Faculty for Informatics

Technical University of Munich

Natural Language Processing
IN2361

Prof. Dr. Georg Groh

Social Computing
Research Group

Deep NLP
Part C:
Machine Translation, Seq2Seq and Attention

- content is based on [2] (lecture 10)
- certain elements (e.g. figures, equations or tables) were taken over or taken over in a modified form from [2]
- citations of [2] are omitted for legibility; citations of original sources cited in [2] are omitted for legibility if not explicitly discussed
- errors on these slides are fully in the responsibility of Georg Groh
- BIG thanks to Richard Socher and his colleagues at Stanford for publishing materials [2] of a great Deep NLP lecture

Machine Translation

- task: translate sentence from source language to target language

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

⇩

Man is born free, but everywhere he is in chains

- 1950- … : early machine translation: rule-based systems based on bilingual dictionaries

[3]

Machine Translation: Statistical Models

- 1990s-2010s: core idea: learn probabilistic model from data

- find best sentence y in target language given a source language sentence x:

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

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

Translation Model

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

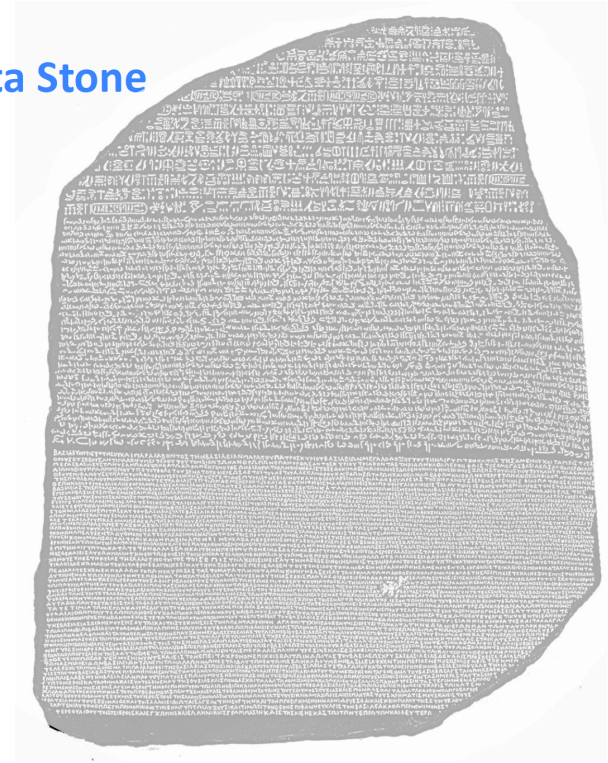
Language Model

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

Machine Translation: Statistical Models

- translation model $P(x|y)$: large amounts of parallel data required

The Rosetta Stone



Ancient Egyptian

Demotic

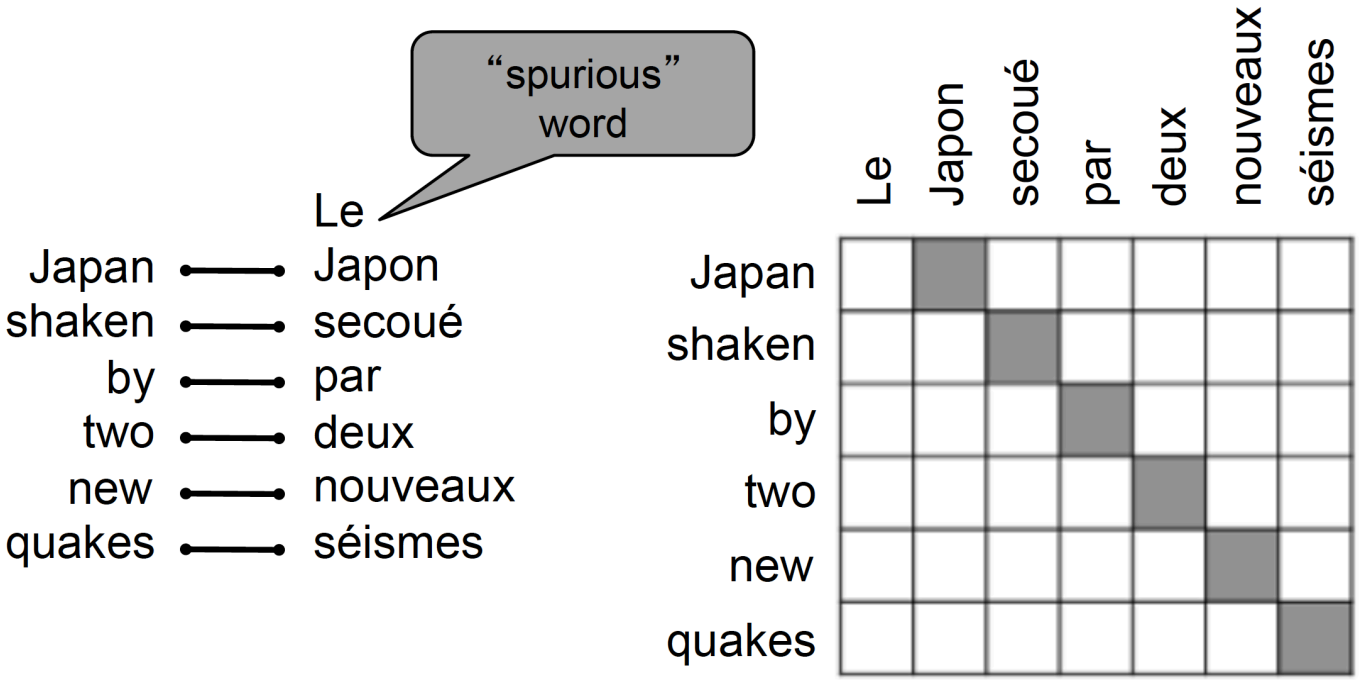
Ancient Greek

- more accurately, we need to learn $P(x, a|y)$, where a is an alignment between source sentence x and target sentence y

Alignment

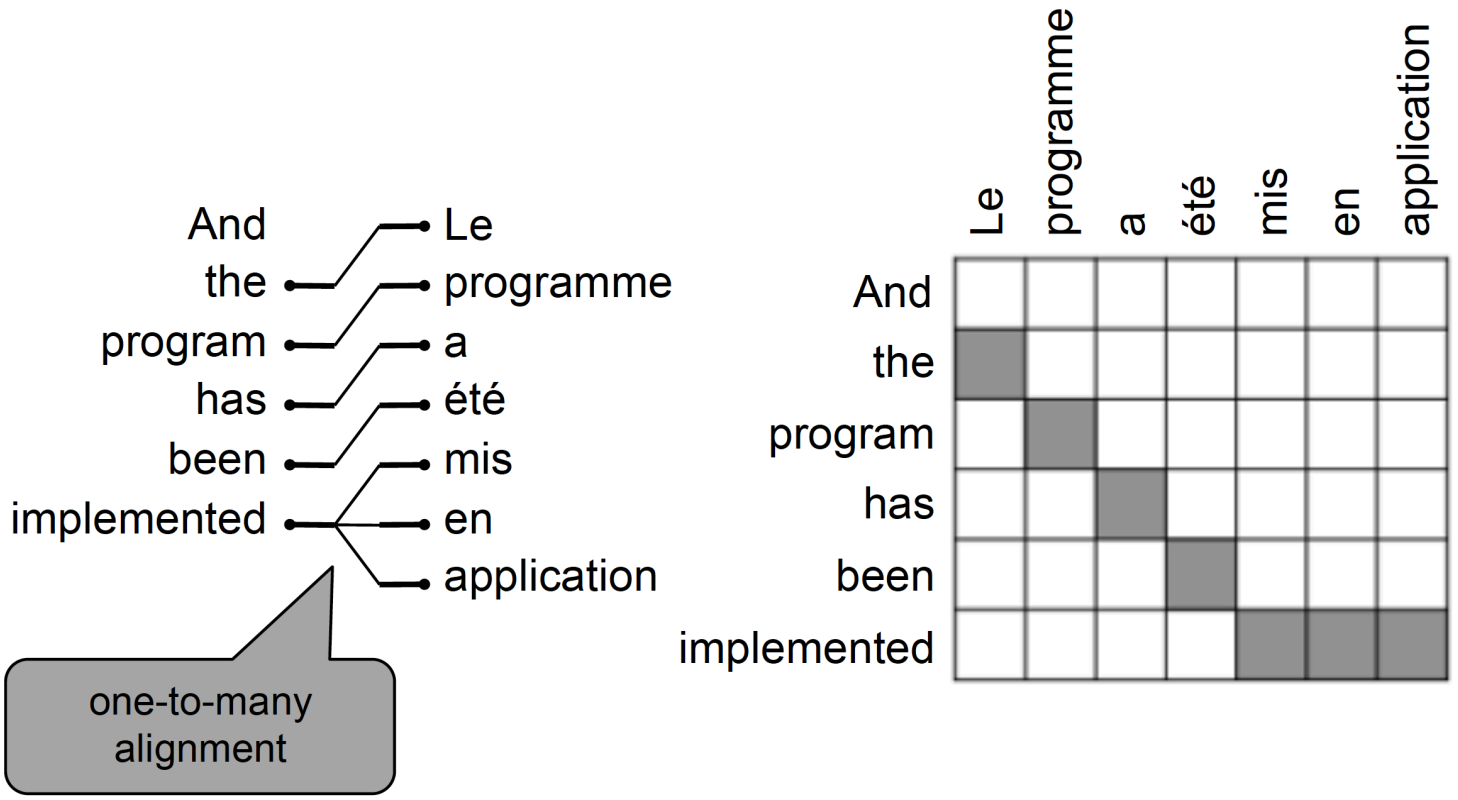
- alignment: correspondence between particular words in translated sentence pair

- some words may have no counterpart



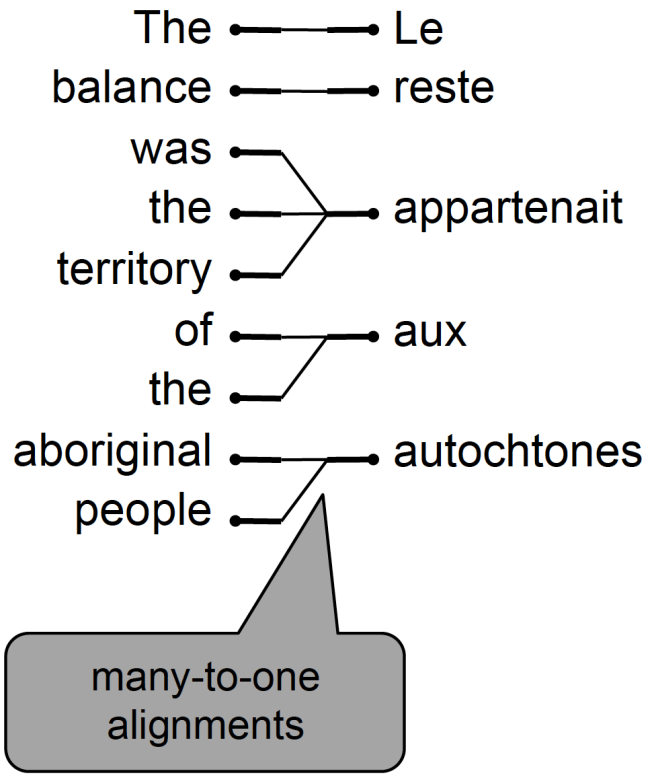
Alignment

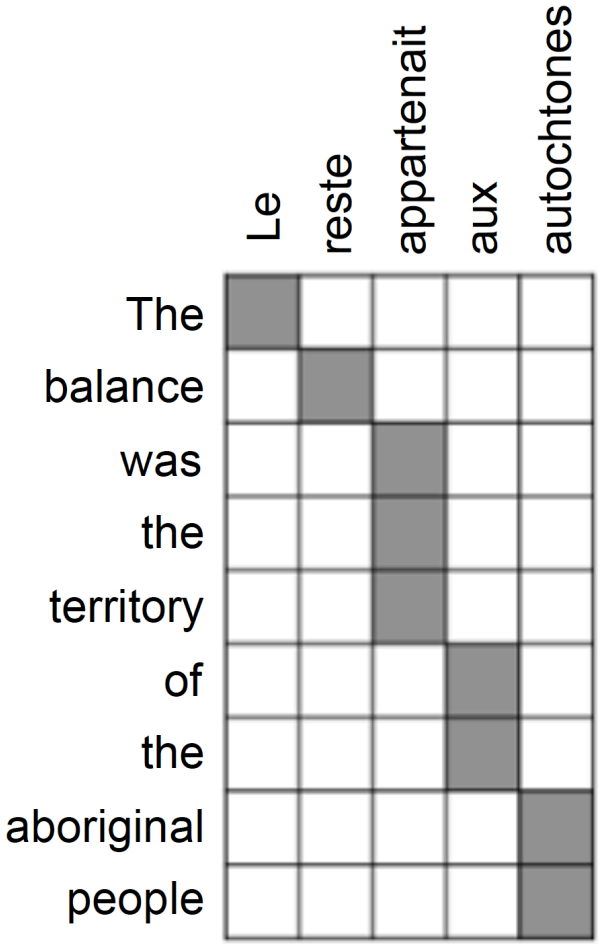
can be one-to-many (“fertile” words)



