Seq2seq model
and application for machine translation

Tran Hong Viet

UET, VNU-Hanoi

Content

- Introduction to Machine Translation
- The seq2seq model
- Attention mechanism
- **Practice:** Machine translation với mô hình seq2seq

Machine Translation
(Dịch máy)

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

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

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

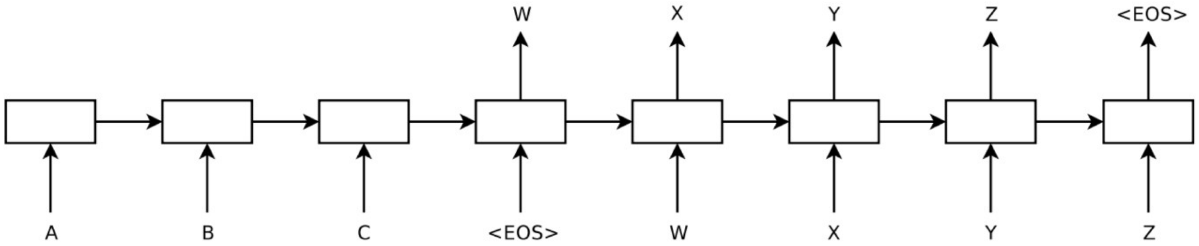
What is Neural Machine Translation?

- Neural Machine Translation (NMT) is a way to do Machine Translation with a *single neural network*

- The neural network architecture is called sequence-to-sequence (aka seq2seq) and it involves *two* RNNs (LSTMs).

Encoder-decoder Framework

Decoder (RNN/LSTM)



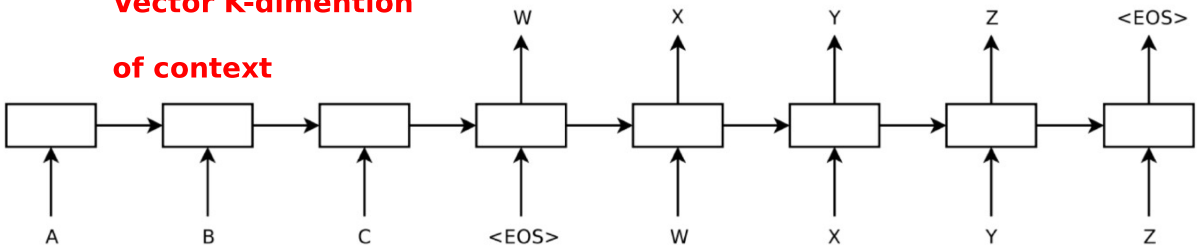
Encoder (RNN/LSTM)

"Sequence to Sequence Learning with Neural Networks", 2014

Encoder-decoder Framework

Vector K-dimention

of context

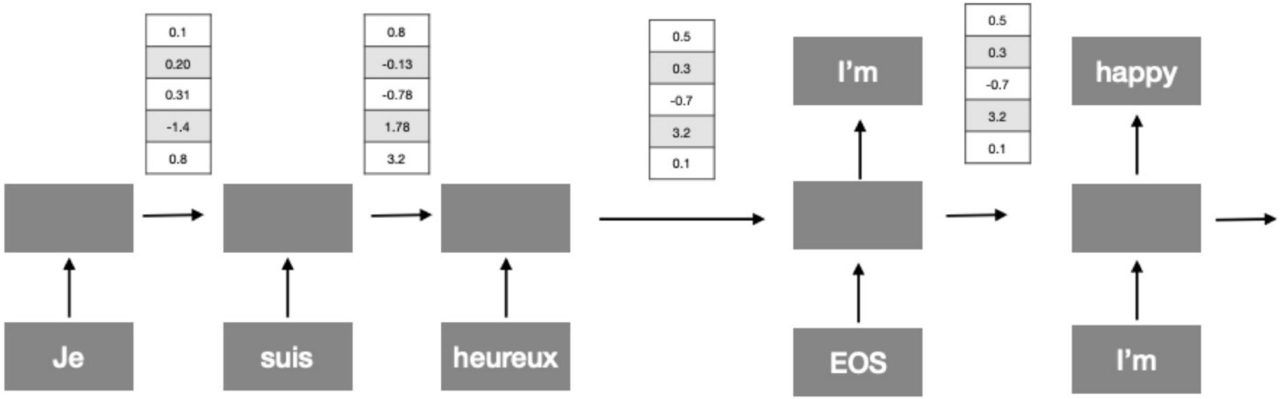


Condition of the word to be generate from

the translation system

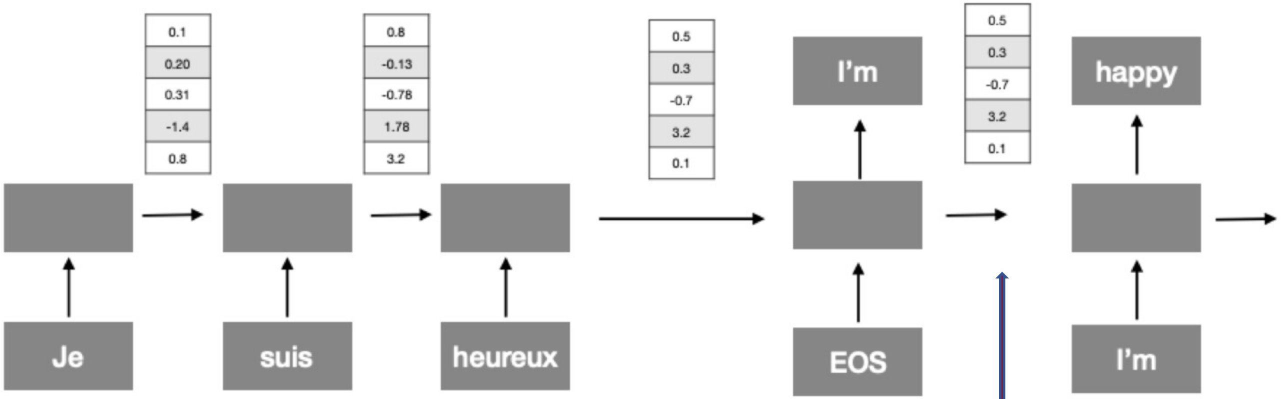
“Sequence to Sequence Learning with Neural Networks”, 2014

Encoder-decoder Framework



Encoder-decoder Framework

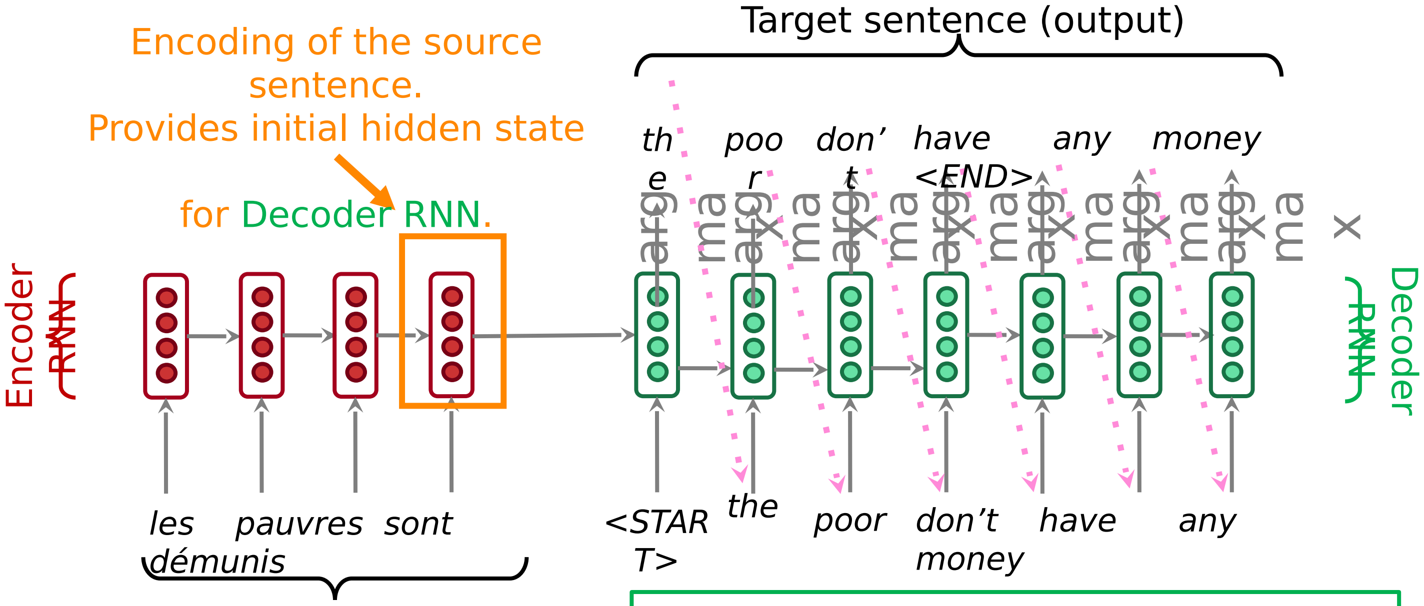
All inputs are aggregatedin this single vector



In the seq2seq, the decoder's state depends only on the previous state and the previous output

Neural Machine Translation

The sequence-to-sequence model



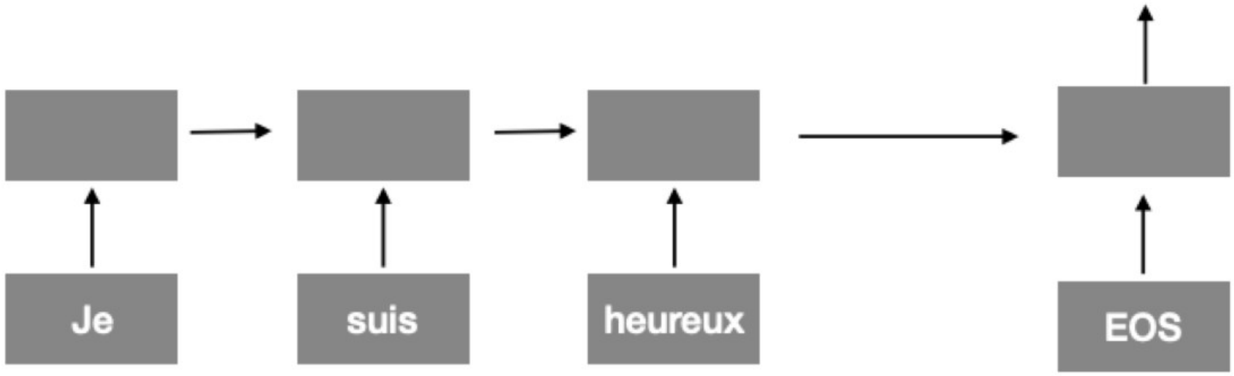
Decoder RNN is a Language Model that generates target sentence conditioned on encoding.

Encoder RNN produces an encoding of the source sentence.

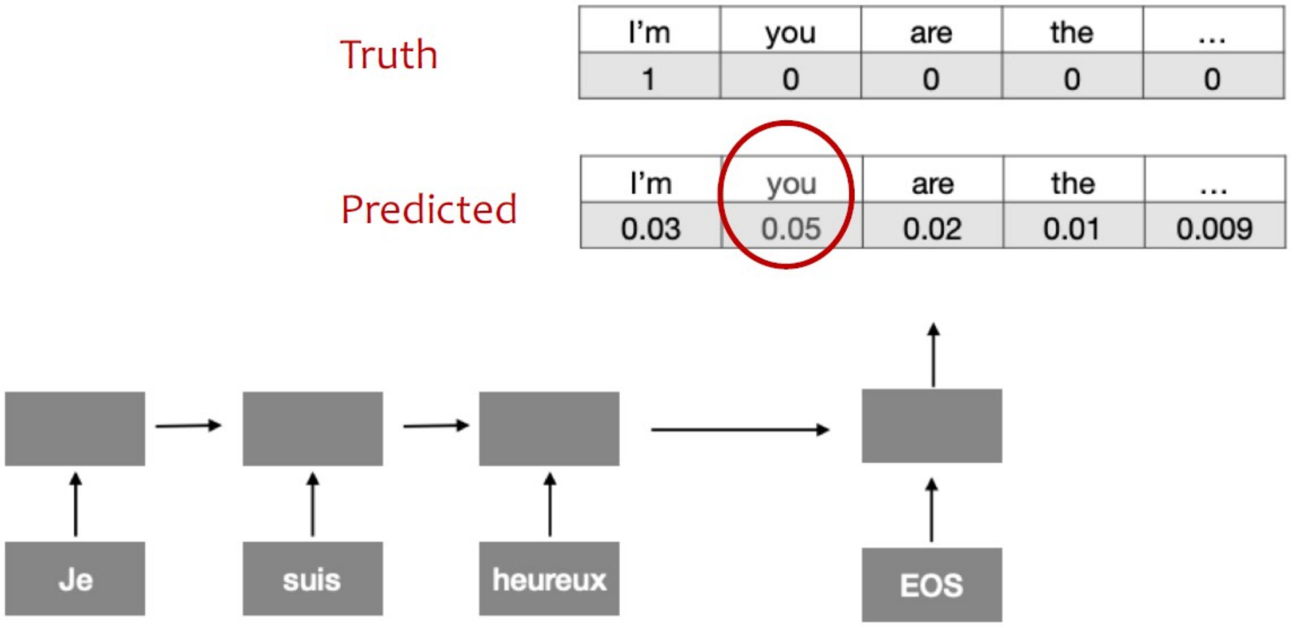
Note: This diagram shows **test time** behavior: decoder output is fed in as next step's input

Training of NMT system

- As in other RNN models, we can train by minimizing the loss function between what we predict at each step and its ground true value.



