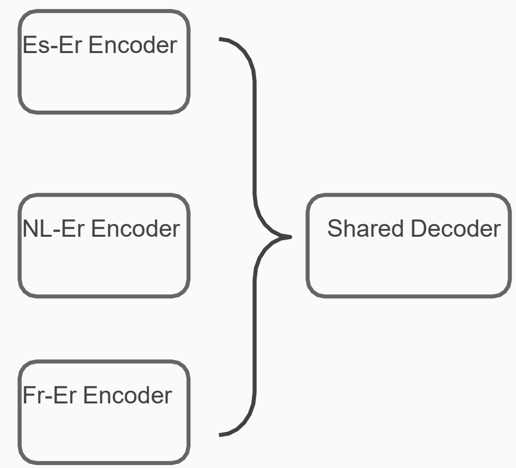
Multiple Encoders → Multiple Decoders [1]



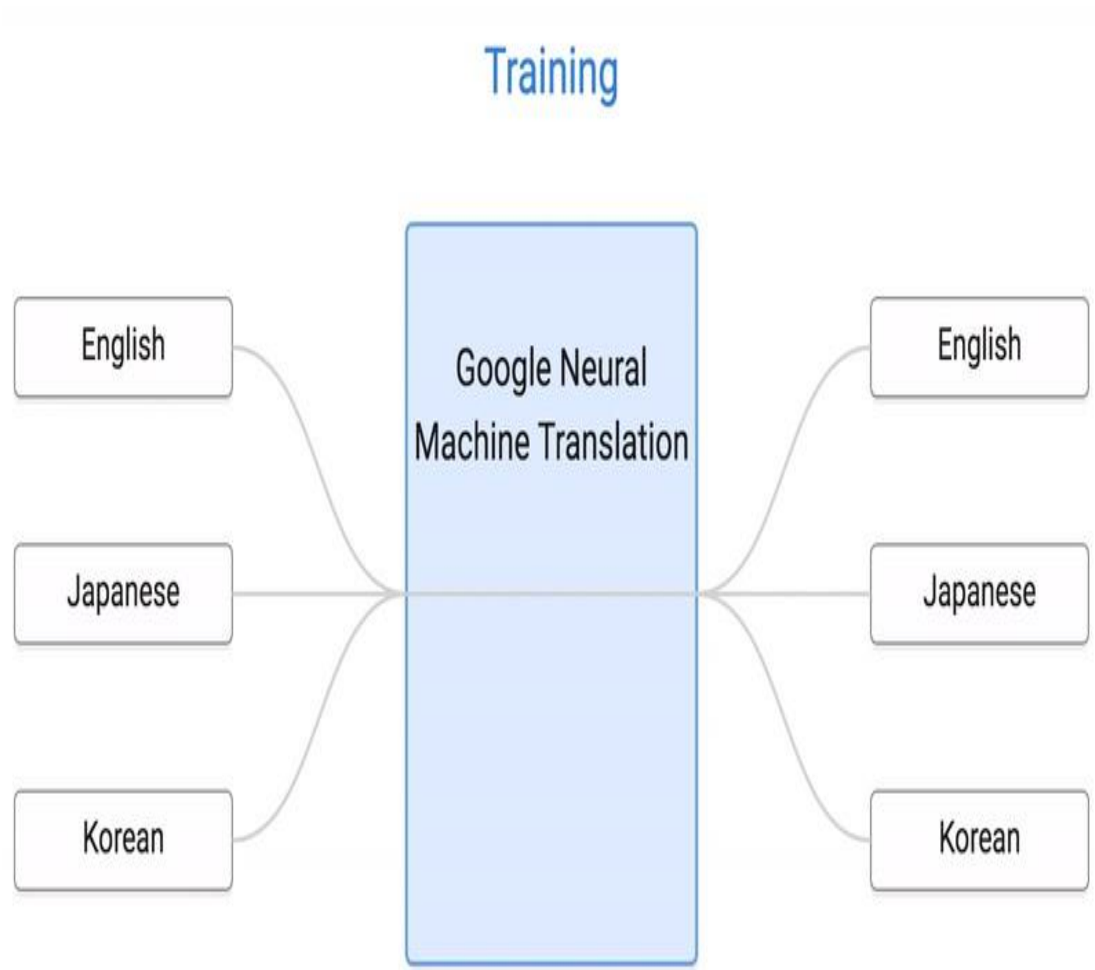
Shared Encoder → Multiple Decoder [2]



Multiple Encoders → Shared Decoder [3]



Google’s Multilingual NMT System Benefits



- **Simplicity**: single model
- **Low-resource language improvements**
- **Zero-shot translation**

Google’s Multilingual NMT System Architecture

Artificial token at the beginning of the input sentence to indicate the target language

Hello, how are you? -> ¿Hola como estás?