Alignment

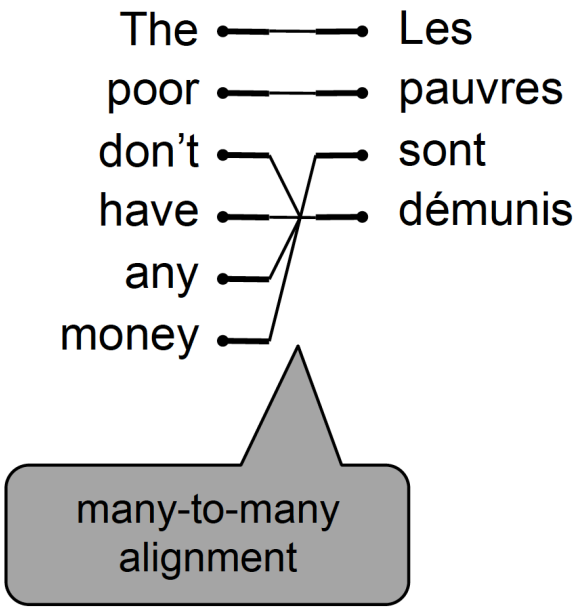
or many-to-one

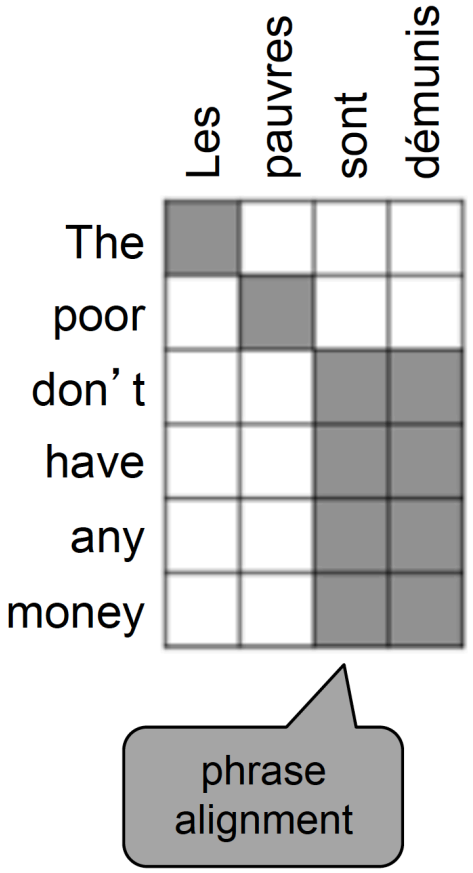




Alignment

or many-to-many (phrase level)





Statistical Models

- influences on $P(x, a|y)$:
  - probability of alignment between of particular words
  - fertility probability of words
  - …

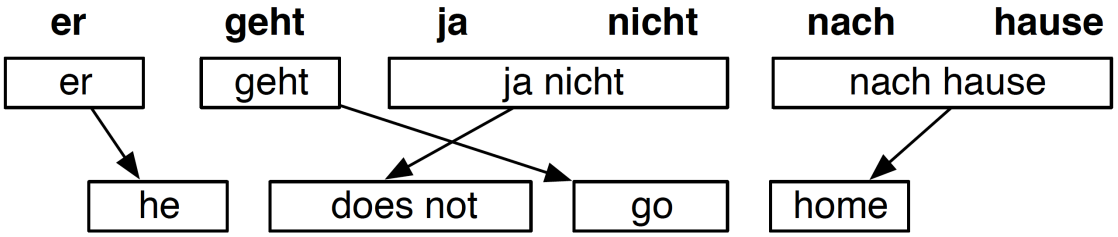
- argmax:

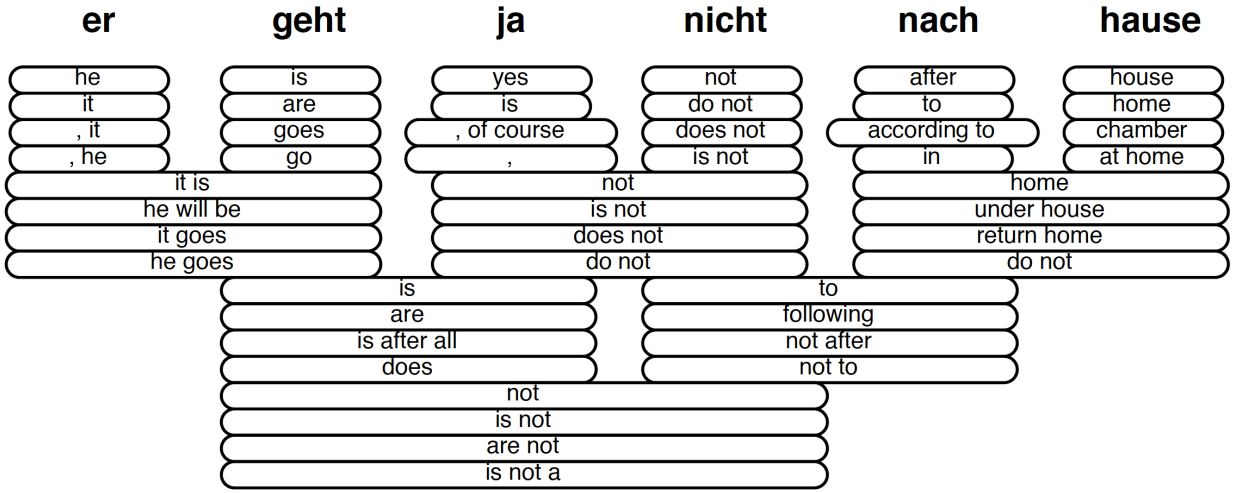
$$\underbrace{\text{argmax}_y}_{\text{Question:}} \underbrace{P(x|y)}_{\text{Translation Model}} \underbrace{P(y)}_{\text{Language Model}}$$

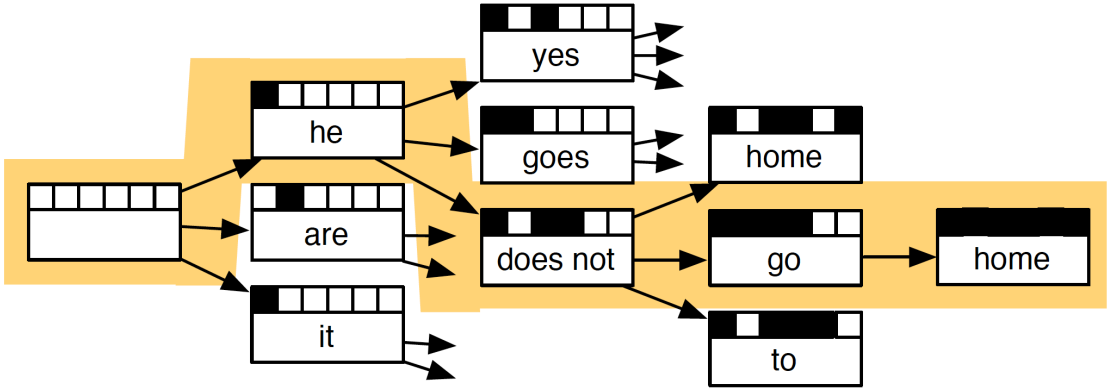
Question: How to compute this argmax?

  - enumerate all possible y and calculate probability → too expensive!
  - instead: heuristic search: build up translation gradually, discard low probability hypotheses

Heuristic Search



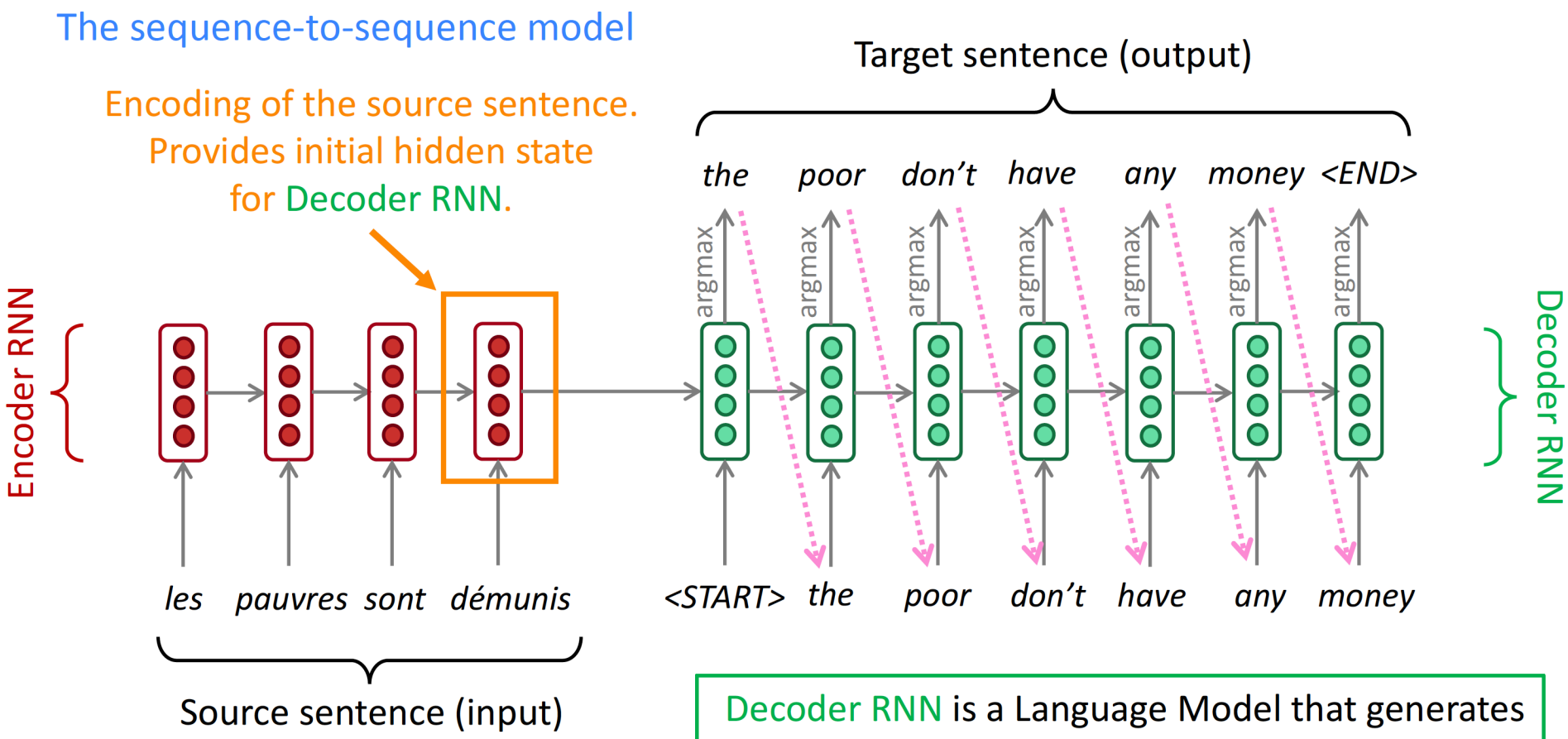




Statistical Models

- best systems extremely complex
- lots of important details
- many separately-designed subcomponents
- lots of feature engineering
  - need to design features to capture particular language phenomena
- require compiling and maintaining extra resources: e.g. tables of equivalent phrases
- → lots of human effort to maintain for each language pair

Solution: Seq2Seq NN-Based Machine Translation ☺



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

Solution: Seq2Seq NN-Based Machine Translation (NMT) ☺

The sequence-to-sequence model

Target sentence (output)

Encoding of the source sentence.

Encoder RNN

Decoder RNN

- example of Conditional Language Model:
  - **language model**: decoder predicts next word of target sentence conditioned on previous word of target sentence
  - **conditional**: predictions also conditioned on source sentence x

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

Probability of next target word, given target words so far and source sentence *x*

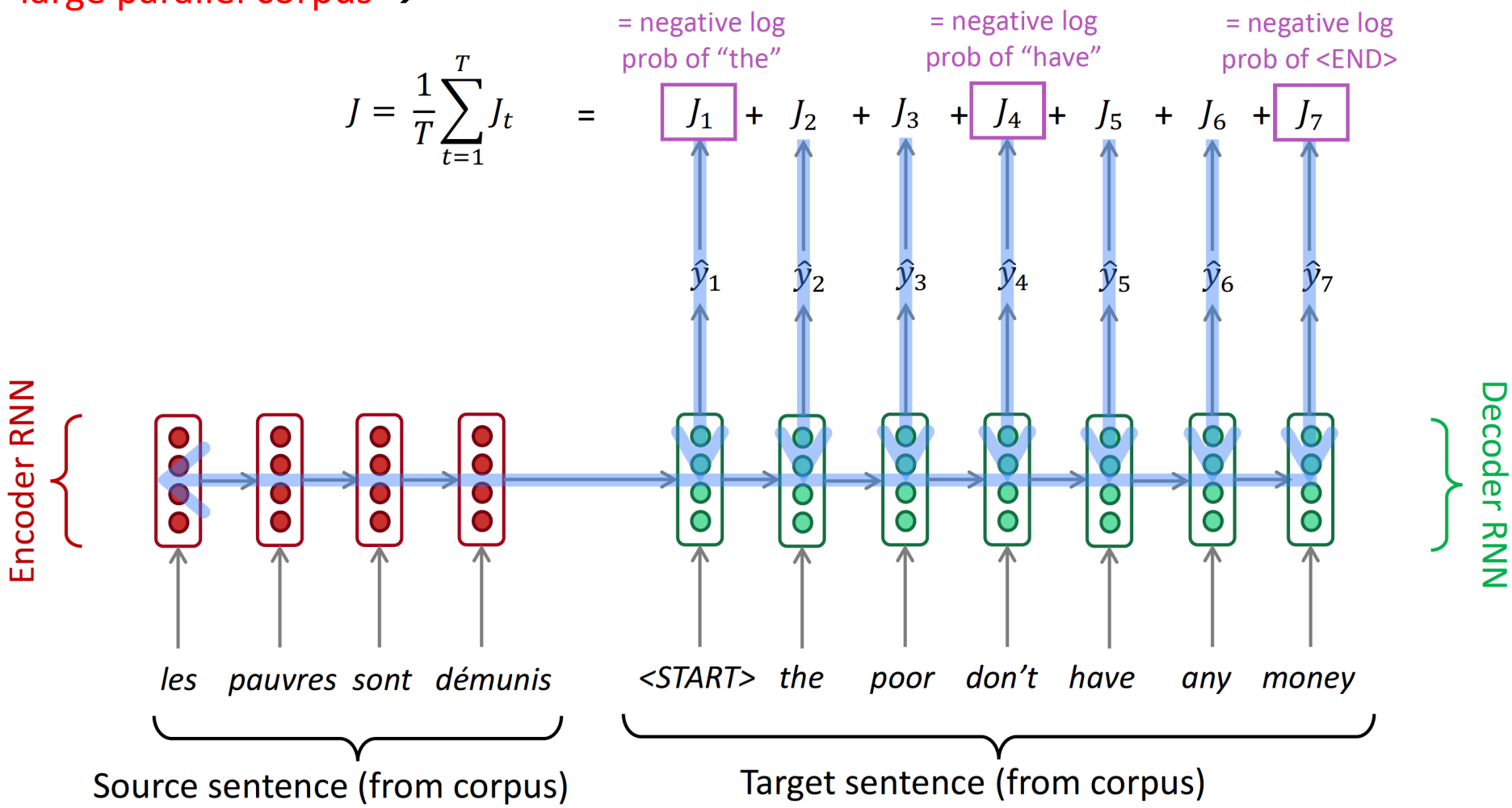
enerates
ling.

an encoding of the source sentence.