Training of NMT system

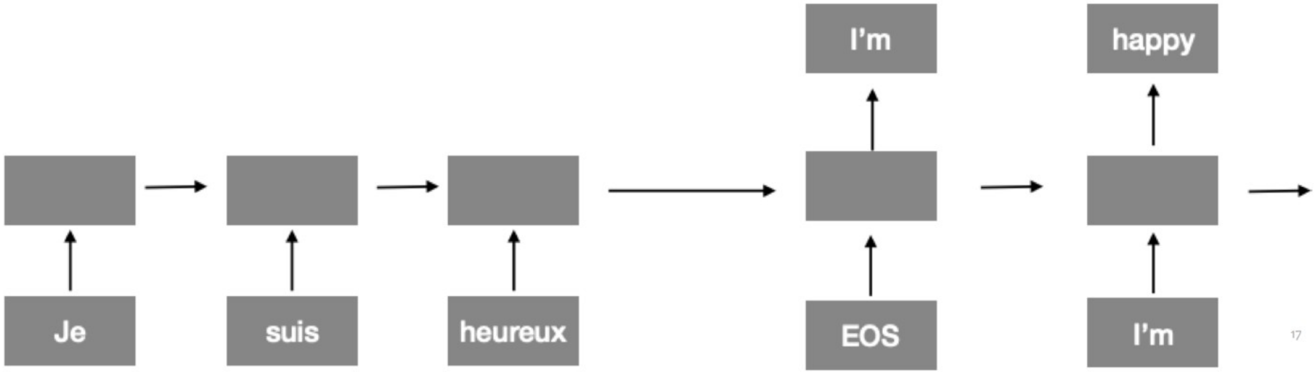


Truth

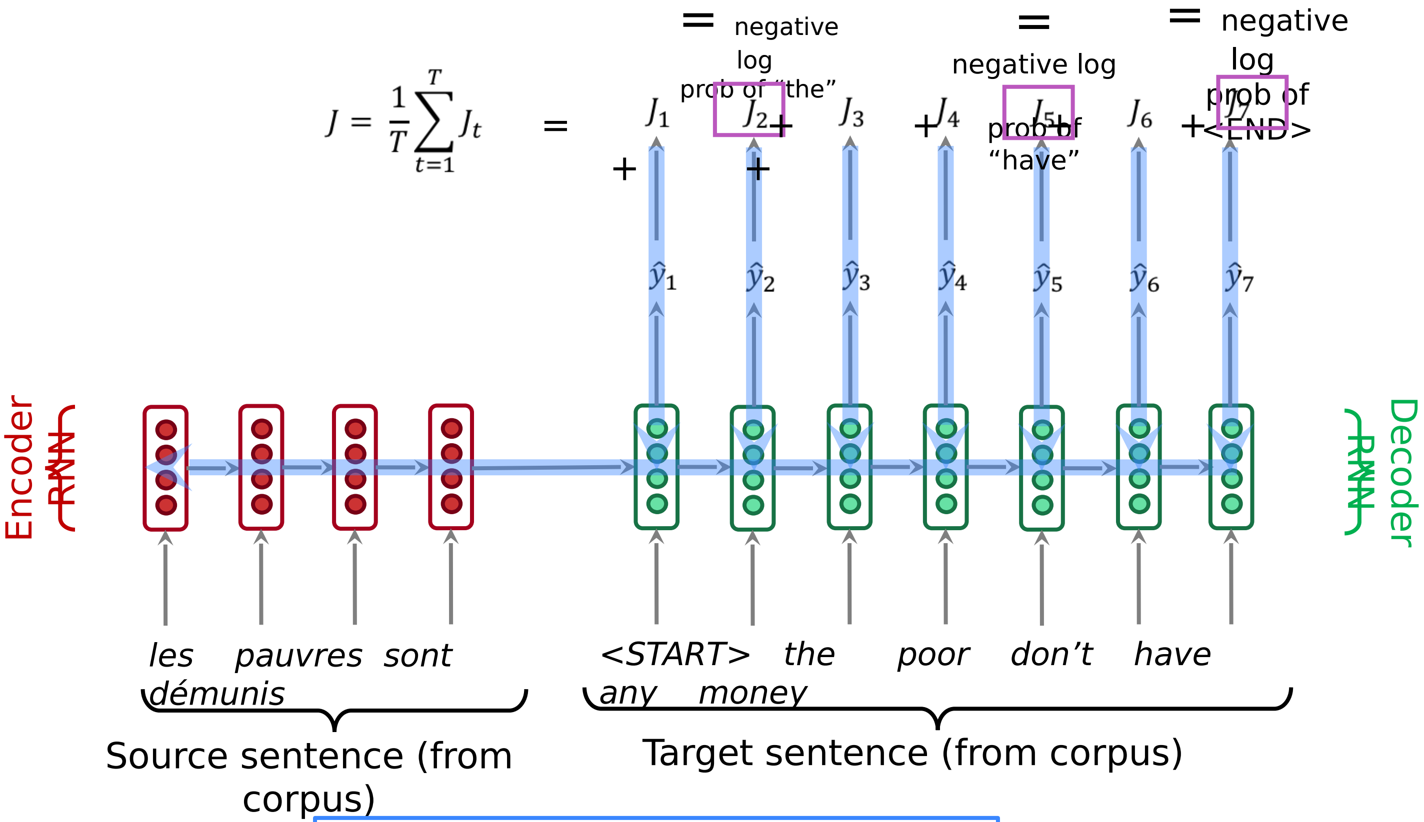
happy	great	bad	ok	...
1	0	0	0	0

Predicted

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



system



Seq2seq is optimized as a **single system.**
Backpropagation operates "*end to*

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

Decoder based on 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!