Add <2es> to indicate that
Spanish is the target language

<2es> Hello, how are you? -> ¿Hola como estás?

Google's Multilingual NMT System Experiments

- WMT'14:
  - Comparable performance: English → French
  - State-of-the-art: English → German, French → English
- WMT'15:
  - State-of-the-art: German → English

Google's Multilingual NMT System Zero-Shot Translation

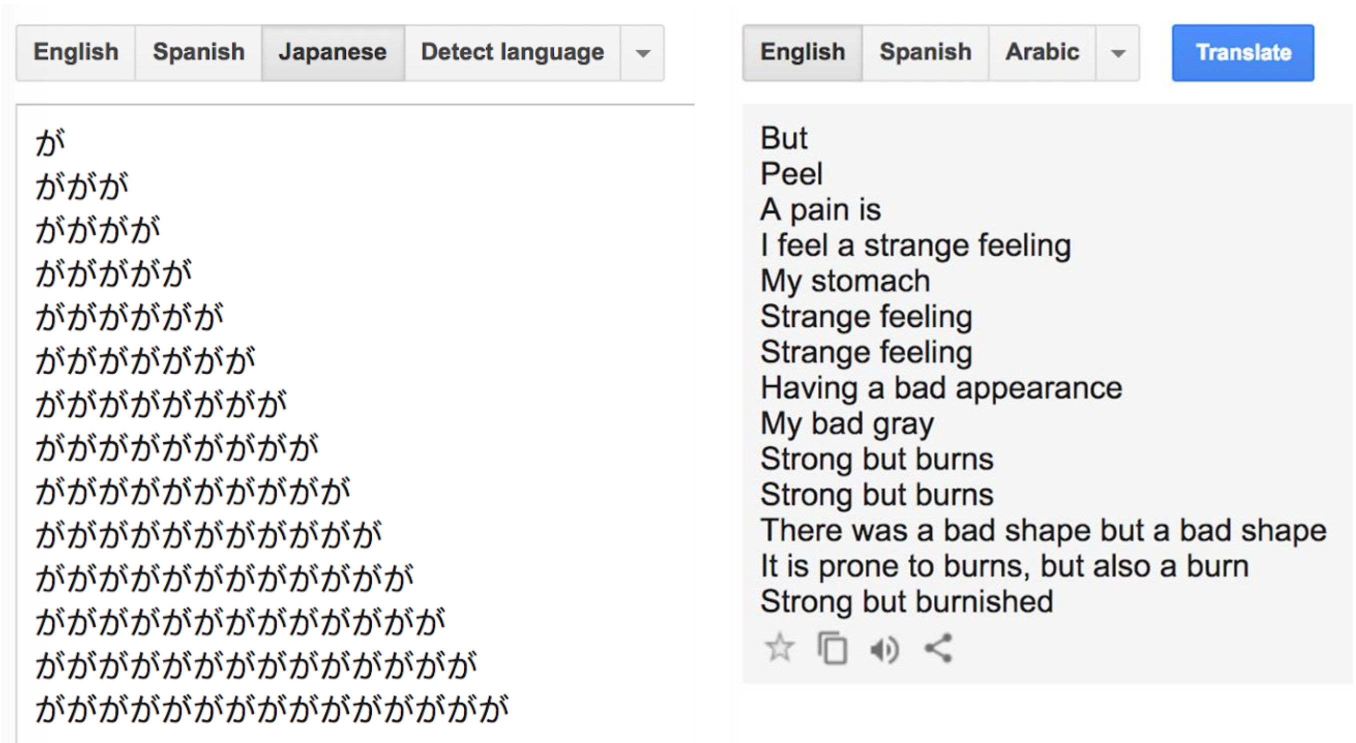
Table 5: Portuguese→Spanish BLEU scores using various models.

	Model	BLEU
(a)	PBMT bridged	28.99
(b)	NMT bridged	30.91
(c)	NMT Pt→Es	31.50
(d)	Model 1 (Pt→En, En→Es)	21.62
(e)	Model 2 (En↔{Es, Pt})	24.75
(f)	Model 2 + incremental training	31.77

- **Train**:
  - Portuguese → English, English → Spanish (Model 1)
  - Or, English ←→ {Portuguese, Spanish} (Model 2)
- **Test**:
  - Portuguese → Spanish **Zero-Shot!**

Vậy là MT đã được giải quyết chưa?

- Nope!
- Uninterpretable systems do strange things



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

Seq2seq là rất linh hoạt!

- 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)

Kết luận

- 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!