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

NMT Training: End to End

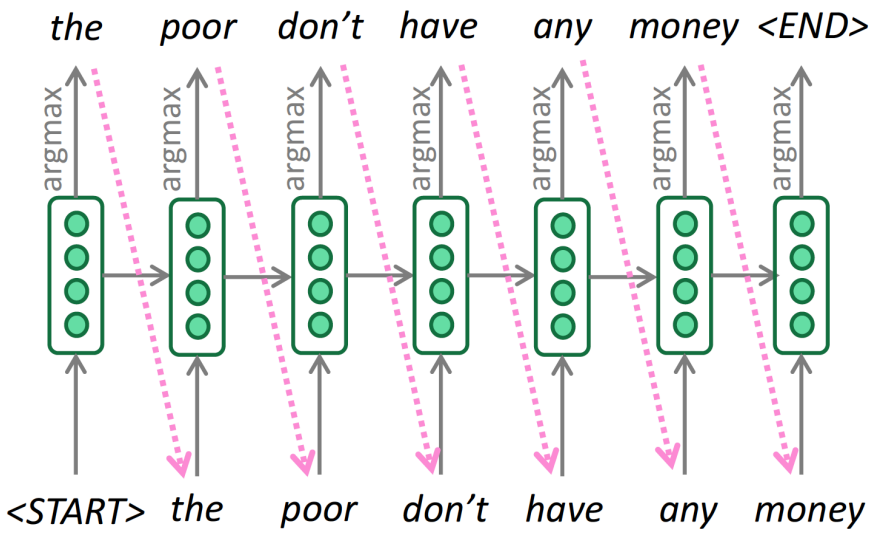
large parallel corpus →



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

Problems of Greedy Decoding

- decoder uses greedy decoding: take most probably word at each step:

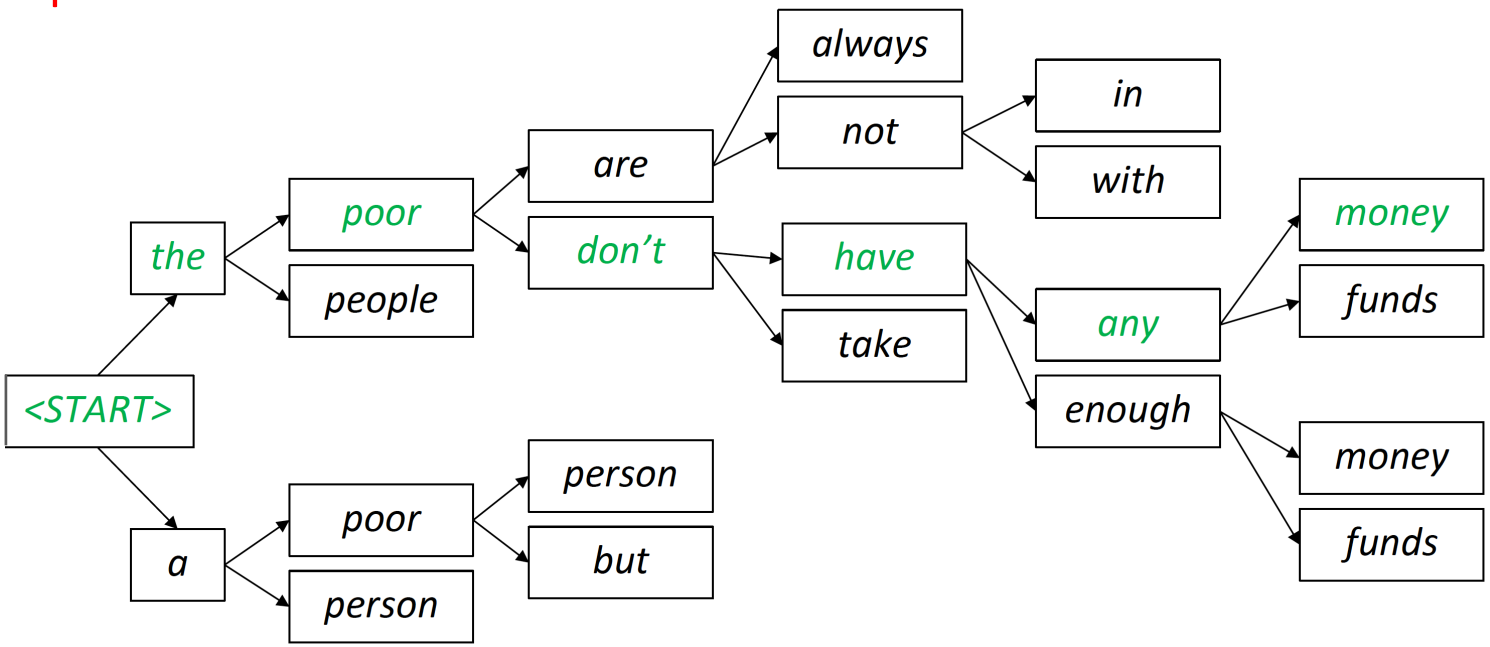


- problem: no way to undo decisions:
  - *les pauvres sont démunis (the poor don't have any money)*
  - → *the* ____
  - → *the poor* ____
  - → *the poor are* ____

Beam Search Decoding

- → use Beam Search: on each step of decoder, keep track of the k most probable partial translations
  - k: beam size (in practice around 5 to 10)
  - not guaranteed to find optimal solution, but much more efficient!

- example: k=2:



Advantages and Disadvantages of NMT

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

Advantages and Disadvantages of NMT

Disadvantages of NMT?

Compared to SMT:

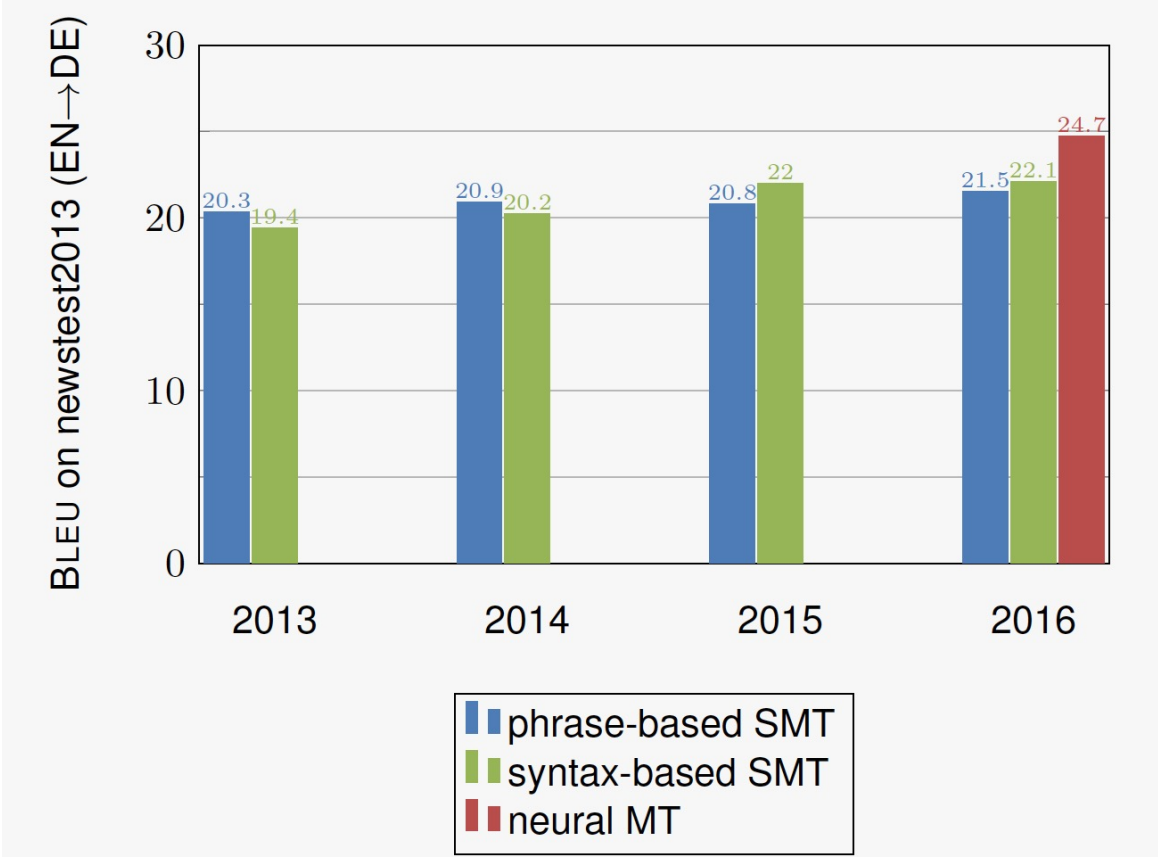
- NMT is less interpretable
  - Hard to debug
- NMT is difficult to control
  - For example, can't easily specify rules or guidelines for translation
  - Safety concerns!

Evaluation of Machine Translation

- BLEU (Bilingual Evaluation Understudy): compare machine translation to one or more human translations & compute similarity score based on:
  - n-gram precision (usually n=3 or 4)
  - & extra penalties for too short machine translations

- disadvantages of BLEU: many valid translations exist → poor BLEU score of an otherwise good translation that just differs in n-gram overlap from the available human translations in corpus

Progress in Machine Translation

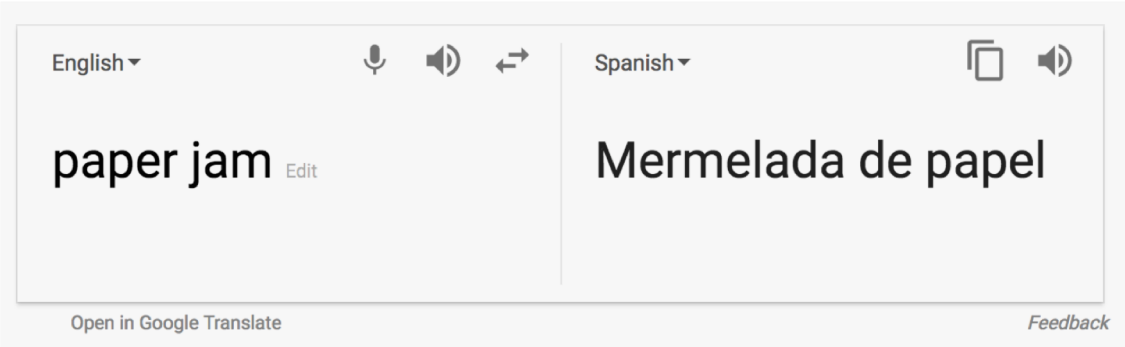


[4]

- 2014: first seq 2 seq paper
- 2016: Google translate switches from SMT to NMT

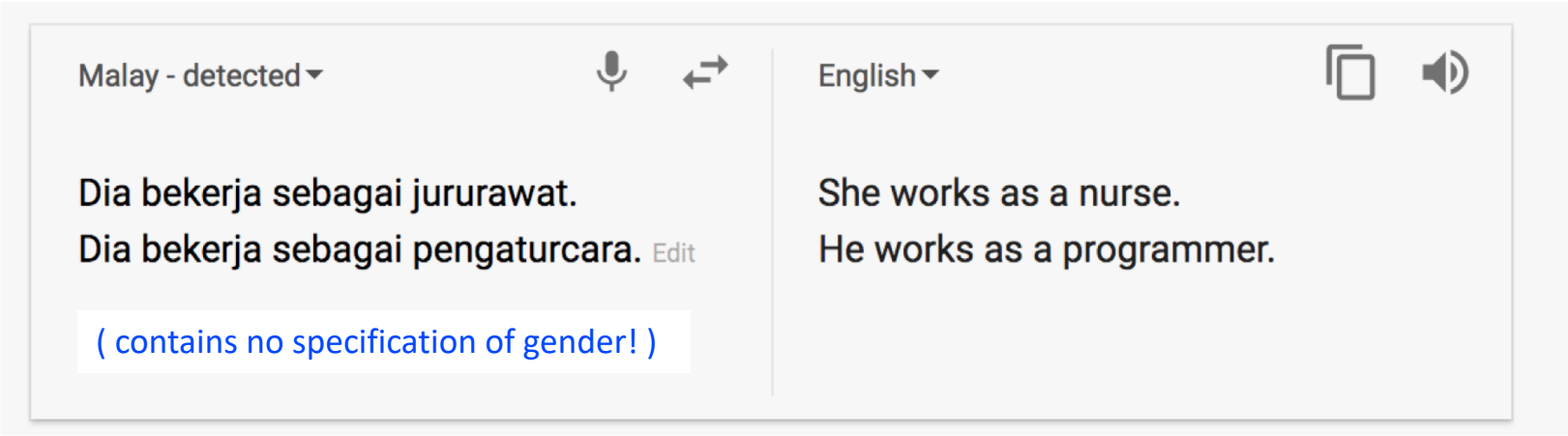
Prevailing Problems

- out-of-vocabulary words
- domain mismatch between train and test data
- maintaining context over longer text
- low-resource language pairs
- using common sense still hard:



Prevailing Problems

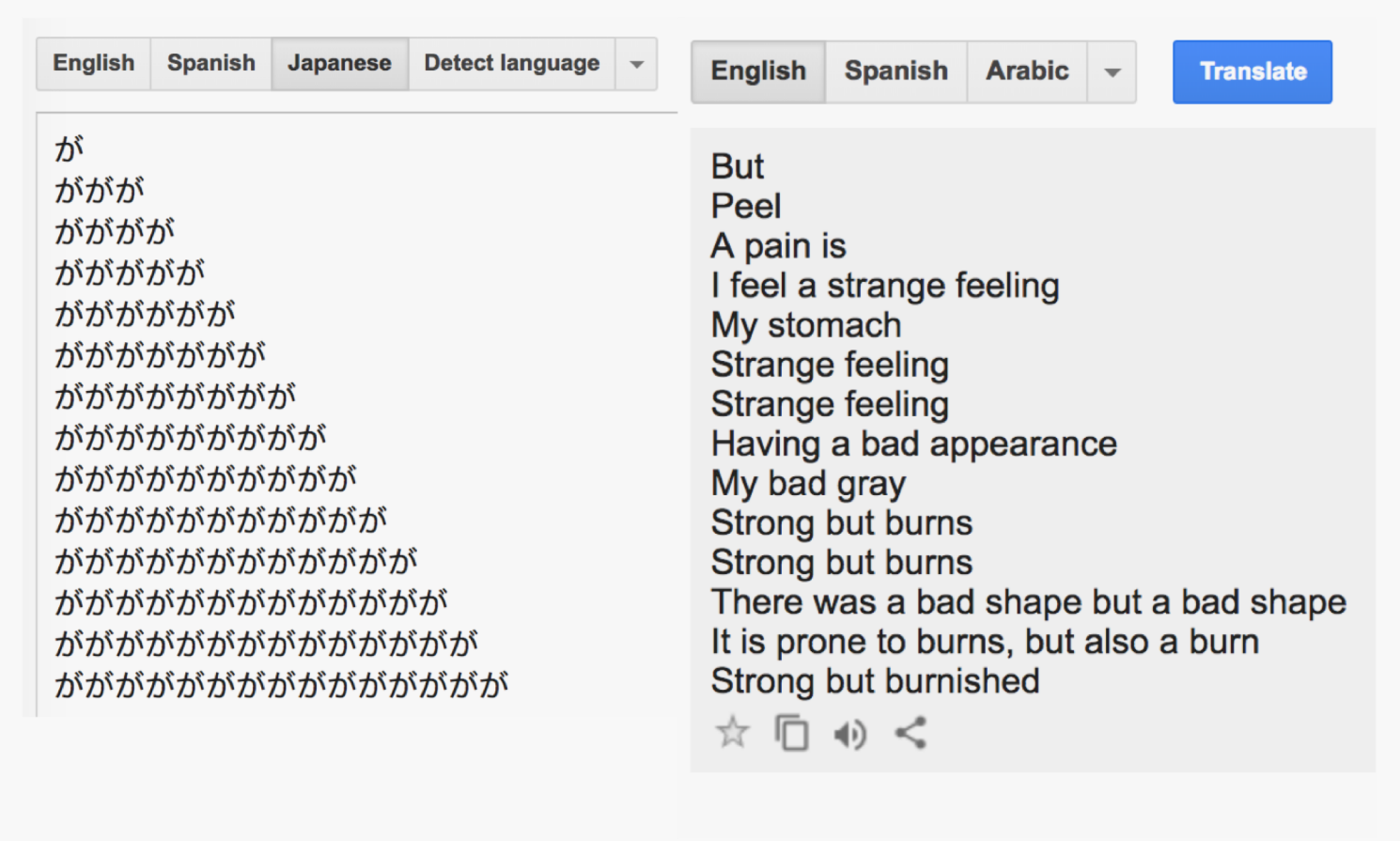
- NMT picks up cultural bias in training data:



[5]

Prevailing Problems

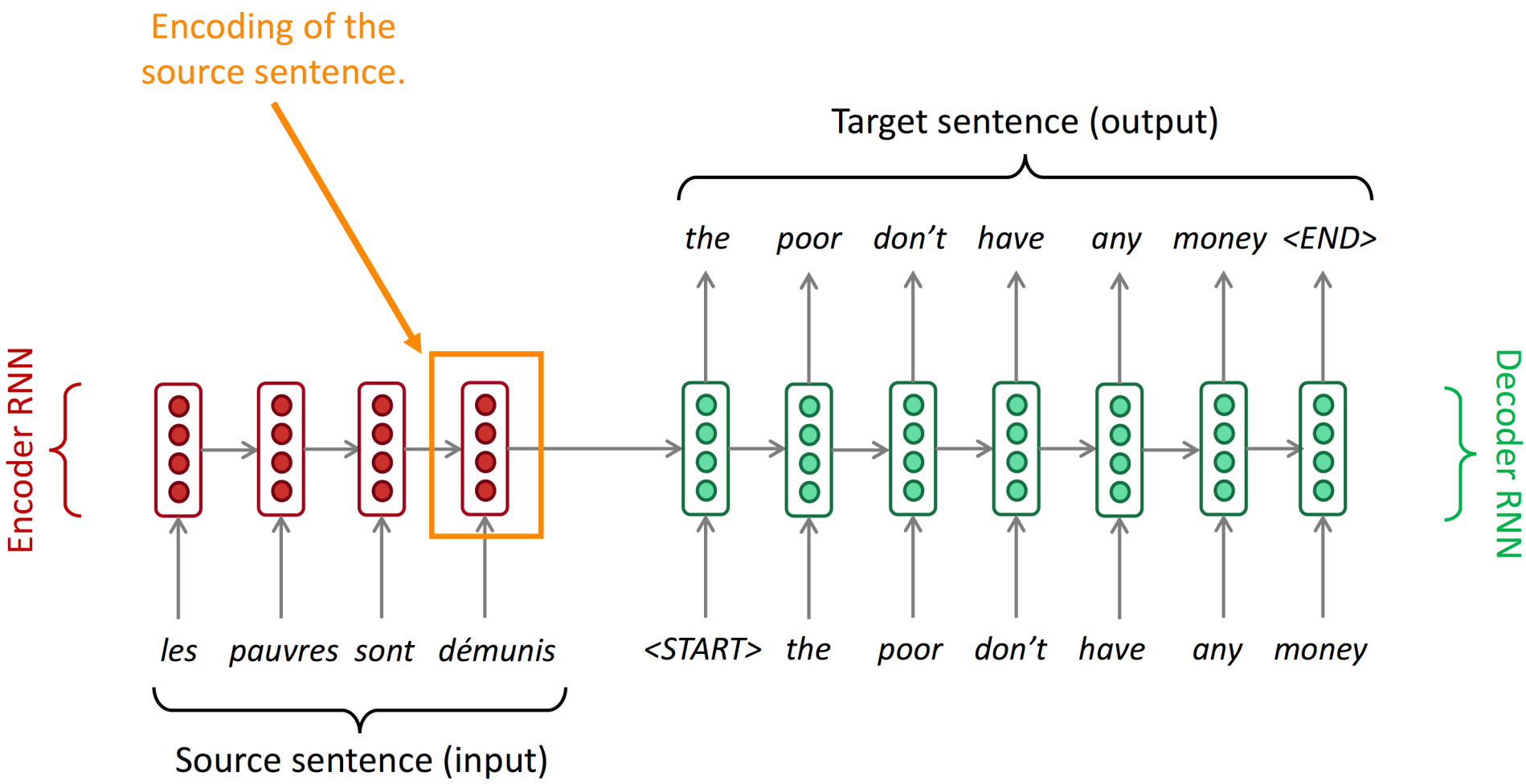
- Uninterpretable systems do strange things [1] :



[6]

[1] : philosophical question: is this really a problem? would humans perform systematically better?

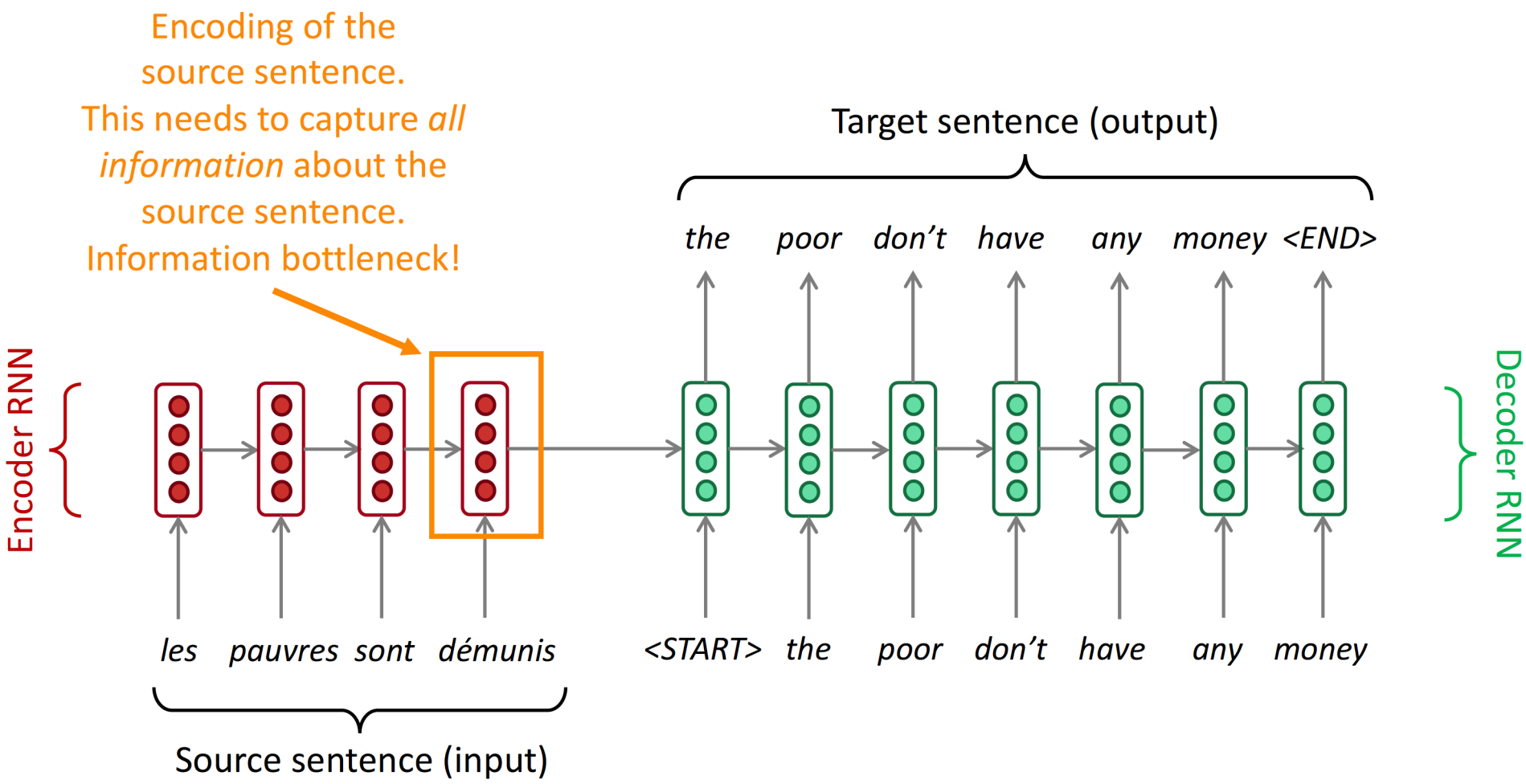
Seq2Seq NMT



Problems with this architecture?