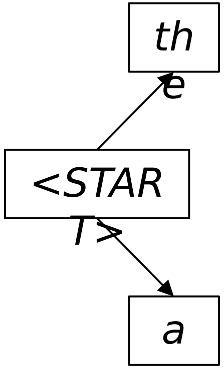
Beam search: Example

Beam size = 2

<START>

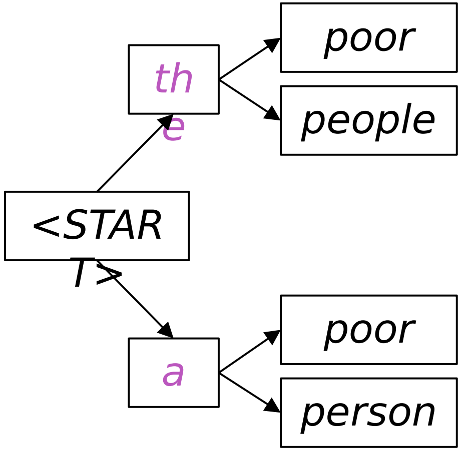
Beam search: Example

Beam size = 2



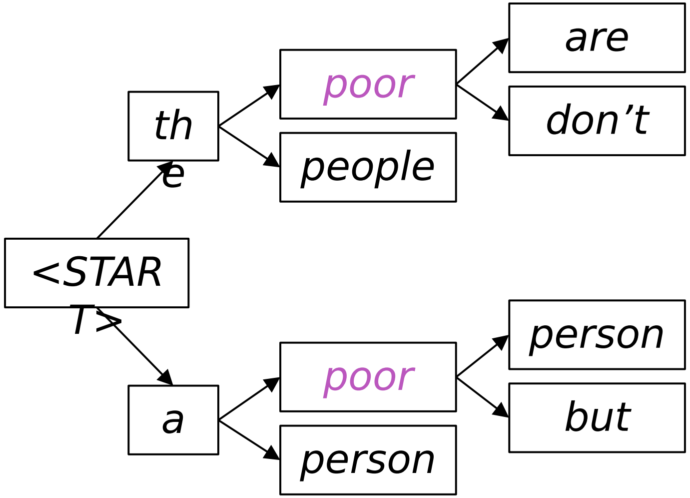
Beam search: Example

Beam size = 2



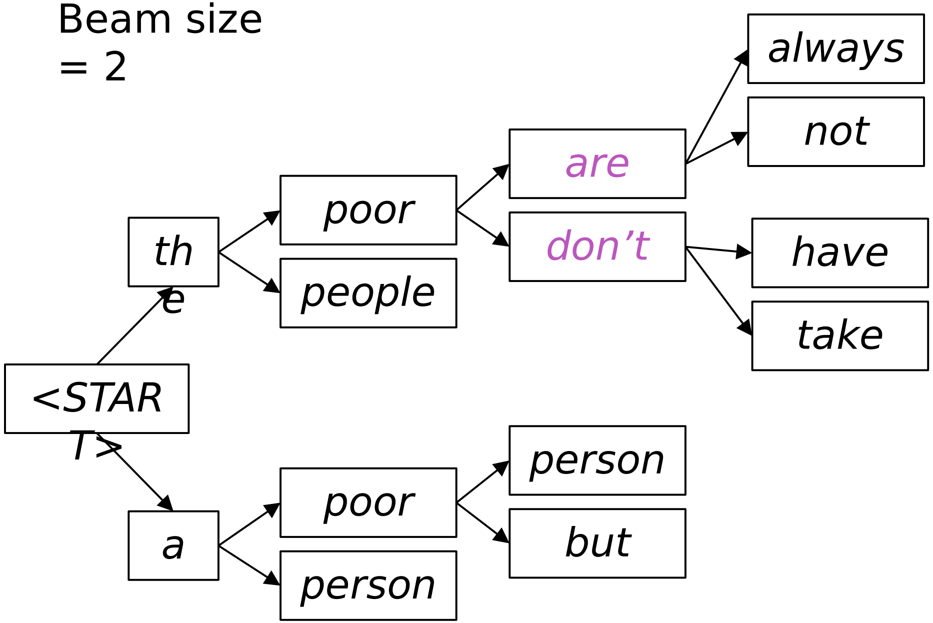
Beam search: Example

Beam size = 2



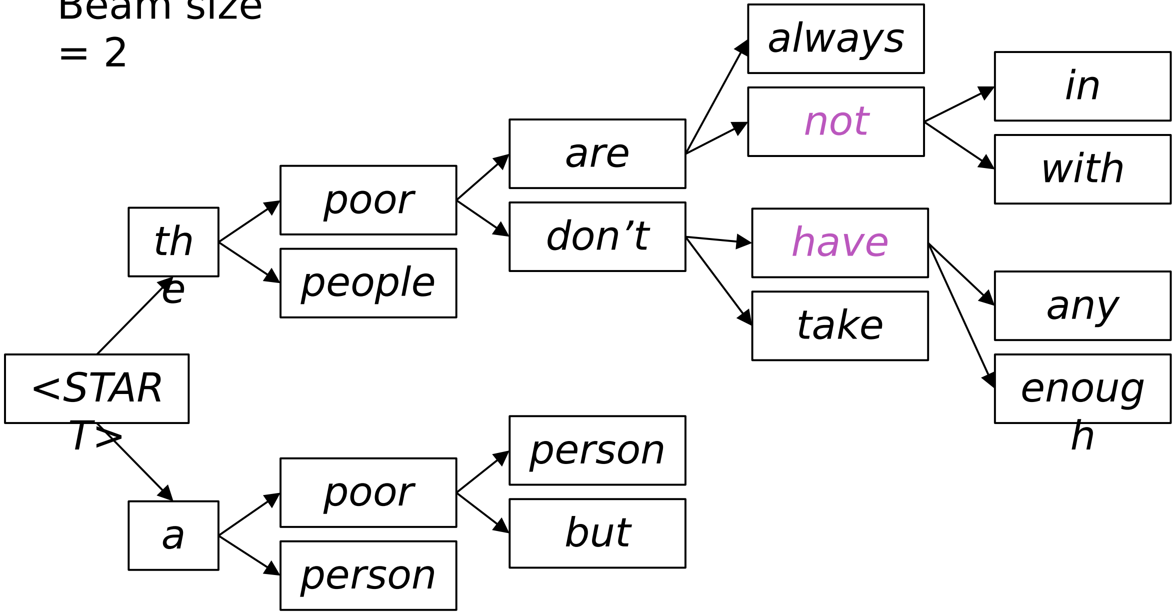
Beam search: Example

Beam size = 2



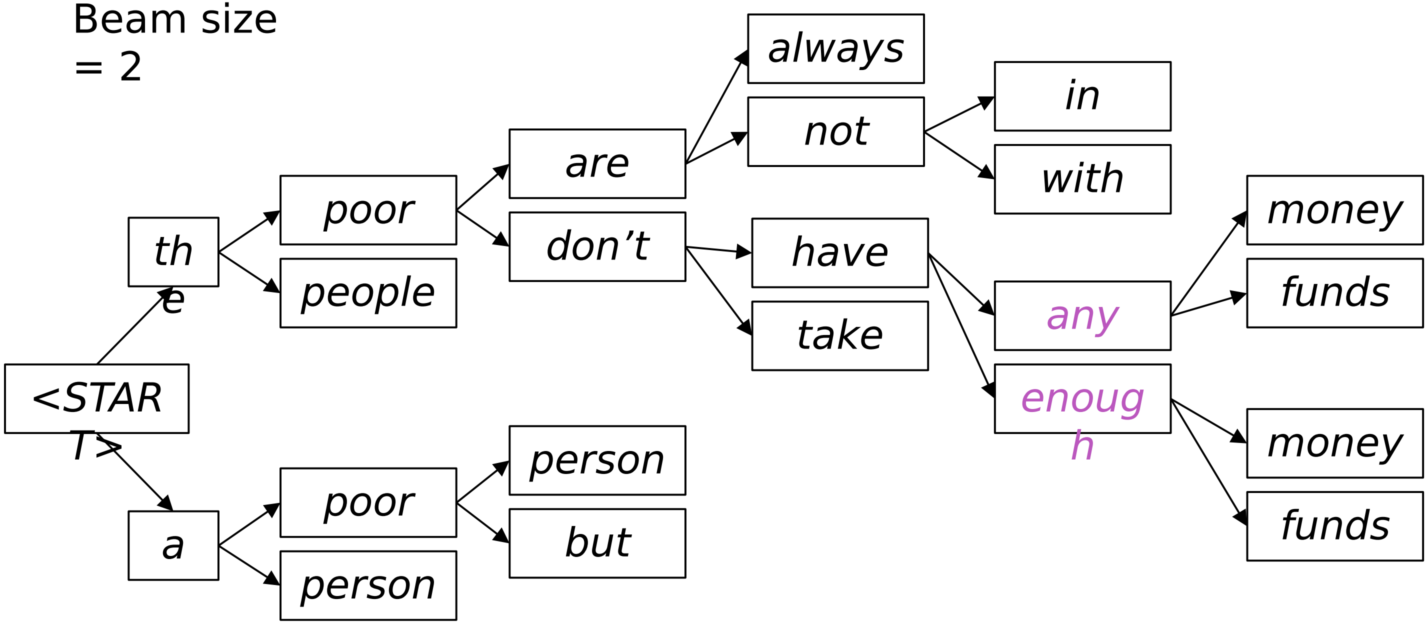
Beam search: Example

Beam size = 2



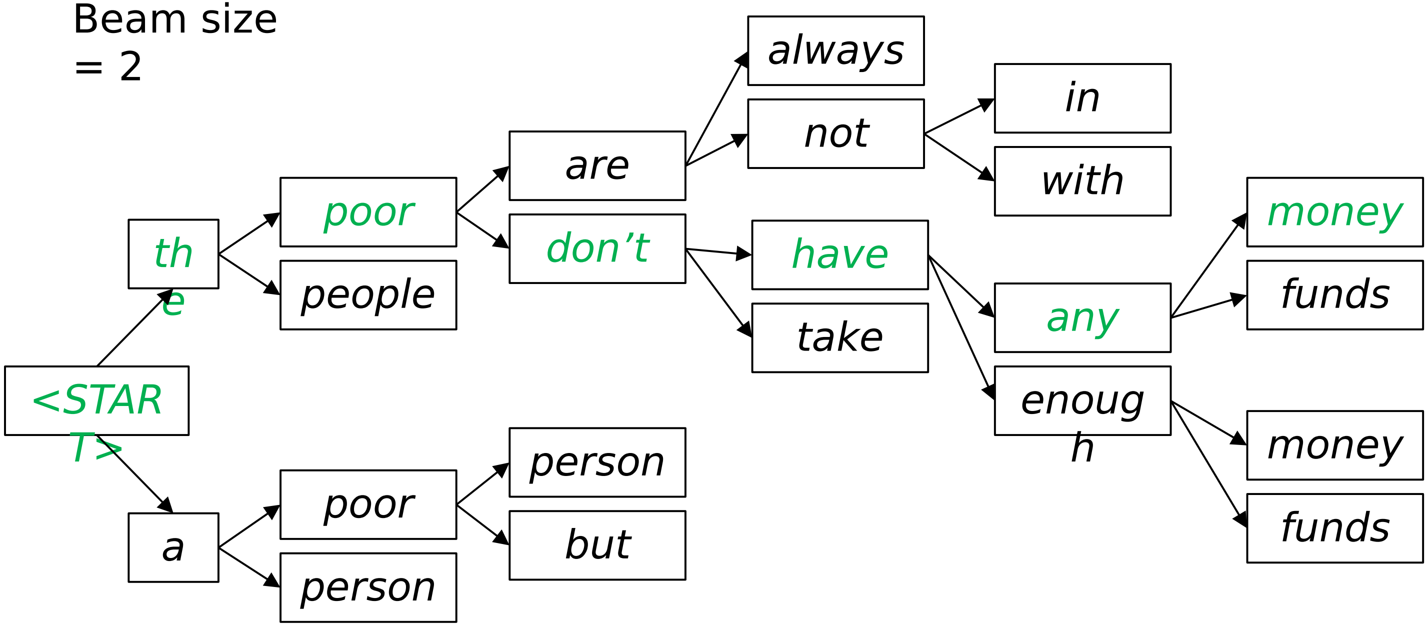
Beam search: Example

Beam size = 2



Beam search: Example

Beam size = 2



Beam search: stopping criterion

- In greedy decoding, usually we decode until the model produces an <END>

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

- In beam search decoding, different hypotheses may produce <END> tokens on different timesteps
  - When a hypothesis produces <END>, that hypothesis is **complete**.
  - **Place it aside** and continue exploring other hypotheses via beam search.
- Usually we continue beam search until:
  - We reach timestep T (where T is some pre-defined cutoff), or
  - We have at least n completed hypotheses (where n is pre-defined cutoff)

Advantage 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

Weakness of NMT?

Compared to SMT:

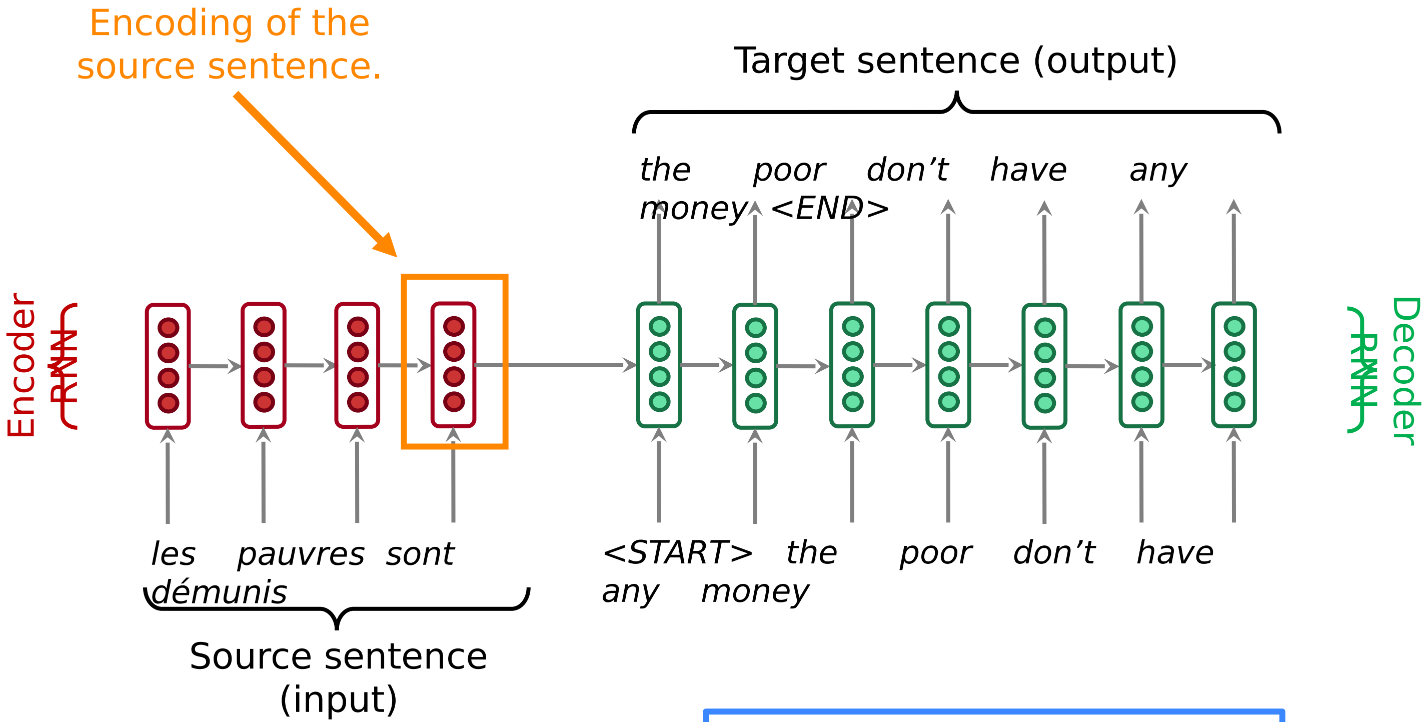
- NMT is less interpretable
  - Hard to debug
- NMT is difficult to control
  - For example, can’t easily specify rules or guidelines for translation
  - Safety concerns!

How to evaluation MT system?

BLEU (**Bi**lingual **E**valuation **U**nderstudy)

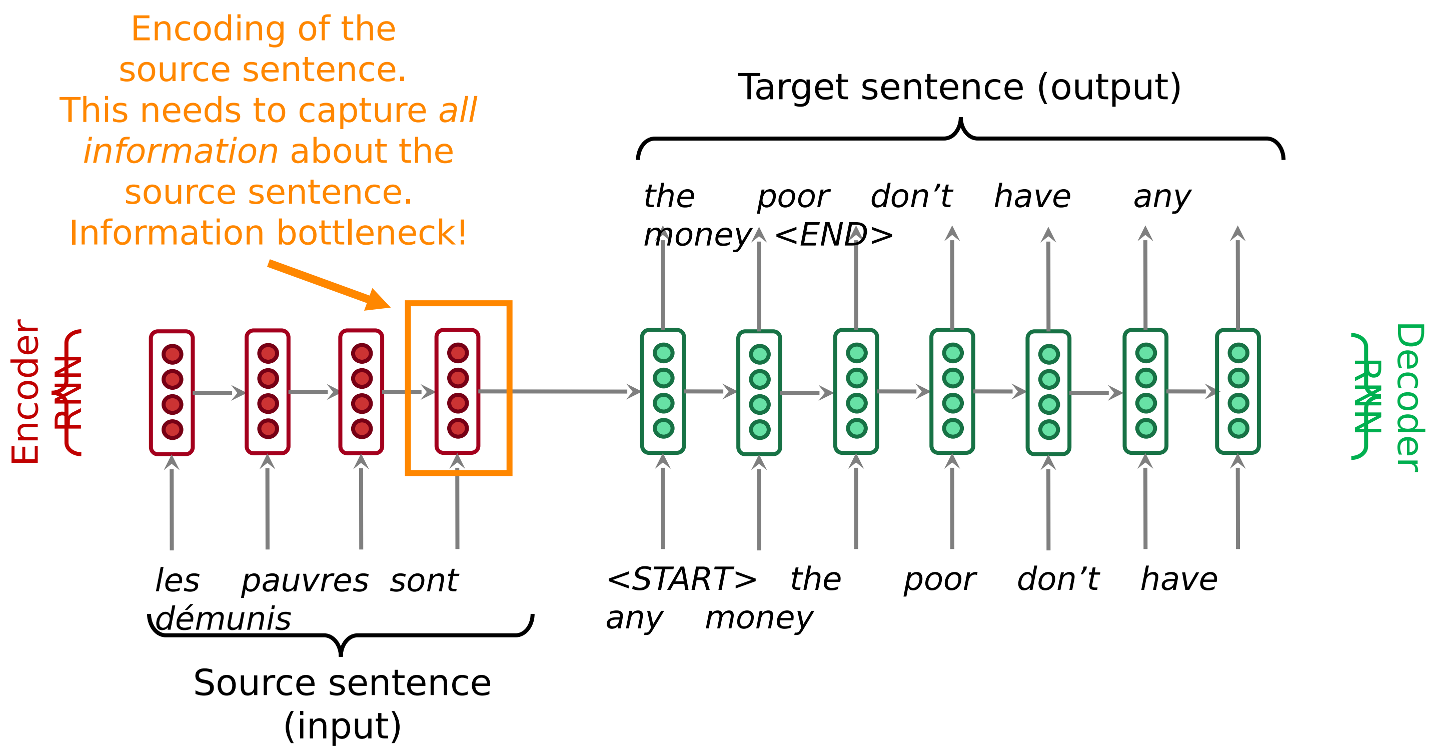
- BLEU compares the machine-written translation to one or several human-written translation(s), and computes a similarity score based on:
  - *n*-gram precision (usually up to 3 or 4-grams)
  - Penalty for too-short system translations
- BLEU is useful but imperfect
  - There are many valid ways to translate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