Attention

- bottleneck problem

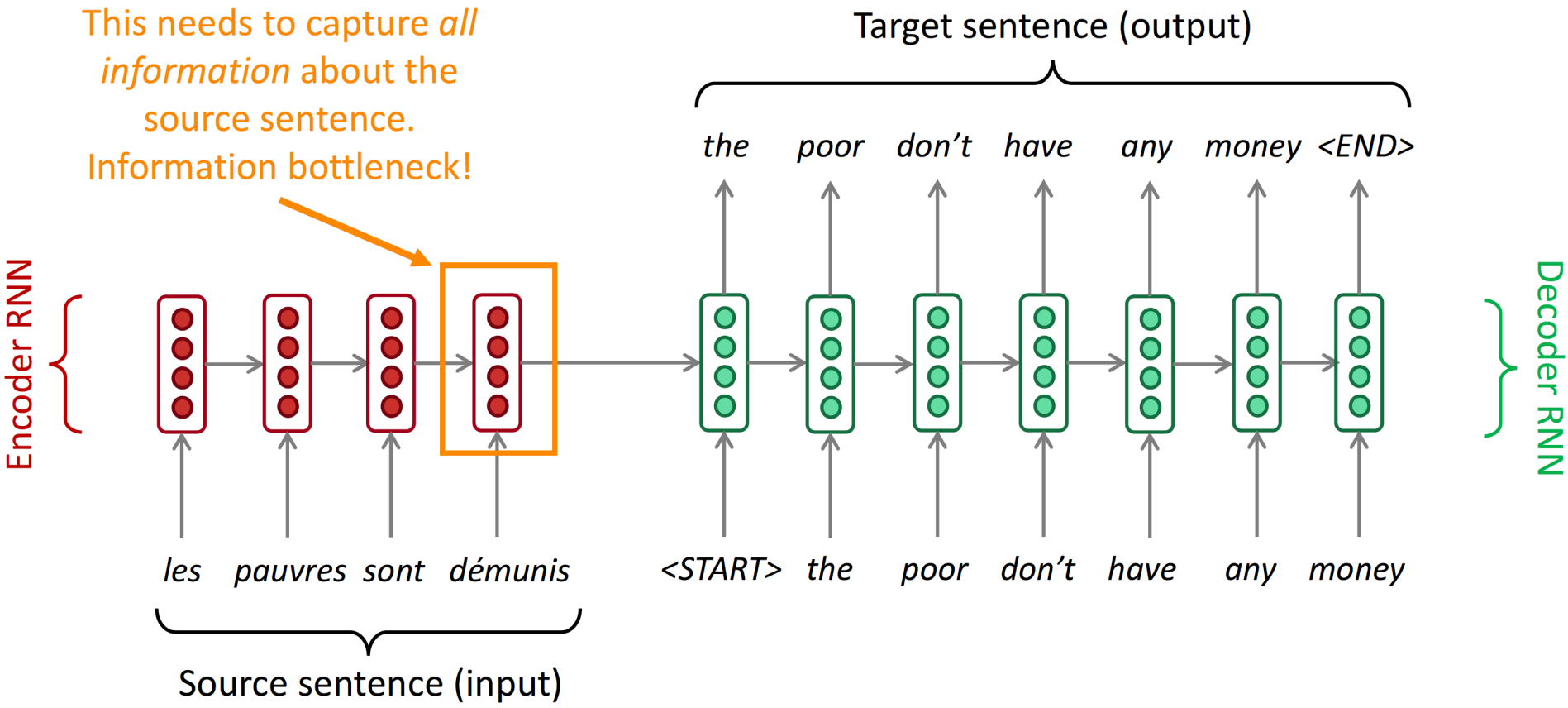


Attention

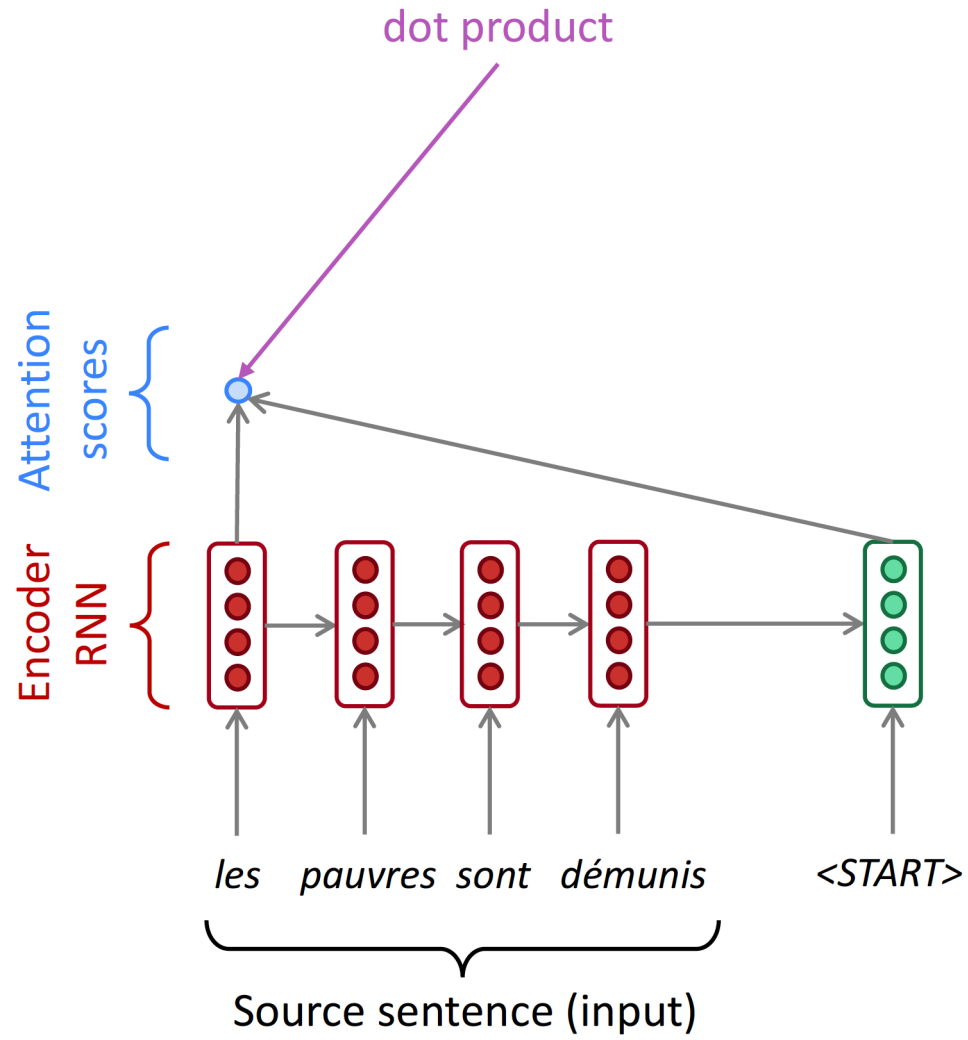
- bottleneck problem → most important new method in NMT: Attention

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

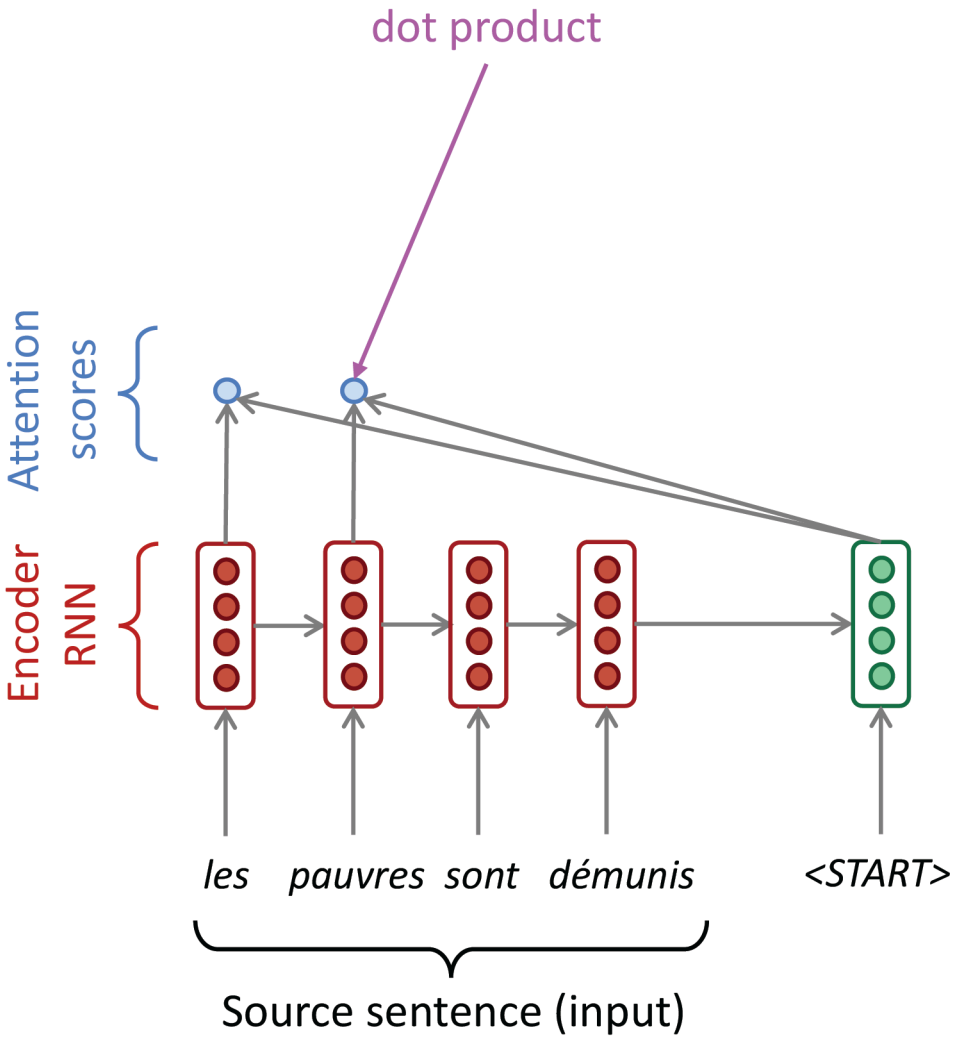
Encoding of the source sentence. This needs to capture *all information* about the source sentence. Information bottleneck!