Sequence-to-sequence: the bottleneck problem

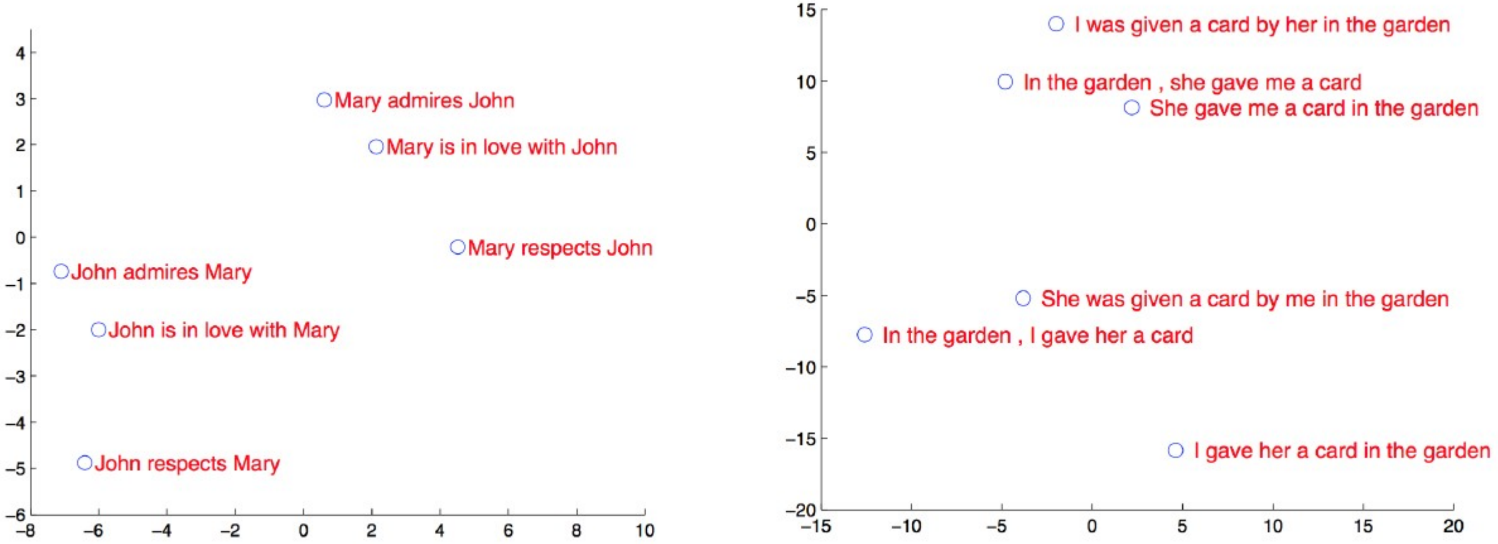


Problems with this architecture?

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)

Watching sentence embedding



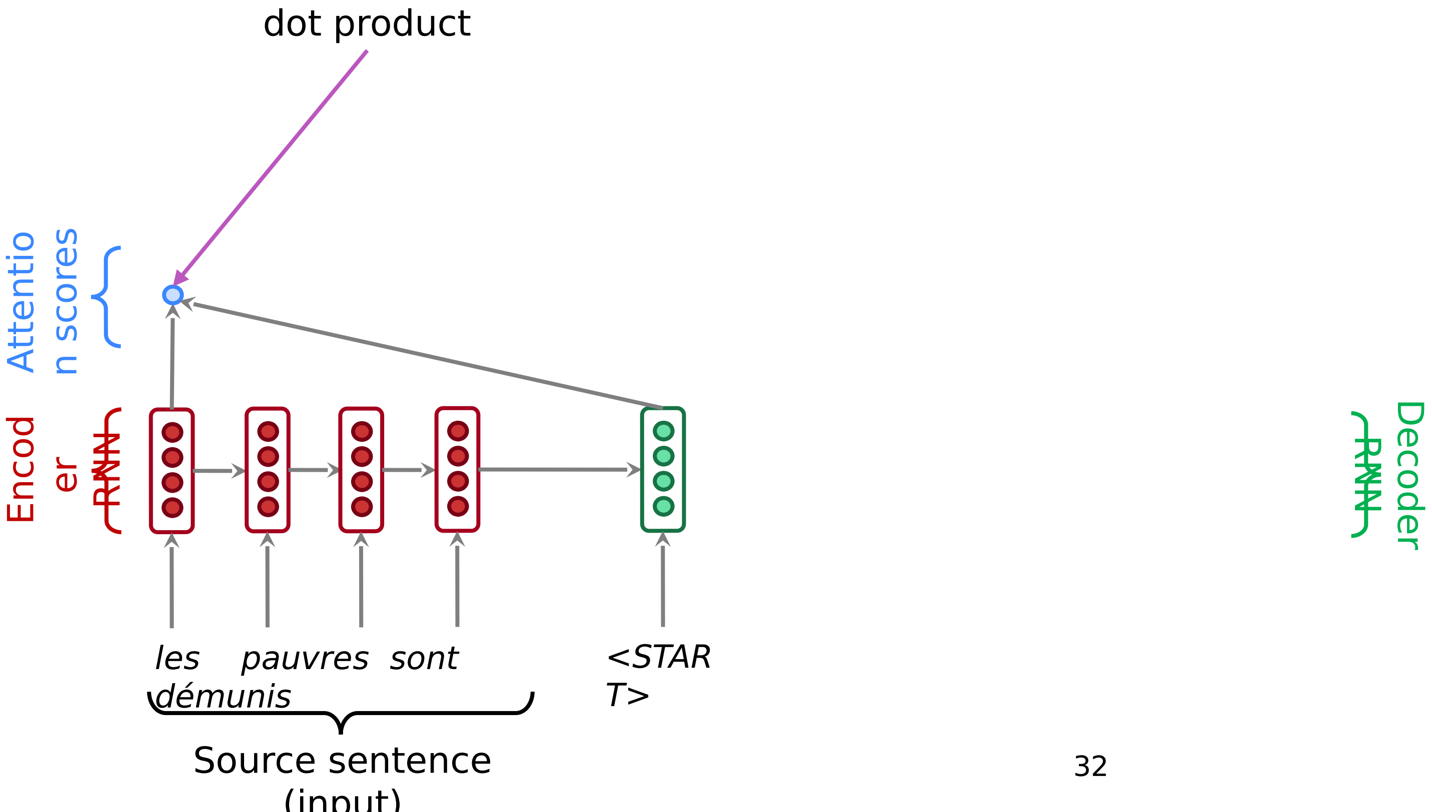
Sutskever et al, NIPS 2014, https://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf

Attention (Chú ý)

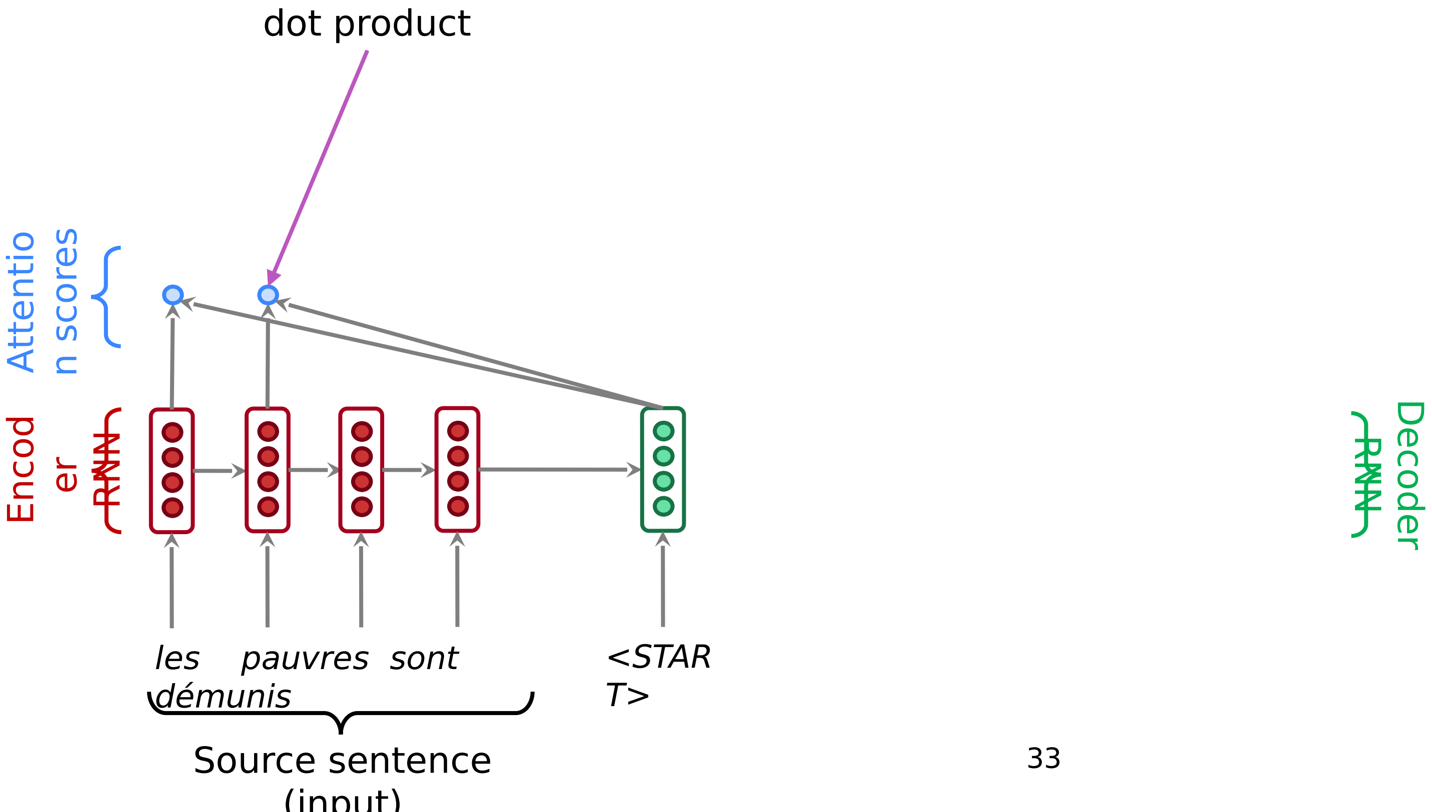
- **Attention: Solution** for bottleneck problem.

- **<u>Main idea</u>:** on each step of the decoder, use ***direct connection to the encoder*** *to* ***focus on a particular part*** of the source sequence