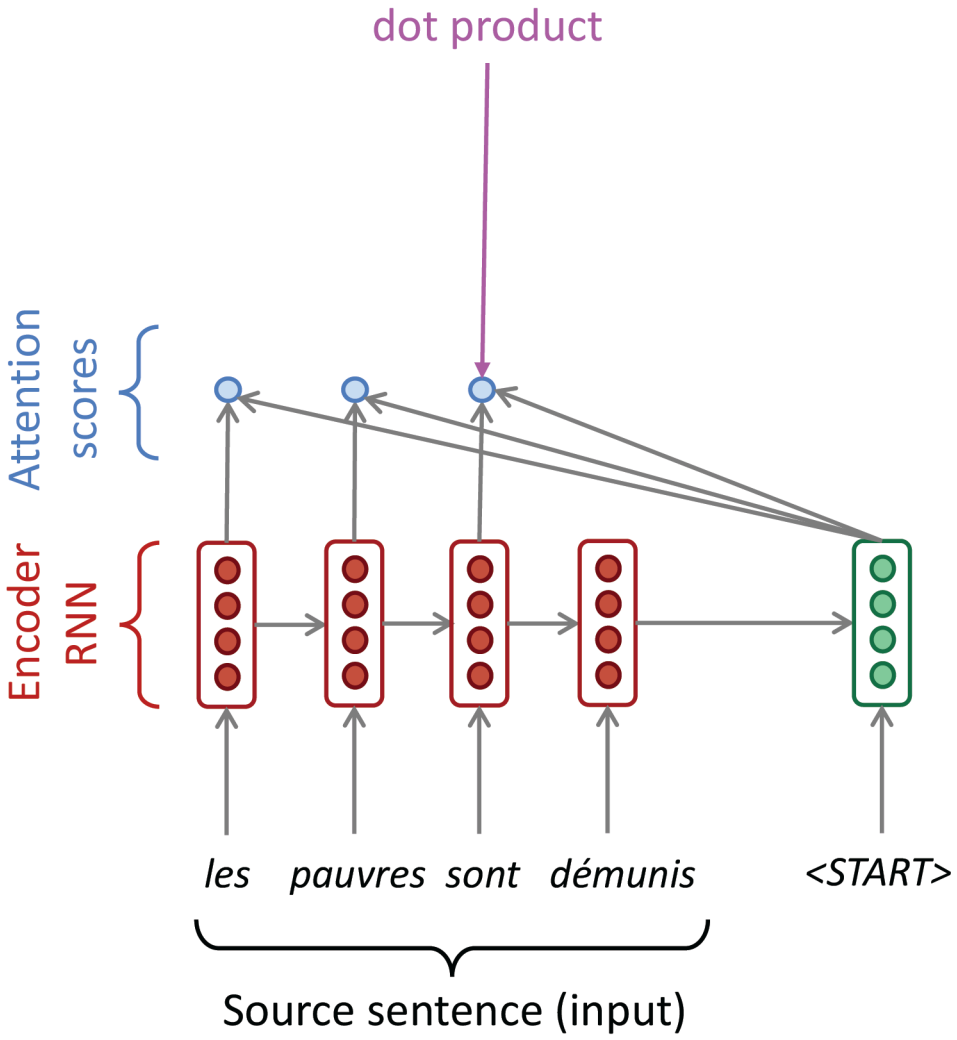
Seq2Seq with Attention



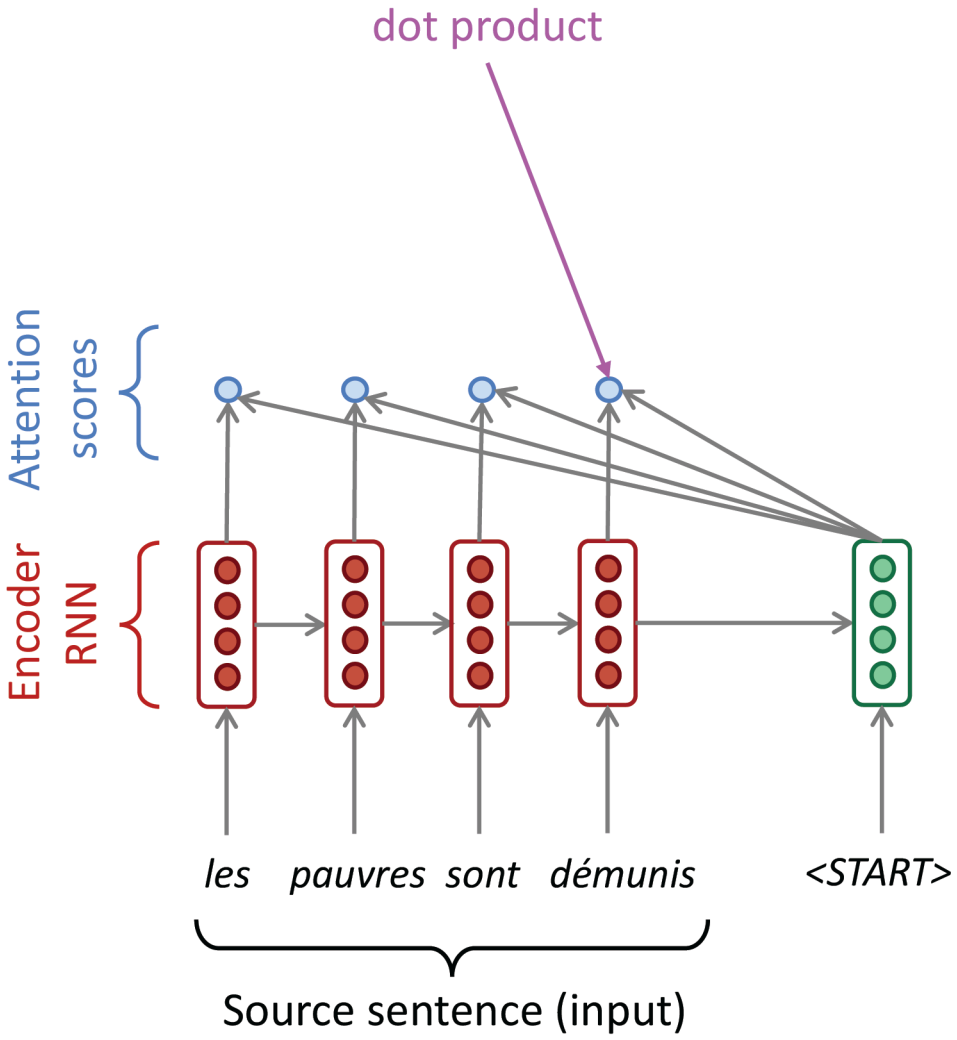
Seq2Seq with Attention



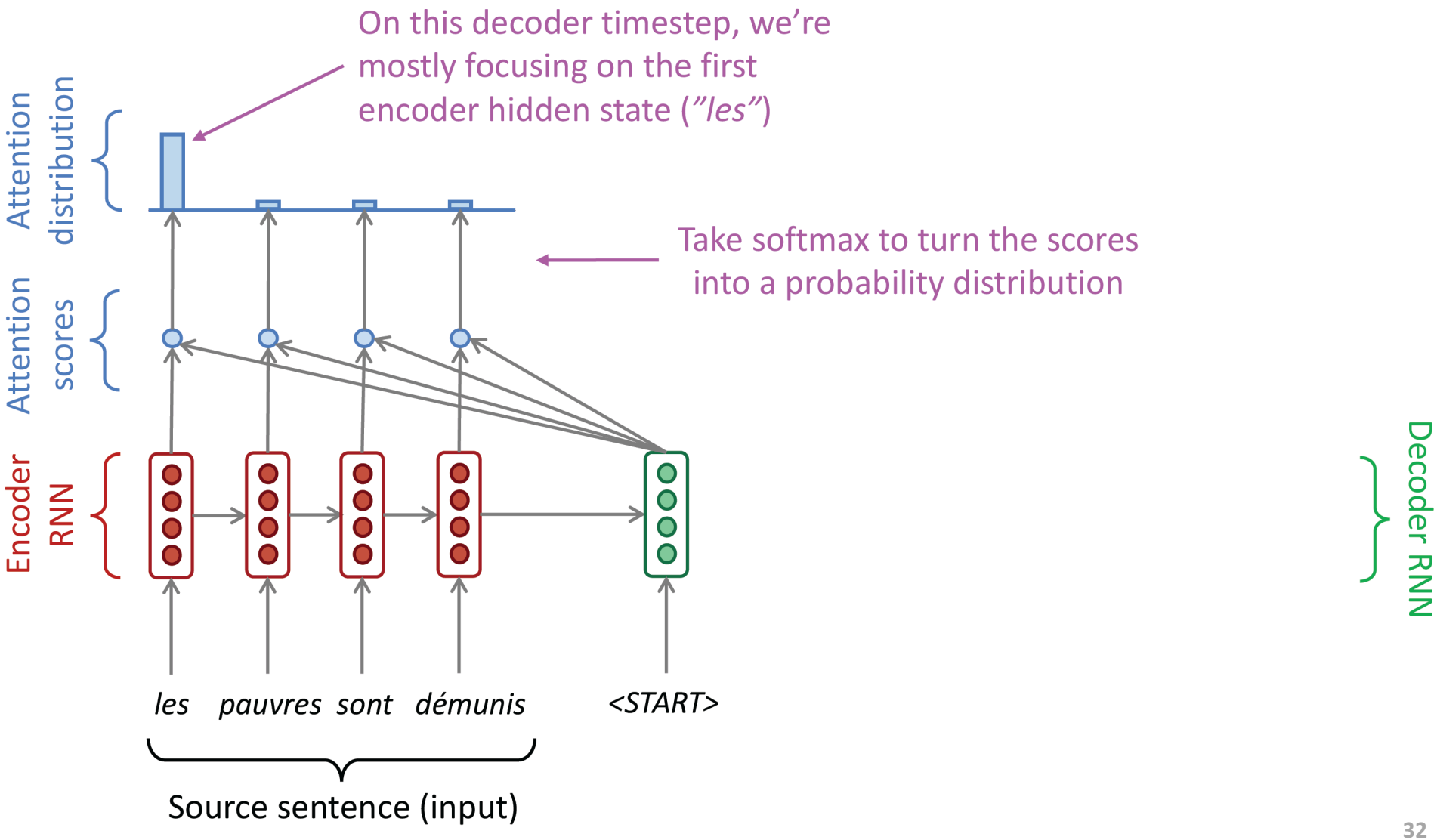
Seq2Seq with Attention



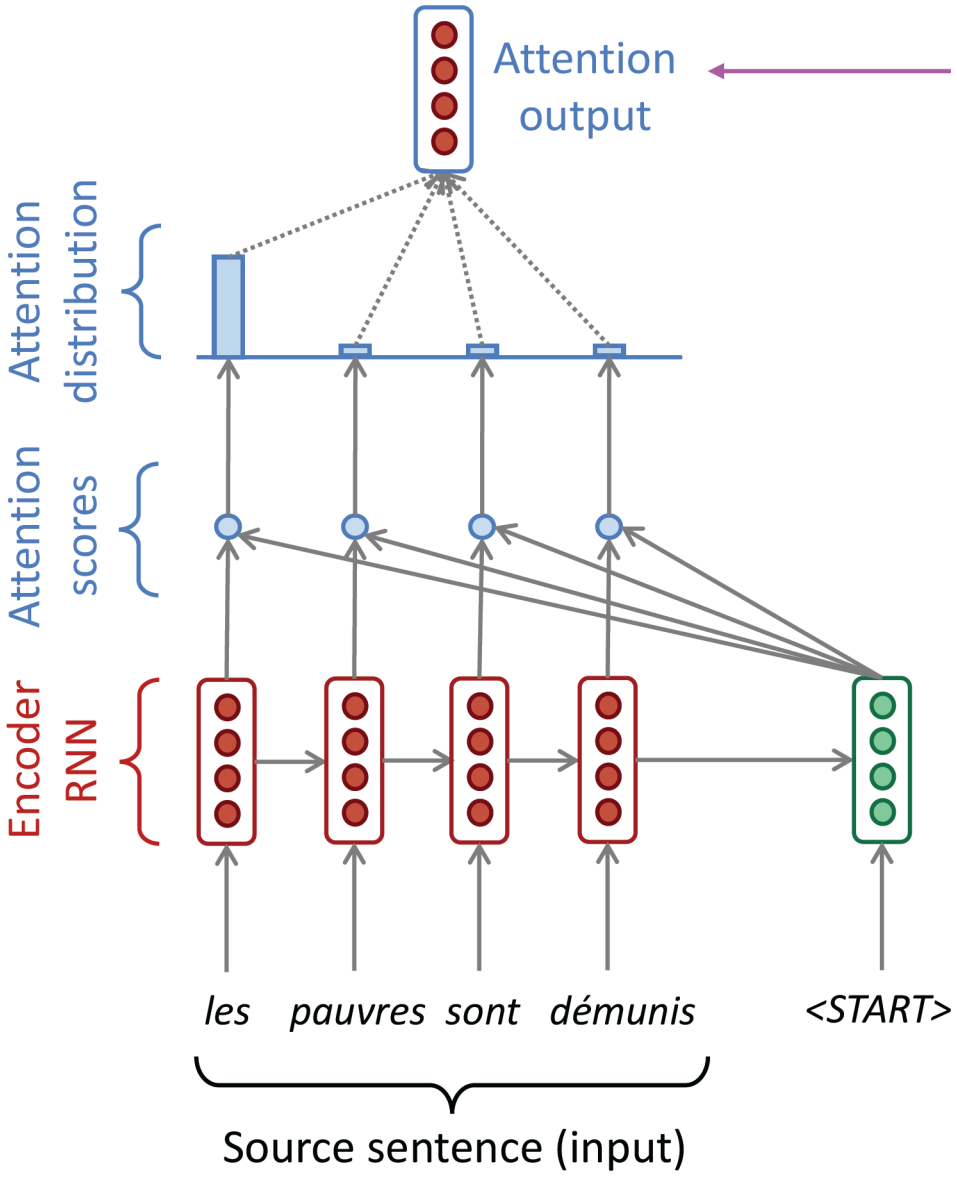
Seq2Seq with Attention



Seq2Seq with Attention



Seq2Seq with Attention

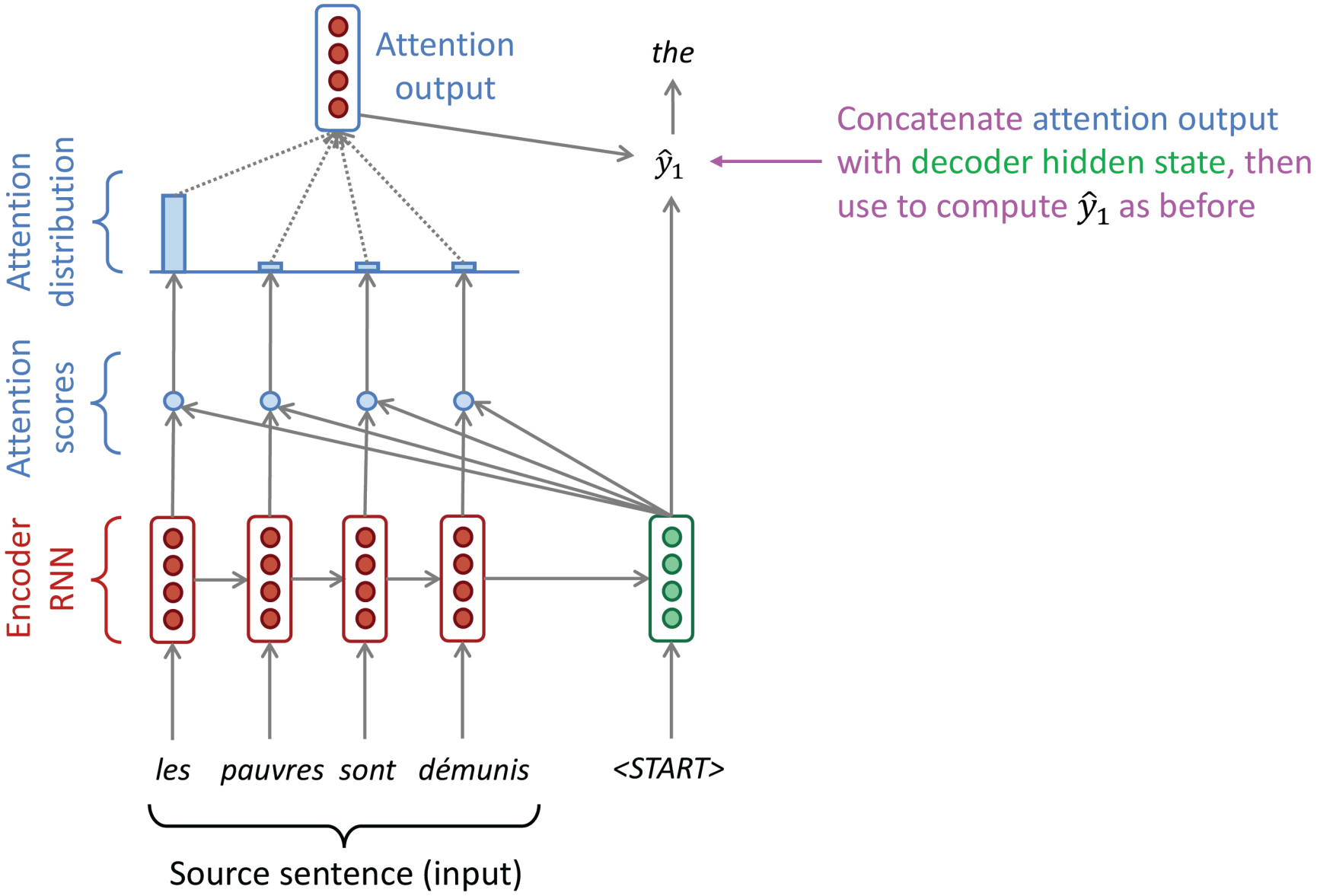


Use the attention distribution to take a **weighted sum** of the encoder hidden states.