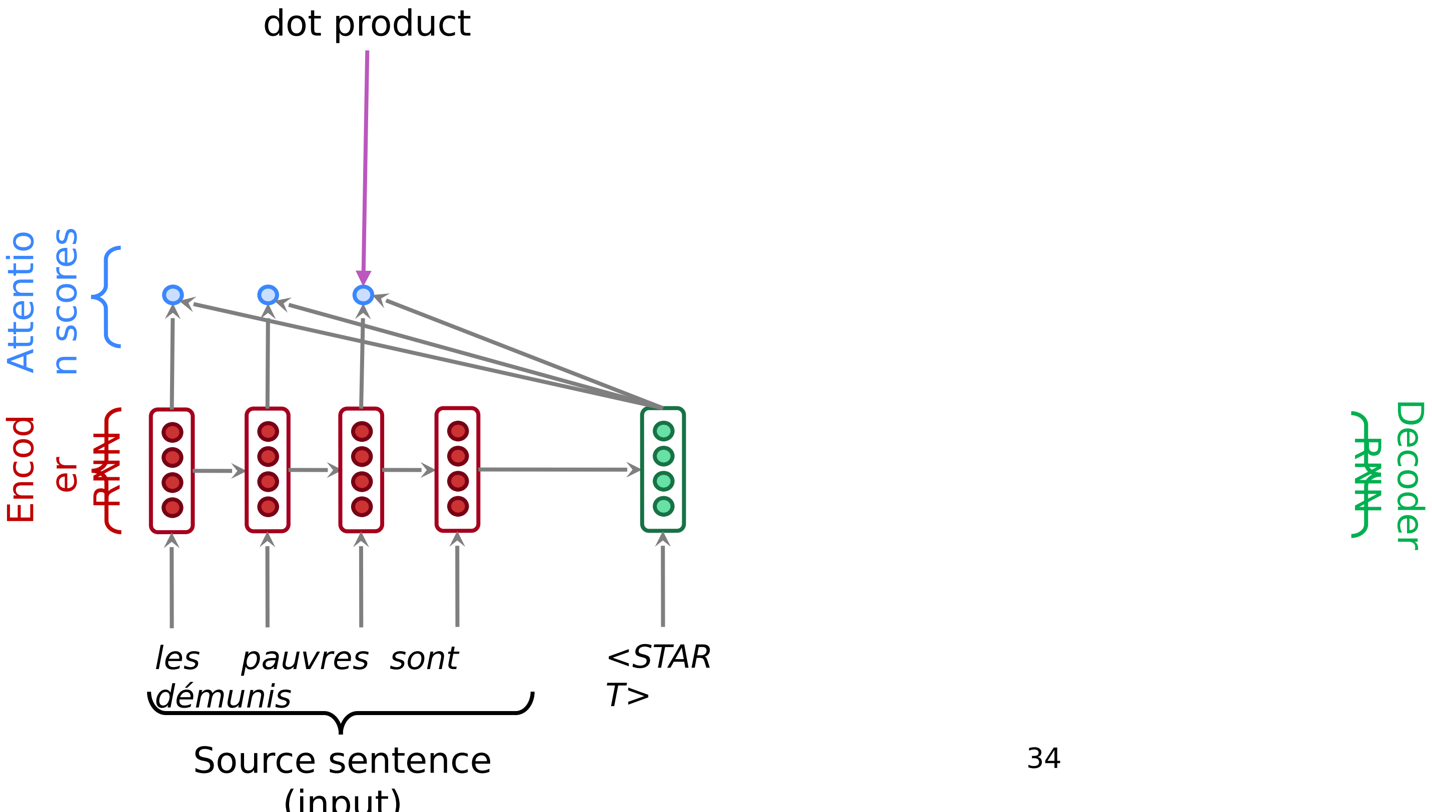
Encoder-Decoder với Attention



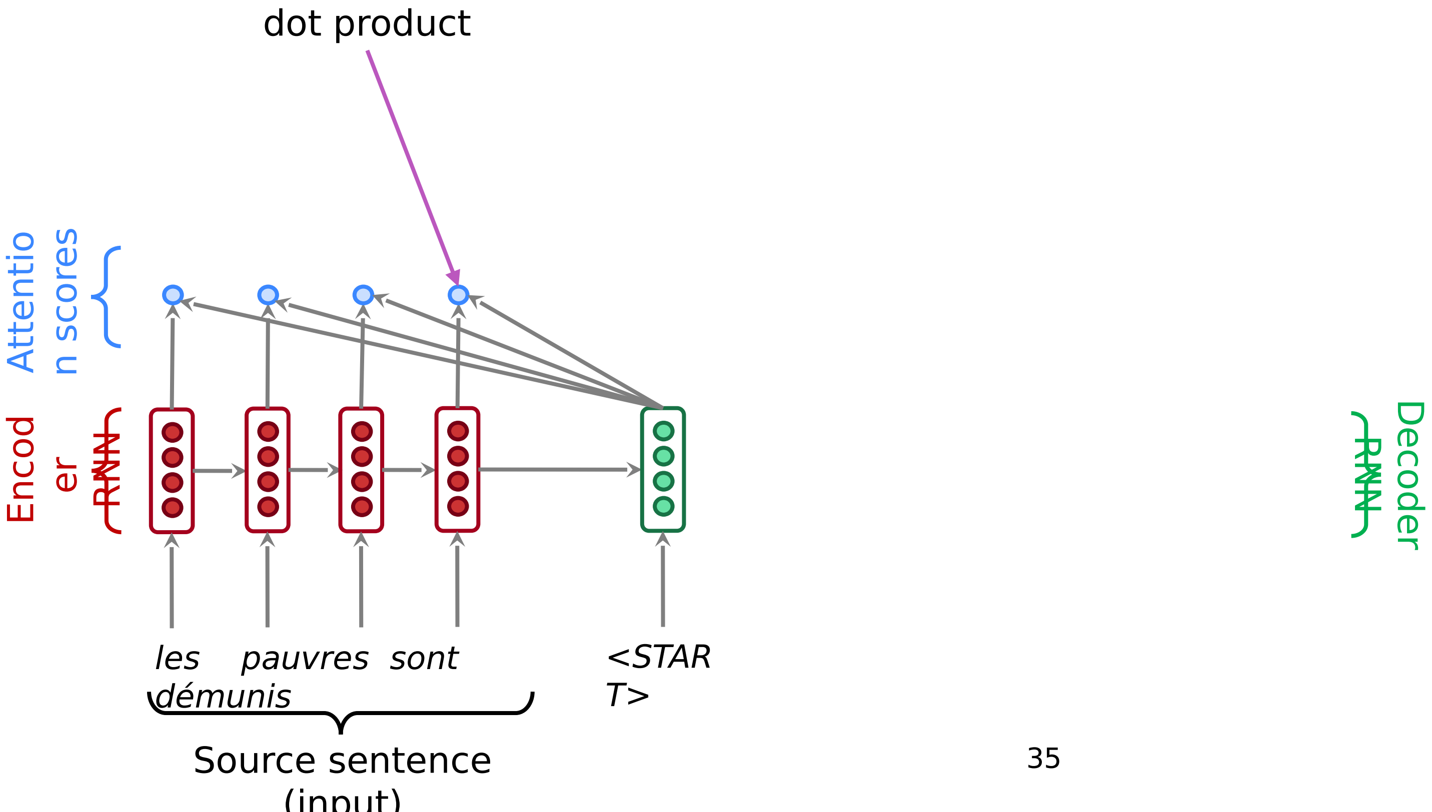
Seq2Seq with attention



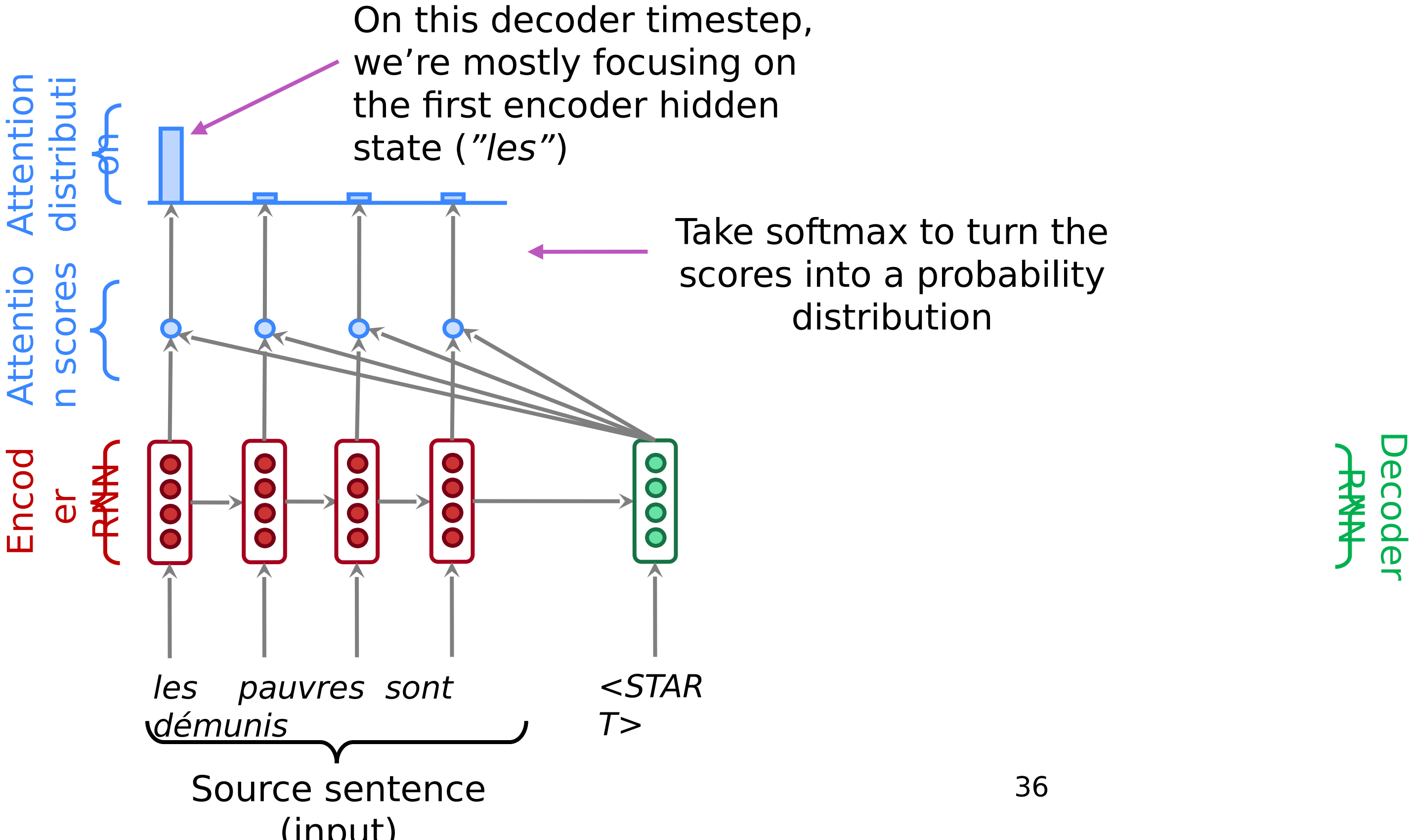
Sequence-to-sequence with attention



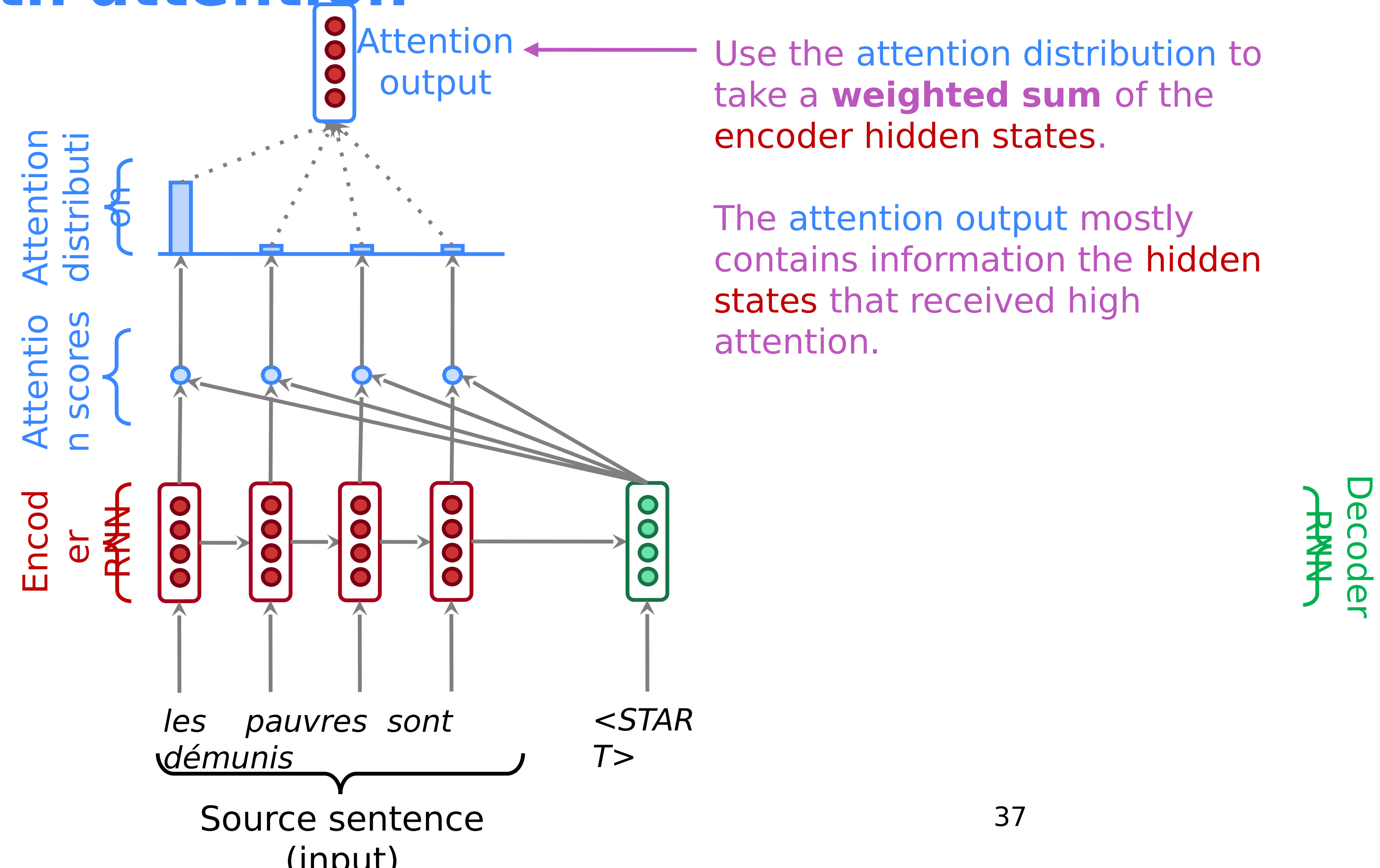