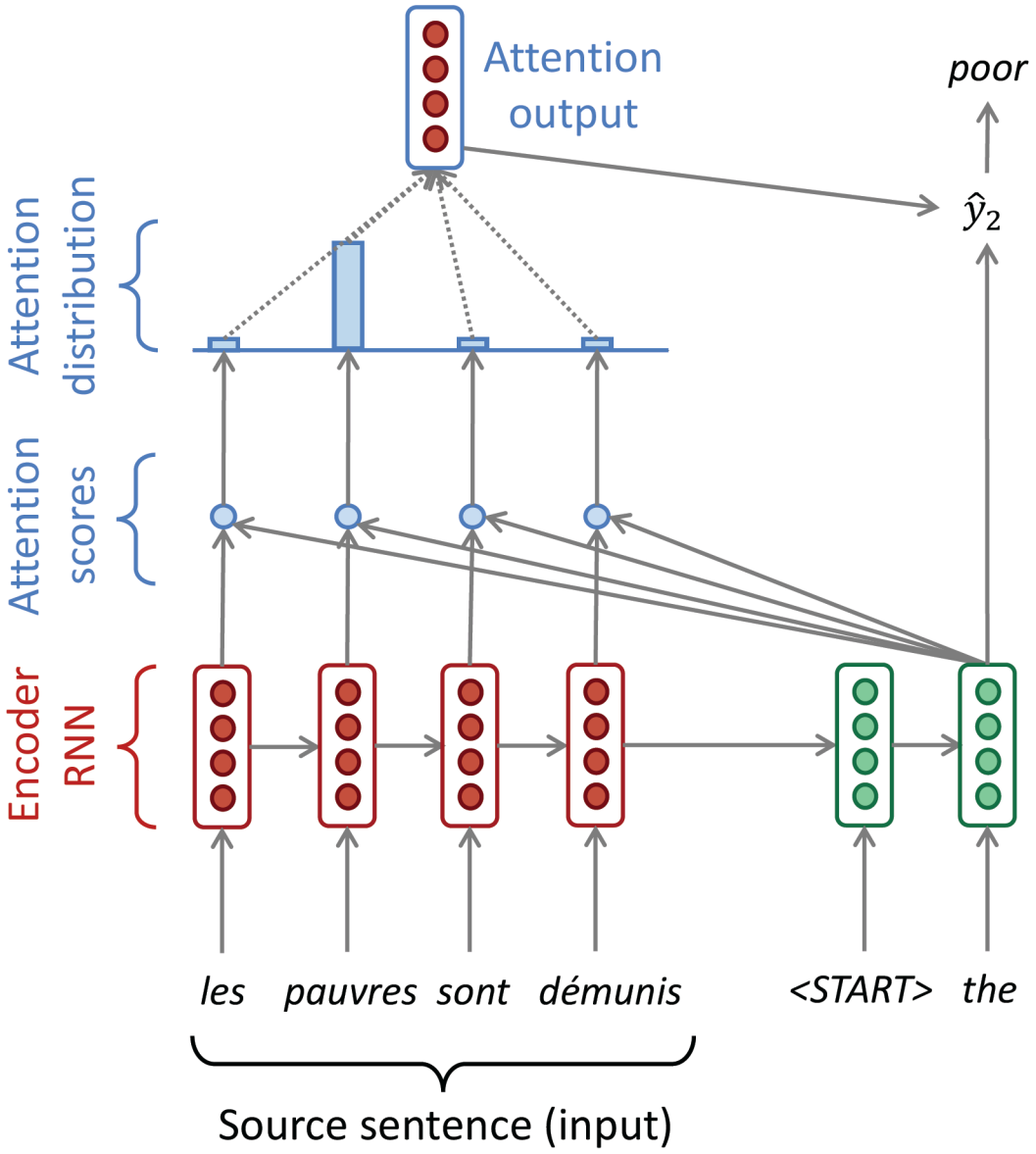
The attention output mostly contains information the hidden states that received high attention.