Sequence-to-sequence with attention



Sequence-to-sequence with attention



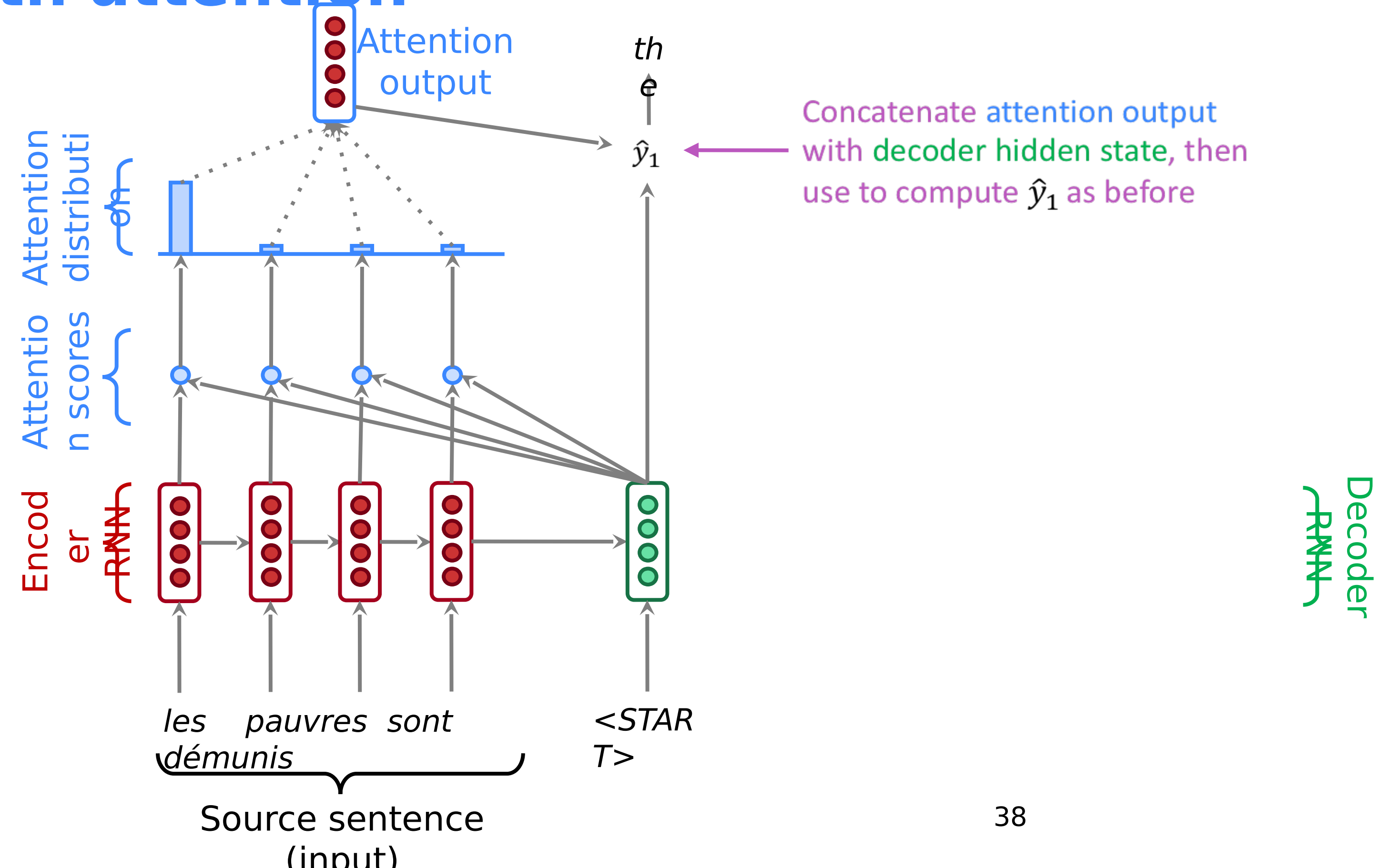
Sequence-to-sequence with attention



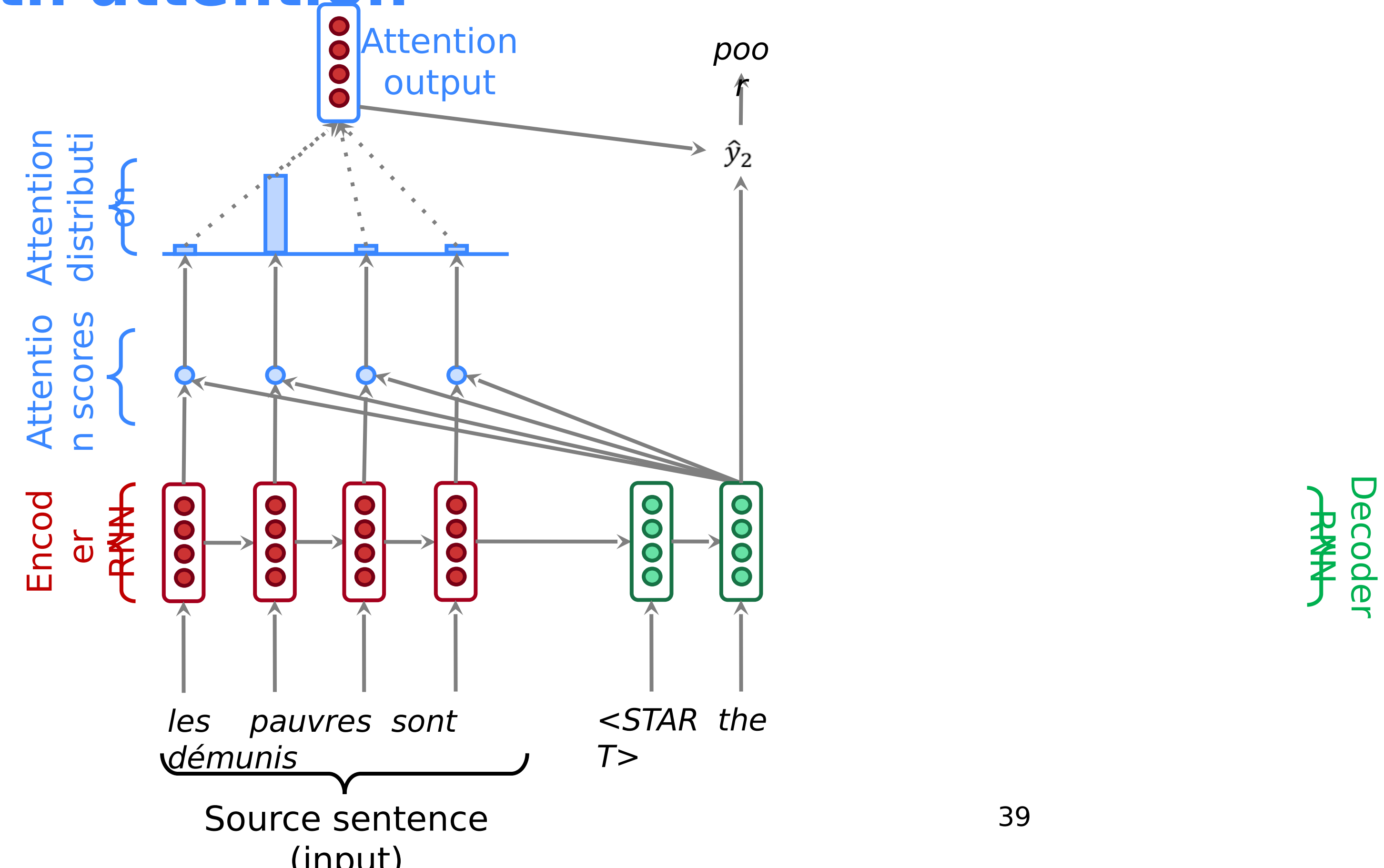
Use the attention distribution to take a **weighted sum** of the encoder hidden states.

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

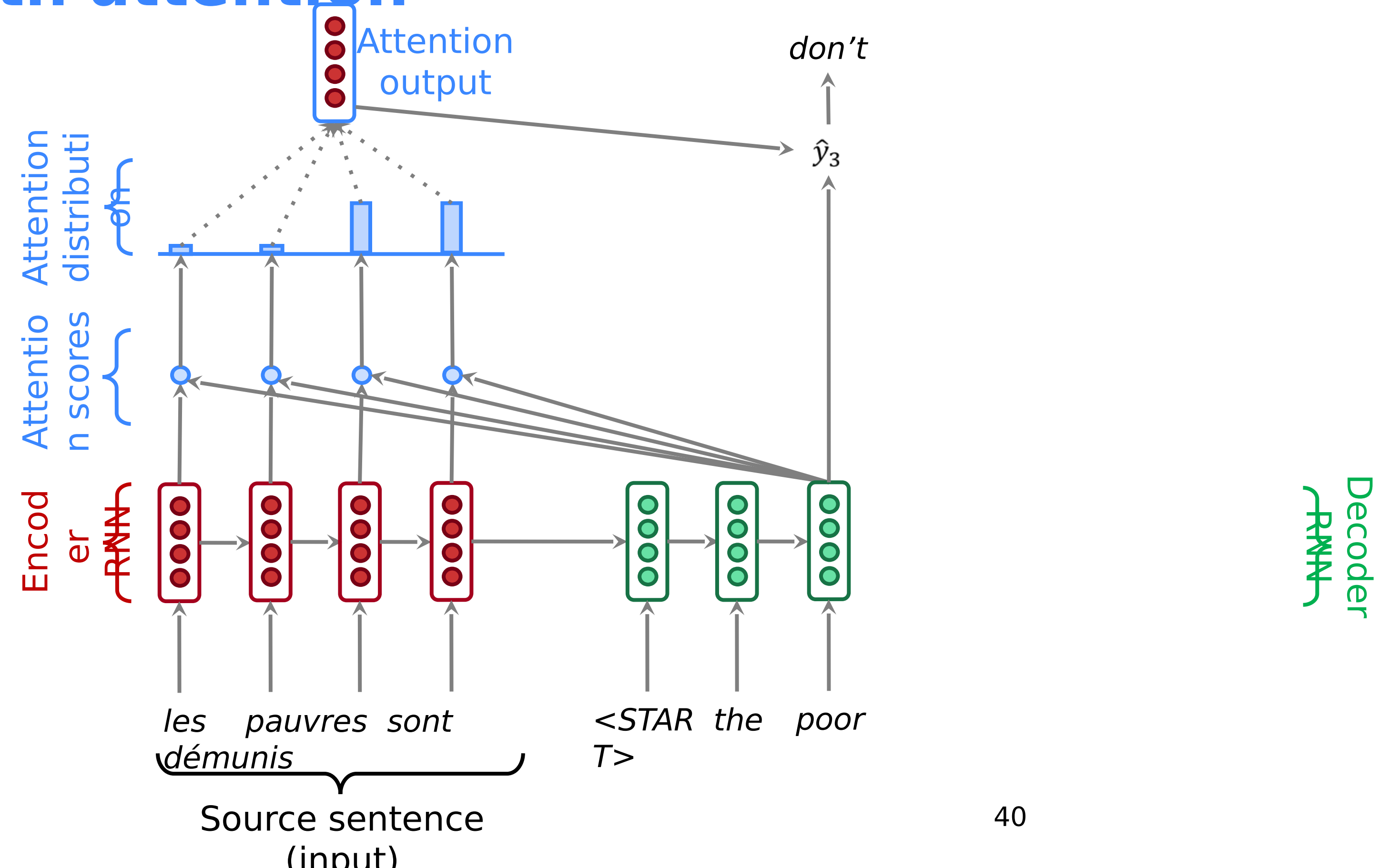
Sequence-to-sequence with attention



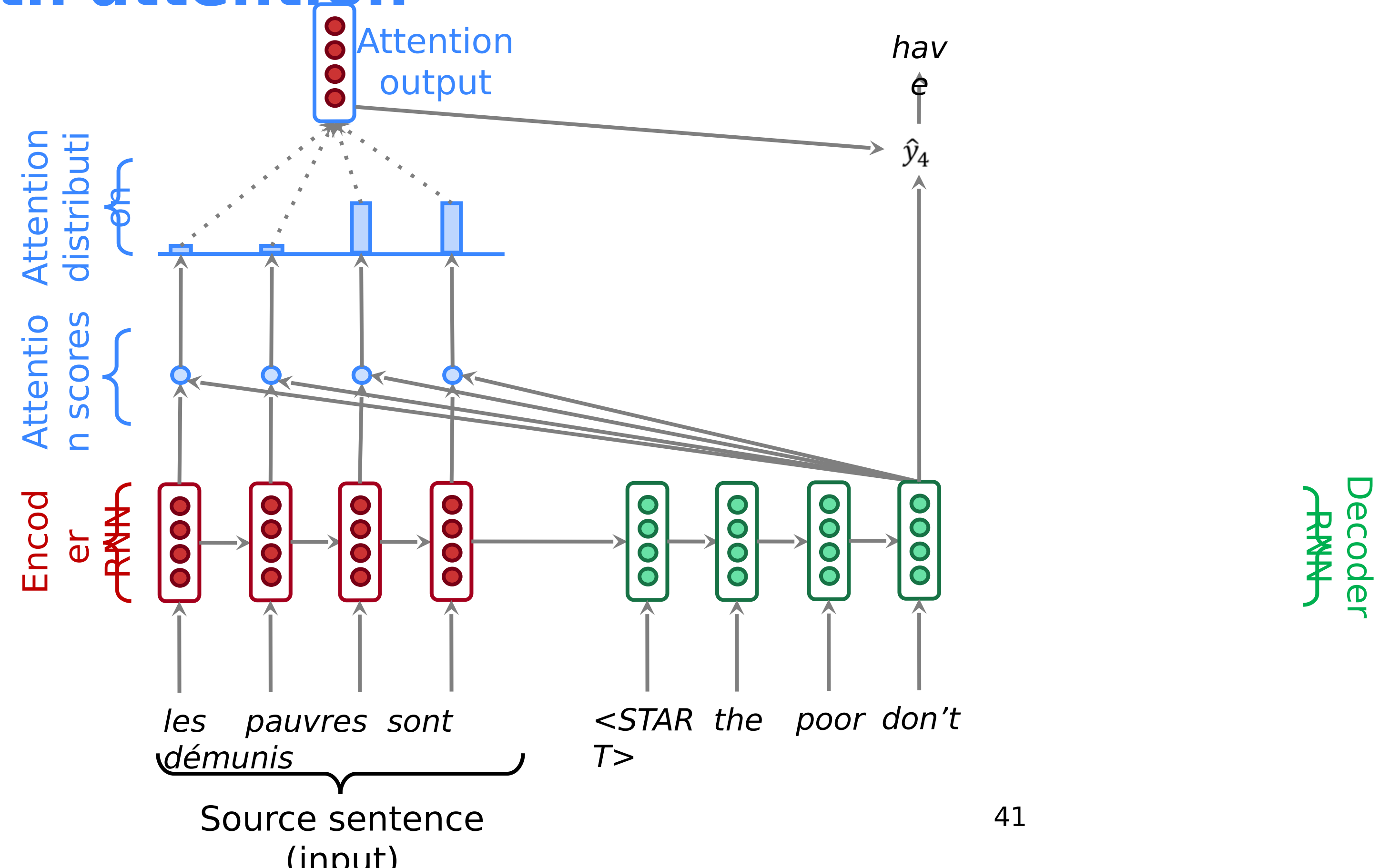
Sequence-to-sequence with attention



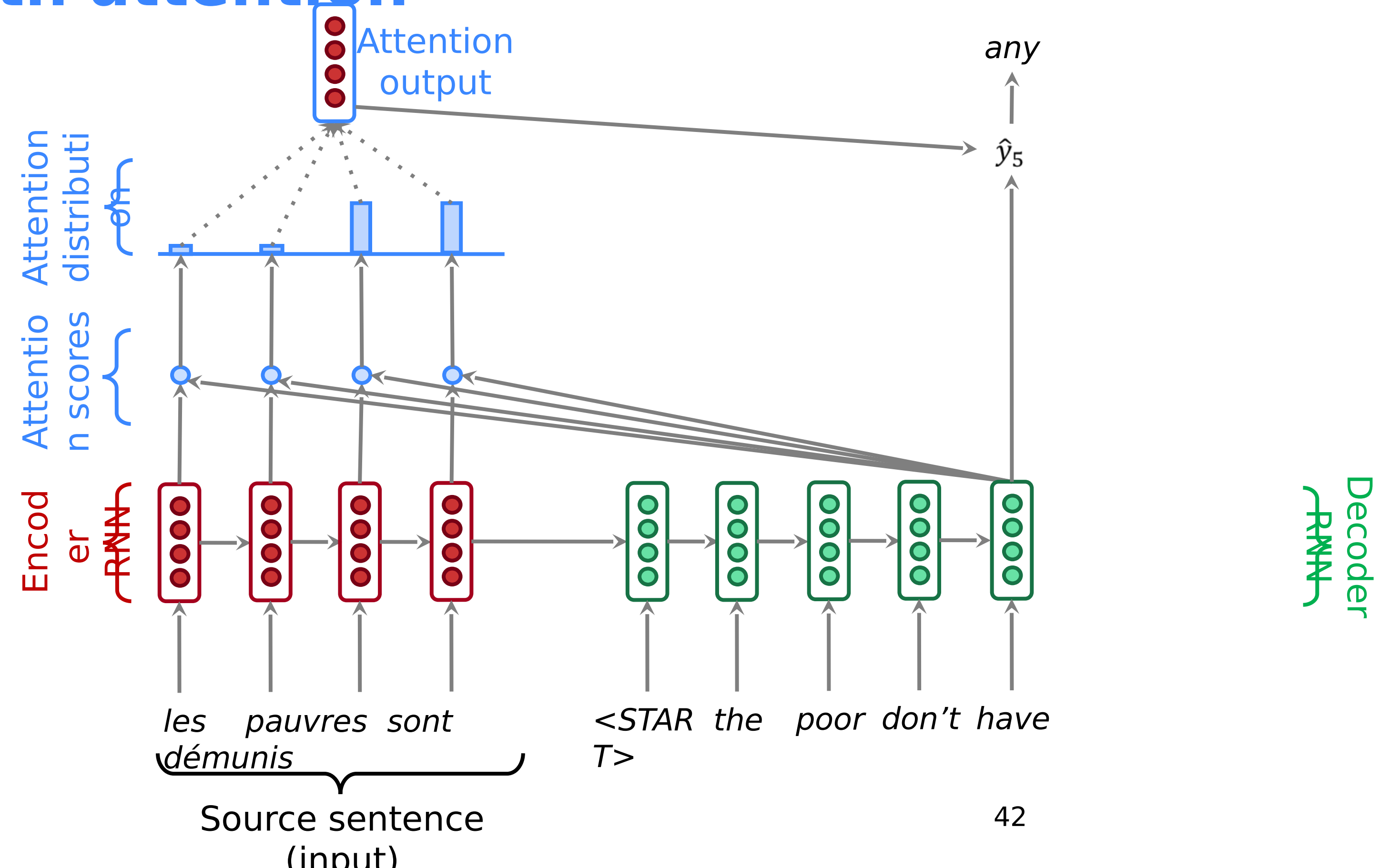
Sequence-to-sequence with attention



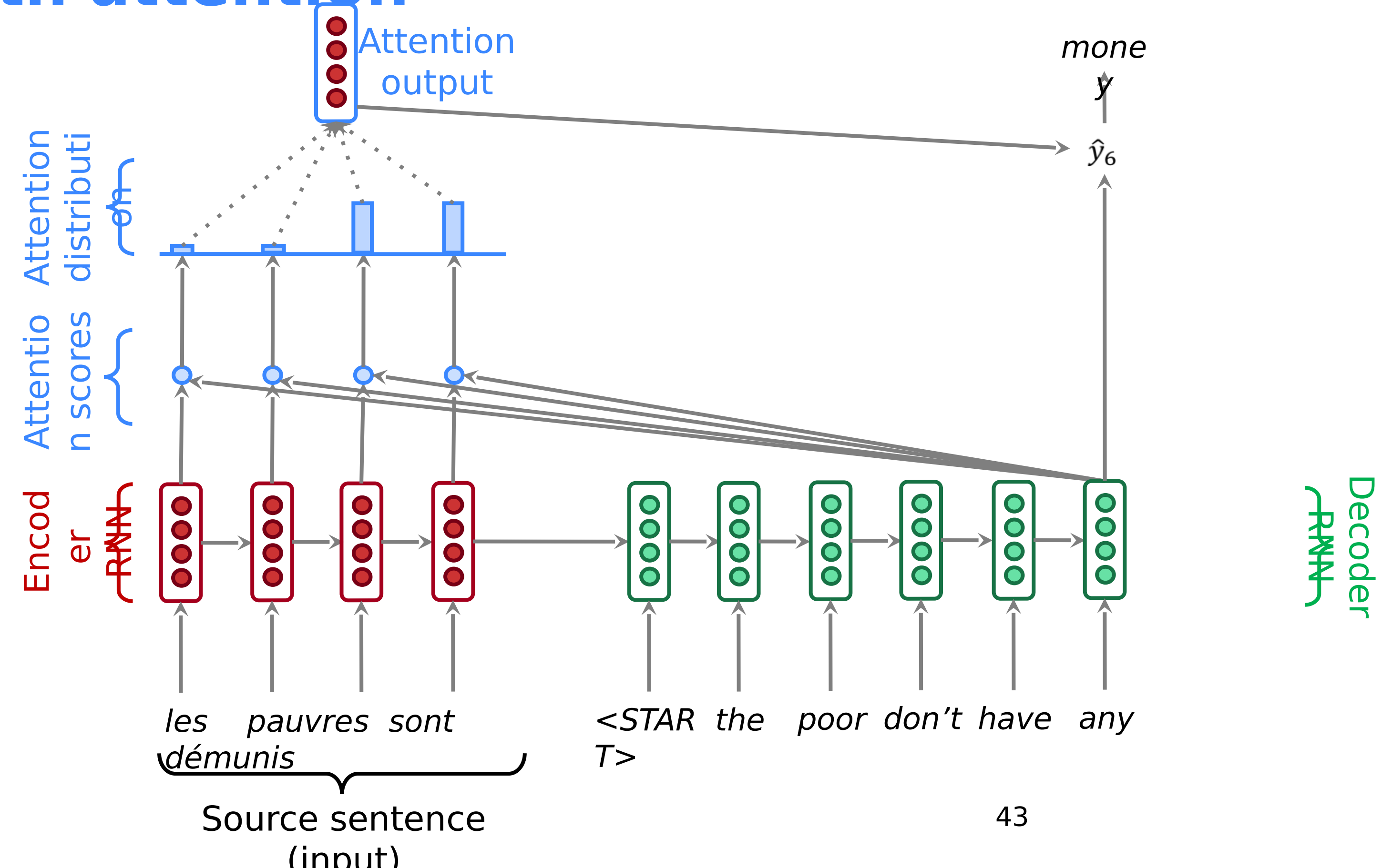
Sequence-to-sequence with attention



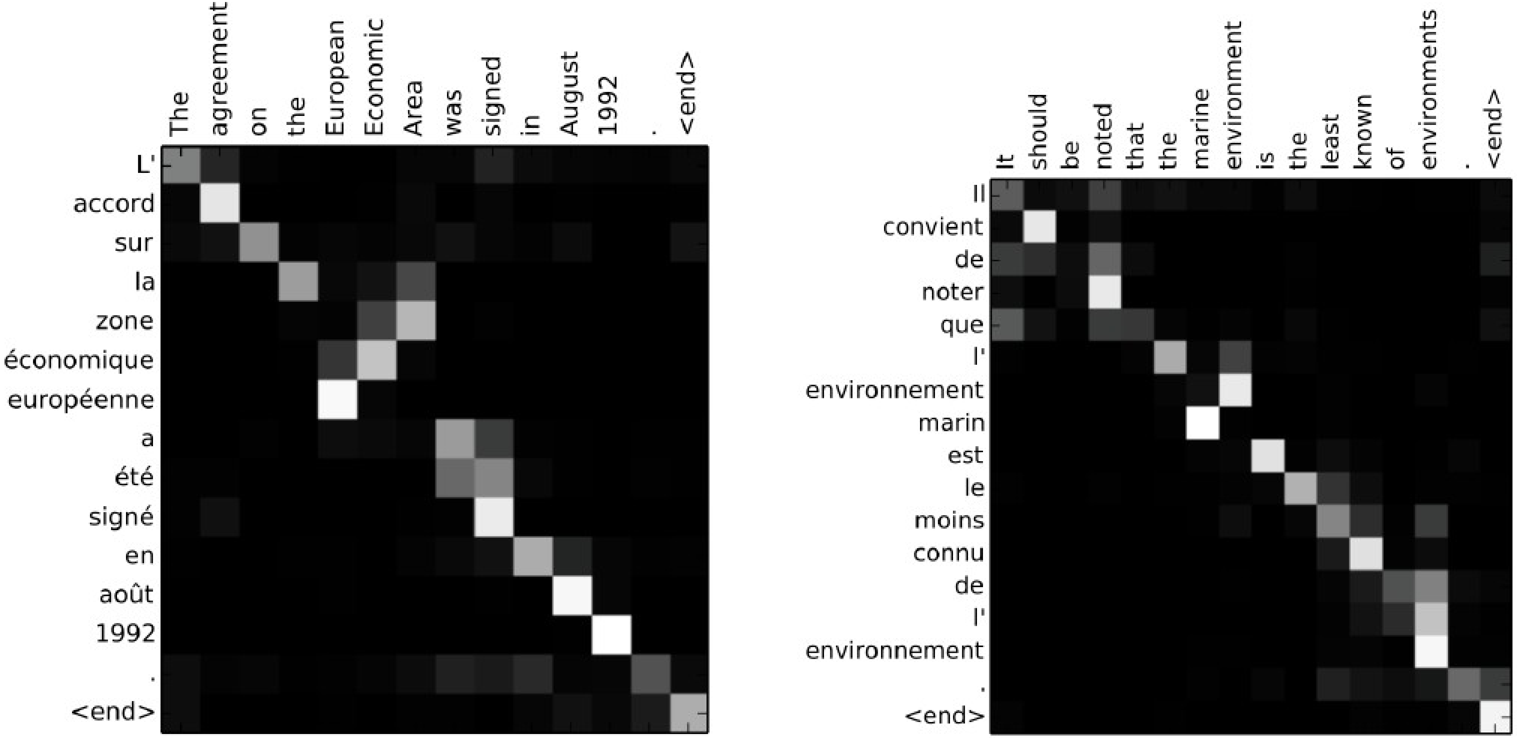
Sequence-to-sequence with attention



Sequence-to-sequence with attention

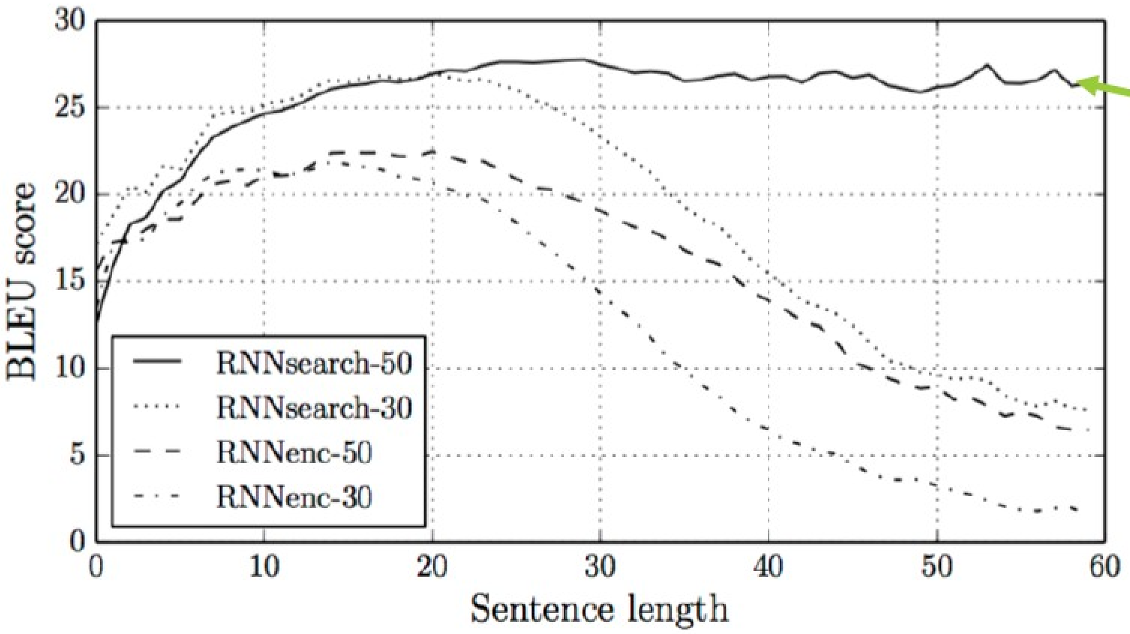


Neural Machine Translation by Jointly Learn to Align and Translate



Source: Bahdanau et al., ICLR 2015,
https://arxiv.org/abs/1409.0473

Neural Machine Translation by Jointly Learn to Align and Translate



No performance drop for long sentences!