Decoder RNN

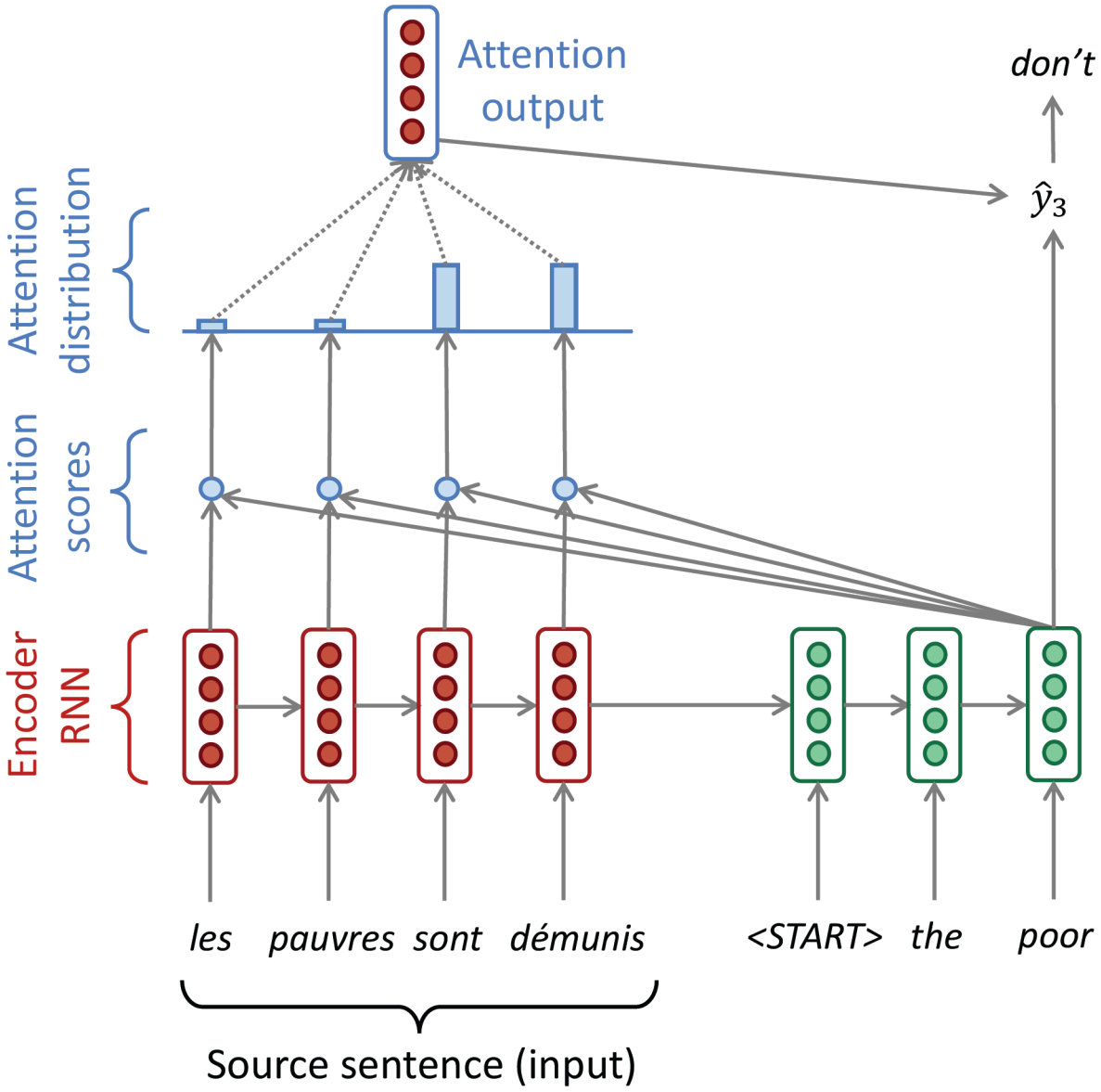
Seq2Seq with Attention



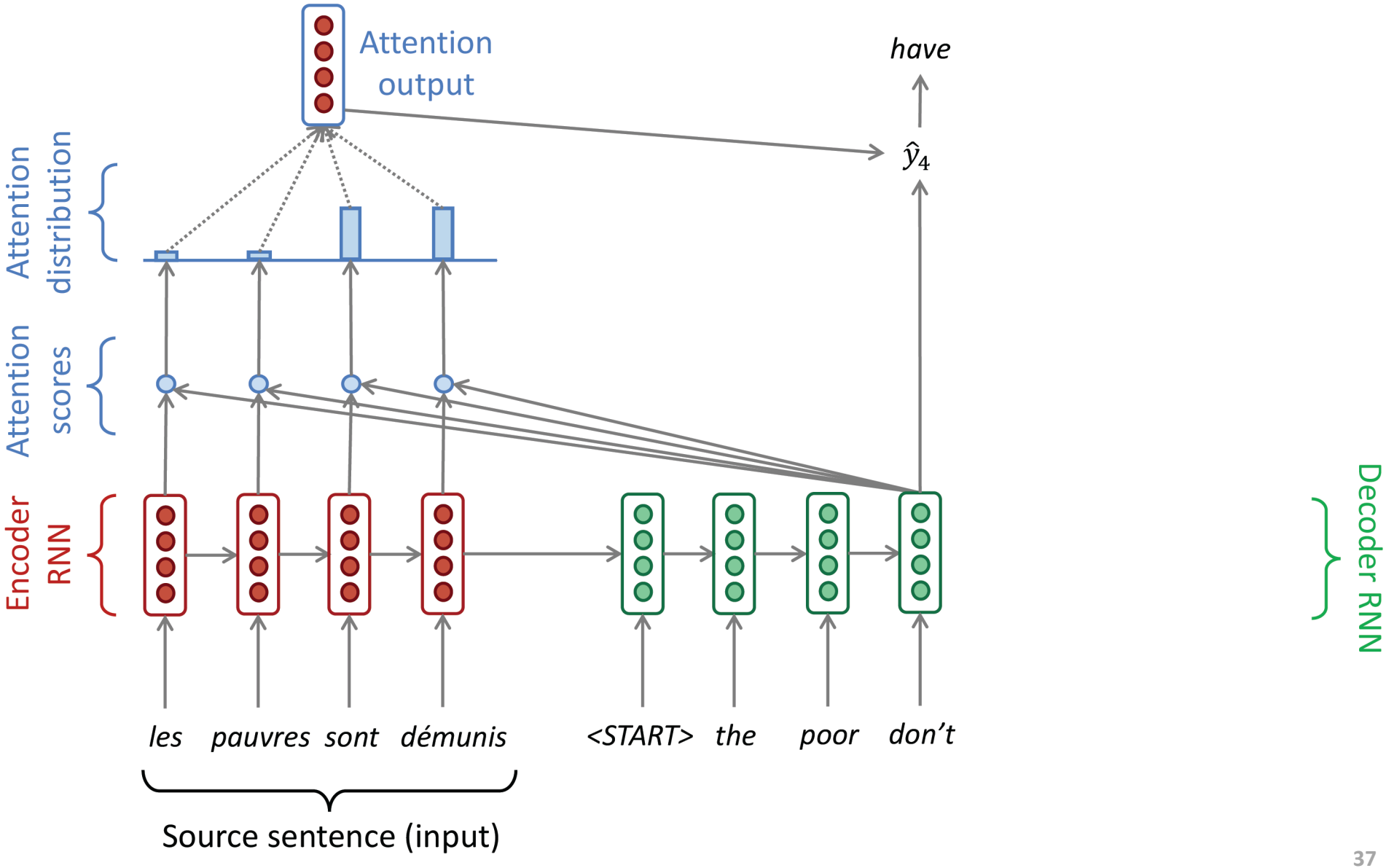
Seq2Seq with Attention



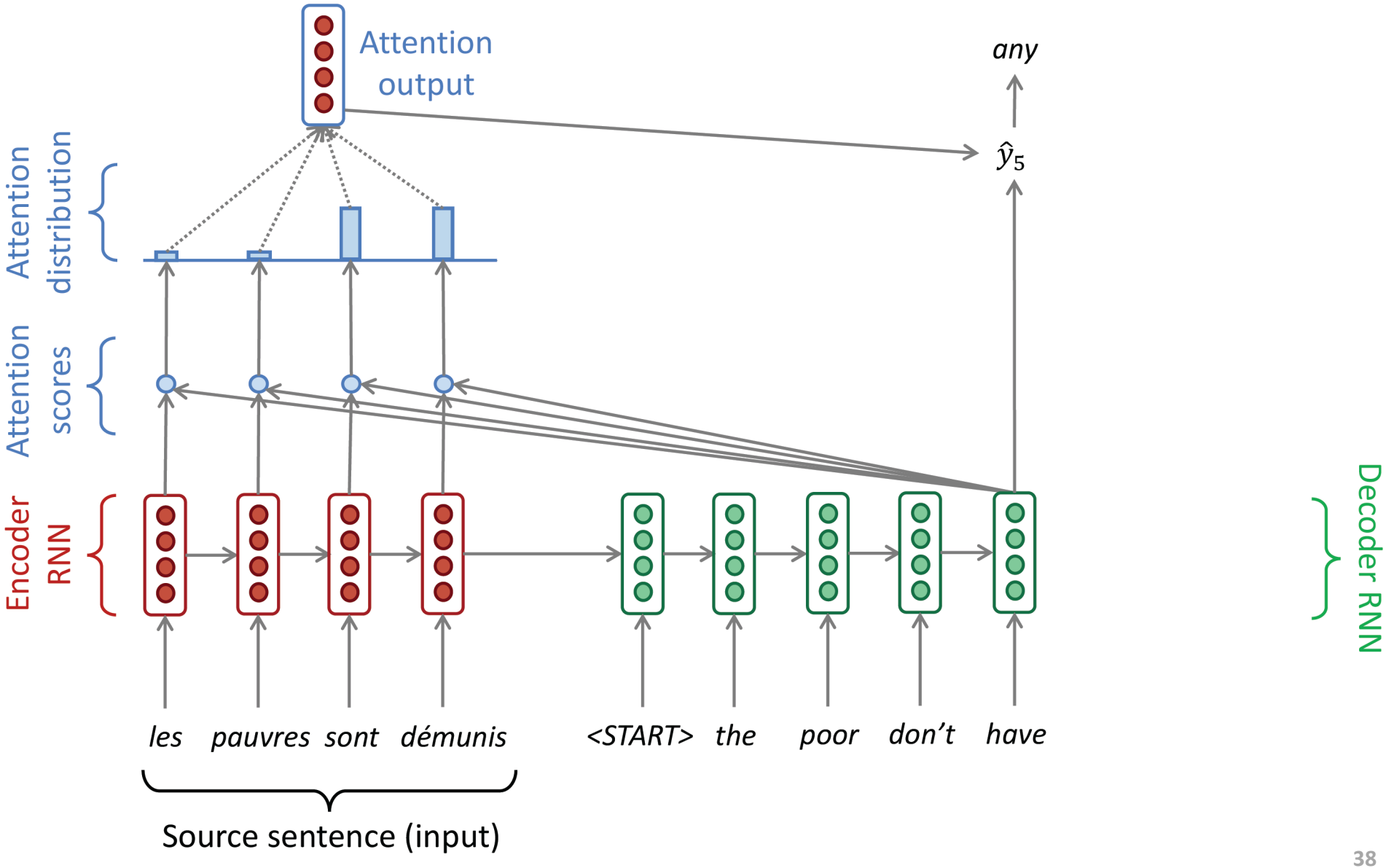
Seq2Seq with Attention



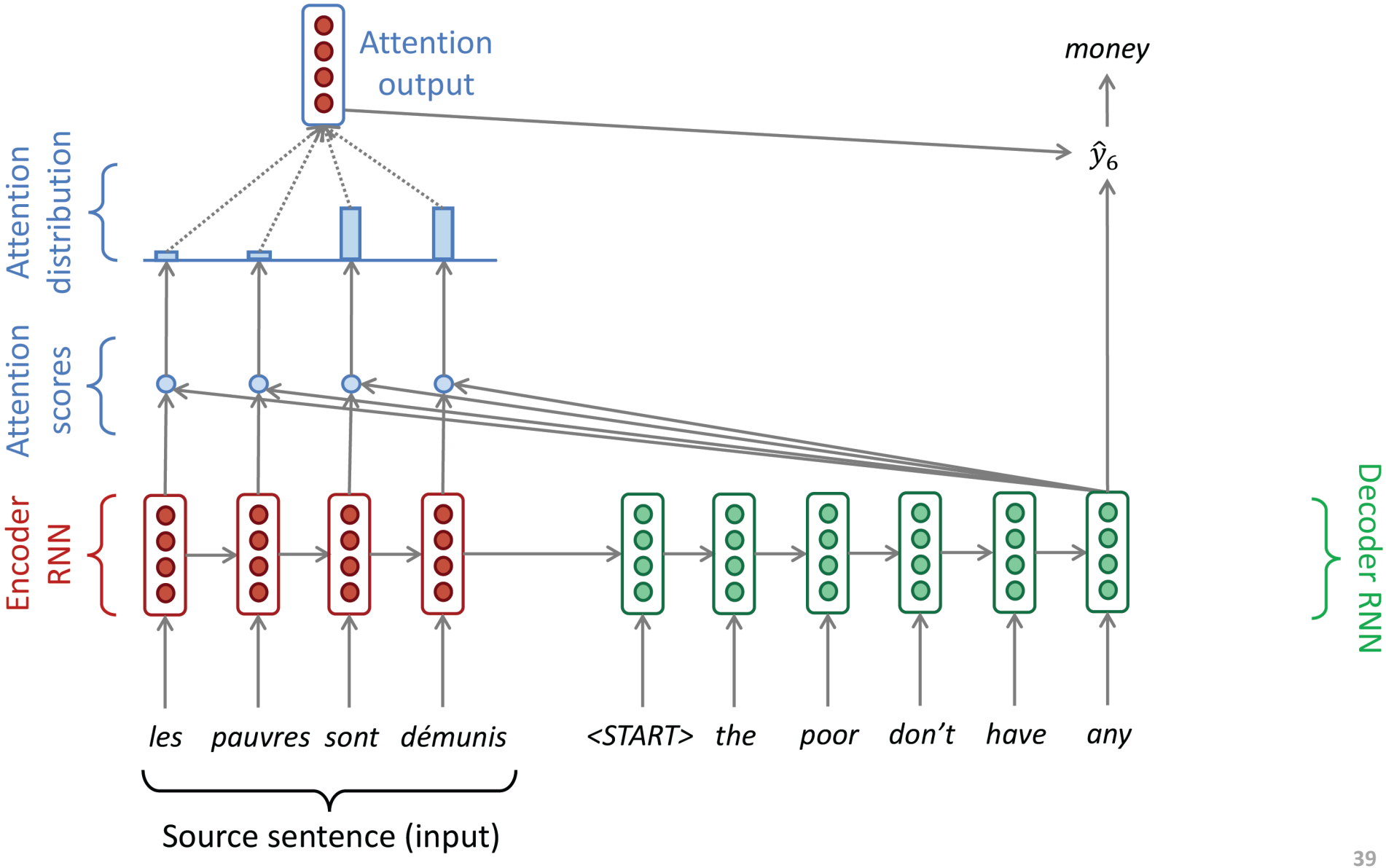
Seq2Seq with Attention



Seq2Seq with Attention



Seq2Seq with Attention



Seq2Seq with Attention

- encoder hidden states $\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_N$ (where each $\boldsymbol{h}_i \in \mathbb{R}^h$)
- at time-step t:
  - decoder hidden state: $\boldsymbol{s}_t \in \mathbb{R}^h$
  - attention scores: $\boldsymbol{e}^{(t)} = [\boldsymbol{s}_t^T \boldsymbol{h}_1, \boldsymbol{s}_t^T \boldsymbol{h}_2, \ldots, \boldsymbol{s}_t^T \boldsymbol{h}_N] \in \mathbb{R}^N$
  - attention distribution: $\boldsymbol{\alpha}^{(t)} = \text{softmax}\,(\boldsymbol{e}^{(t)}) \in \mathbb{R}^N$
  - attention output: weighted sum of encoder hidden states:

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

  - concatenate attention output and decoder hidden state:

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

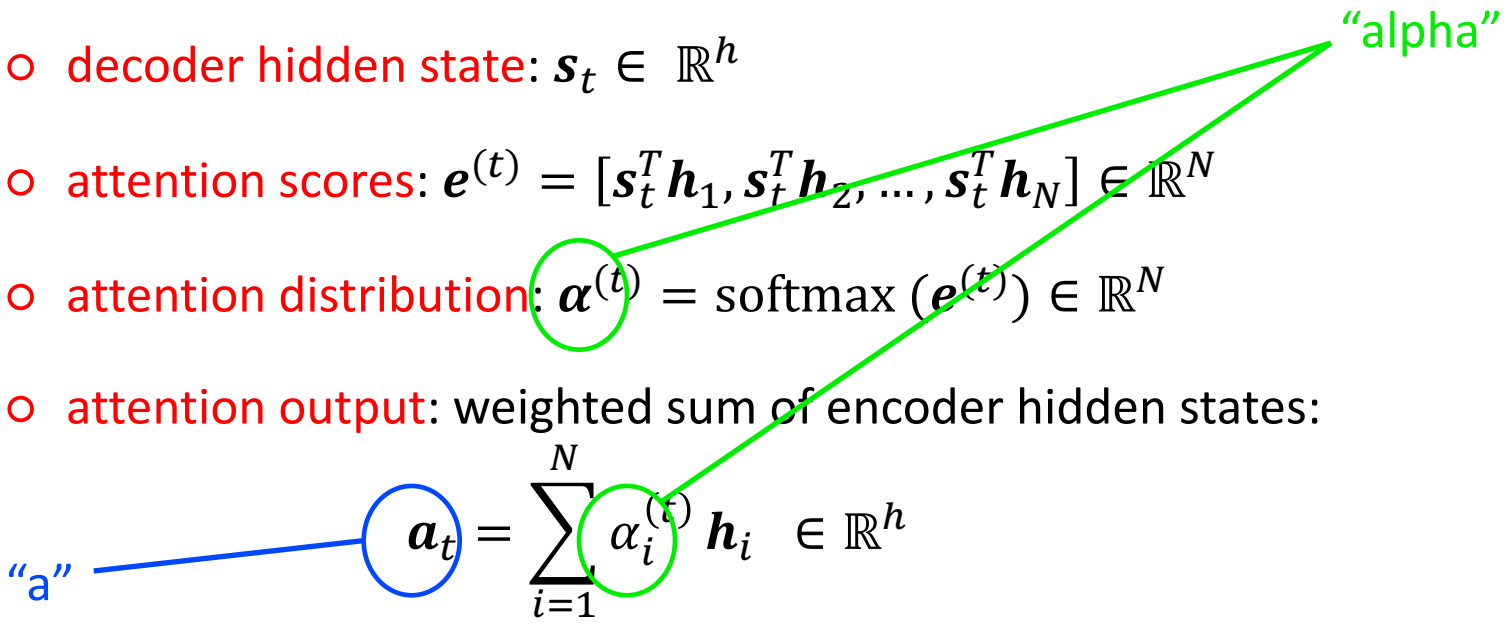
Seq2Seq with Attention

- encoder hidden states $\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_N$ (where each $\boldsymbol{h}_i \in \mathbb{R}^h$)
- at time-step t:
  - decoder hidden state: $\boldsymbol{s}_t \in \mathbb{R}^h$
  - attention scores: $\boldsymbol{e}^{(t)} = [\boldsymbol{s}_t^T \boldsymbol{h}_1, \boldsymbol{s}_t^T \boldsymbol{h}_2, \ldots, \boldsymbol{s}_t^T \boldsymbol{h}_N] \in \mathbb{R}^N$
  - attention distribution: $\boldsymbol{\alpha}^{(t)} = \text{softmax}(\boldsymbol{e}^{(t)}) \in \mathbb{R}^N$
  - attention output: weighted sum of encoder hidden states:

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

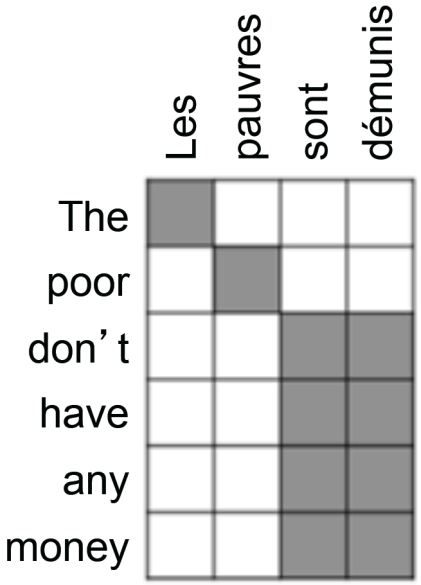
  - concatenate attention output and decoder hidden state:

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



Advantages of Attention

- significantly improves NMT performance
  - very useful to allow decoder to focus on certain parts of the source
- solves the bottleneck problem
  - allows decoder to look directly at source; bypass bottleneck
- helps with vanishing gradient problem
  - Provides shortcut to faraway states
- provides some interpretability
  - by inspecting attention distribution, we can see what the decoder was focusing on
  - we get alignment for free! (we never explicitly trained an alignment system; network just learned alignment by itself)



Bibliography

(1) *<placeholder for upcoming seq2seq chapter in Jurafsky>*

(2) Richard Socher et al: "CS224n: Natural Language Processing with Deep Learning", Lecture Materials (slides and links to background reading) http://web.stanford.edu/class/cs224n/ (URL, May 2018), 2018

(3) https://youtu.be/K-HfpsHPmvw (URL, Aug 2018) (in [2])

(4) Rico Sennrich (University of Edinburgh): Neural Machine Translation: Breaking the Performance Plateau, Talk July 2016; http://www.meta-net.eu/events/meta-forum-2016/slides/09_sennrich.pdf (URL, Aug 2018) (in [2])

(5) https://hackernoon.com/bias-sexist-or-this-is-the-way-it-should-be-ce1f7c8c683c (URL, Aug 2018) (in [2])

(6) http://languagelog.ldc.upenn.edu/nll/?p=35120#more-35120 (URL, Aug 2018) (in [2])

Recommendations for Studying

- minimal approach:

  work with the slides and understand their contents! Think beyond instead of merely memorizing the contents

- standard approach:

  minimal approach + study the corresponding lecture slides from [2] for additional details omitted in our slides

- interested student's approach:

  standard approach + read a selection of the recommended background reading of [2] from lecture 1 up to lecture 9