Source: Bahdanau et al., ICLR 2015, https://arxiv.org/abs/1409.0473

Attention: **Formula**

- We have encoder hidden states ($h_1, \dots, h_N \in \mathbb{R}^h$
- On timestep *t*, we have (**query s**) decoder hidden $s_t \in \mathbb{R}^h$ tate
- We get the attention sc e^t es for th s_t step:

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

There are multiple ways to do this

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

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

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

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

- Finally we concatenate the attention $\boldsymbol{a}_t$ itput with the decoder hidden state and proceed a $[\boldsymbol{a}_t; \boldsymbol{s}_t] \in \mathbb{R}^{2h}$ -attention seq2seq model

Attention scoring function

- ***q*** is the query and ***k*** is the key
- **Multi-layer Perceptron** (Bahdanau et al. 2015)
  - Flexible, often very good with large data

$$a(\boldsymbol{q}, \boldsymbol{k}) = \boldsymbol{w}_2^\intercal \tanh(W_1[\boldsymbol{q}; \boldsymbol{k}])$$

- **Bilinear** (Luong et al. 2015)

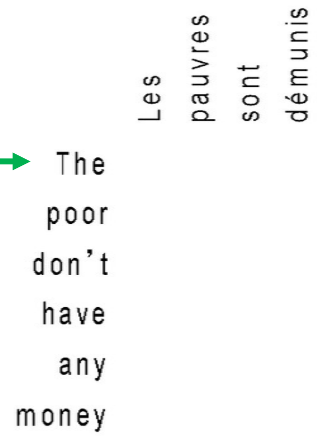
$$a(\boldsymbol{q}, \boldsymbol{k}) = \boldsymbol{q}^\intercal W \boldsymbol{k}$$

- **Dot Product** (Luong et al. 2015)

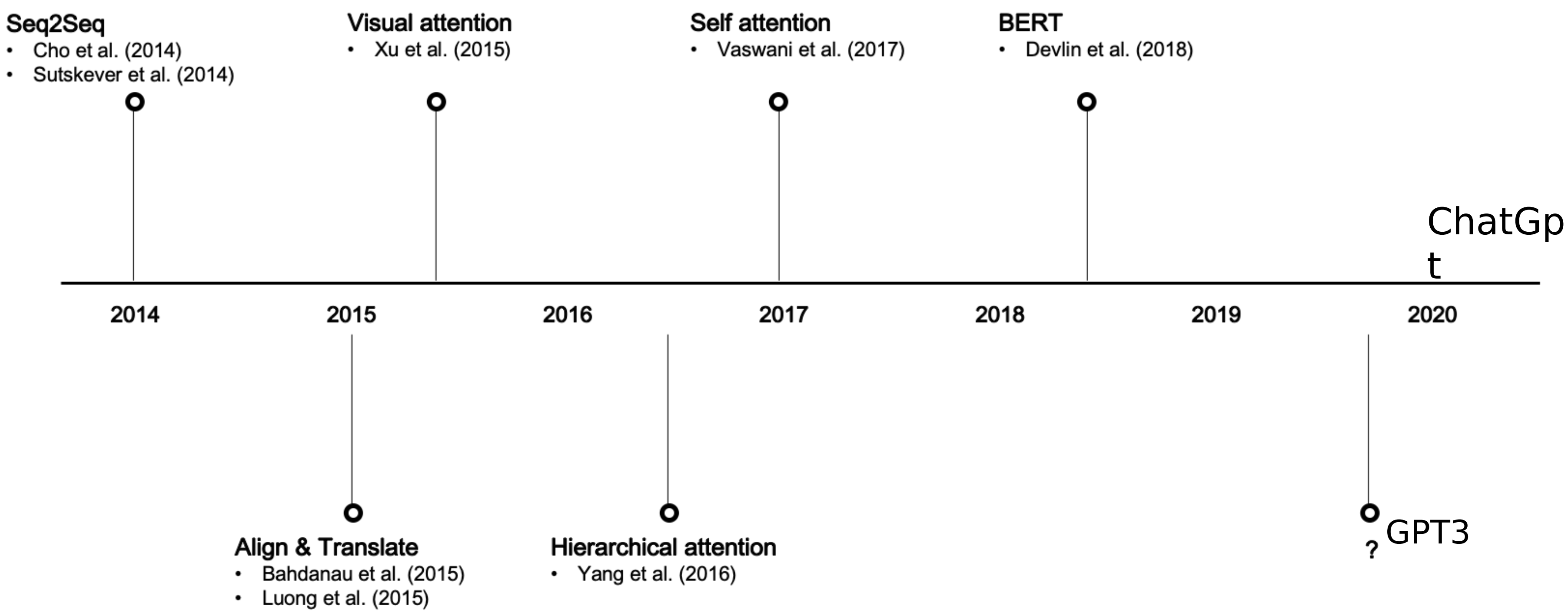
$$a(\boldsymbol{q}, \boldsymbol{k}) = \boldsymbol{q}^\intercal \boldsymbol{k}$$

Attention is so great (Bahdanau at al., 16054 Citations)

- Attention significantly improves NMT performance
  - It’s very useful to allow decoder to focus on certain parts of the source
- Attention provides a more “human-like” model of the MT process
  - You can look back at the source sentence while translating, rather than needing to remember it all
- Attention solves the bottleneck problem
  - Attention allows decoder to look directly at source; bypass bott
- Attention helps with the vanishing gradient problem
- 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



Key developments in attention



Attention is a *general* Deep Learning technique

- **More general definition of attention**:
  - Given a set of vector *values*, and a vector *query*, **attention** is a technique to compute a weighted sum of the values, dependent on the query.

Intuition:

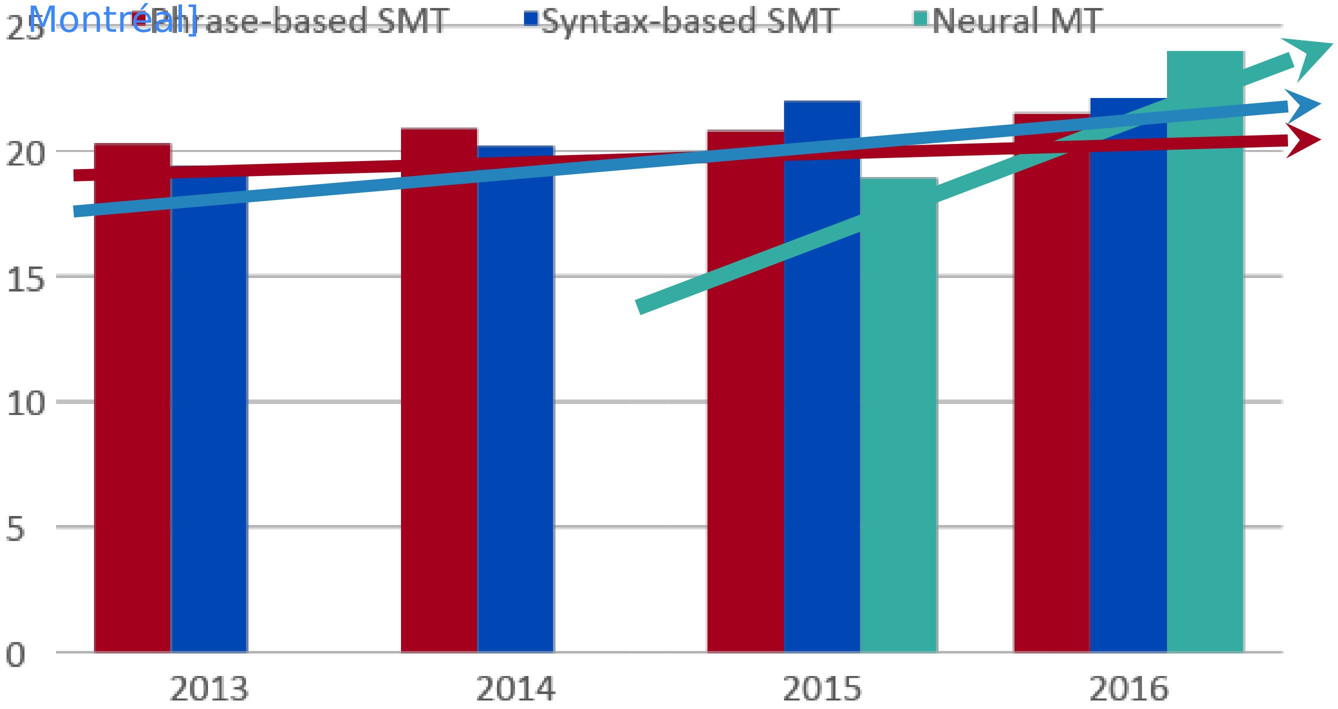
- The weighted sum is a *selective summary* of the information contained in the values, where the query determines which values to focus on.
- Attention is a way to obtain a *fixed-size representation of an arbitrary set of representations* (the values), dependent on some other representation (the query).

Upshot:

- Attention has become the powerful, flexible, general way pointer and memory manipulation in all deep learning models. A new idea from after 2010! From NMT!

Evolution of the MT system 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 first big 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 [Sutskever et al., 2014]

- **2016**: Google Translate switches from SMT to NMT and by 2018

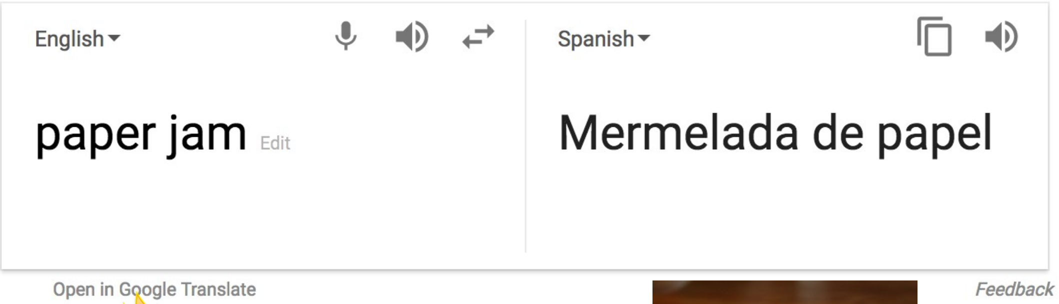
- This is amazing!
  - **SMT** systems, built by hundreds of engineers over many years, outperformed by NMT systems trained by a small groups of

MT solved?

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

MT solved?

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



?

Seq2seq is flexible and efficient!

- Seq2Seq is useful not only in the Machine Translation
- 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)
  - OCR (image of character □ text sequcence)
  - ASR (sequence of Acoustic □ text sequence)

Machine Translation Problem

- Automatic translation from English sentence to Vietnamese sentence
- Training và testing data: IWSLT 2015

Example:

Input: I like a blue book

Oputput: Tôi thích quyển sách màu xanh

Apply seq2seq + attention for NMT

Goals to be achieved:

1. Training seq2seq model for English-Vietnamese translation problem.
2. Evaluating the translation system by BLEU score.
3. Seing Attention score matrix to better understand the model

The tasks to be implemented:

4. Data preprocessing
5. Create training data
6. Write **encoder, decoder, attention** modul
7. Training model
8. Translate new source sentence
9. Show Attention, BLEU score.

Conclusion

- **Sequence-to-sequence** is the architecture of most current NLP problems such as NMT, text generation, ...

- **Attention great** is a way to *focus on particular parts* of the input
  - Improved Seq2Seq model **a lot**!
  - As **the foundation** of the Transformer model (now dominated!)

References

- Speech and Language Processing 2023 (https://web.stanford.edu/~jurafsky/slp3/)
- Machine Translation and Sequence-to-Sequence Models, Neubig, 2019, CMU
- Some slides for Universities of Stanford 2023, MIT, ...