# machine translation 2:
## seq2seq / decoding / eval

CS 585, Fall 2018
Introduction to Natural Language Processing
http://people.cs.umass.edu/~miyyer/cs585/

Mohit Iyyer
College of Information and Computer Sciences
University of Massachusetts Amherst

*some slides adapted from Richard Socher and Marine Carpuat*

# questions from last time…

- info for project proposal? template now posted!
- teammates for project?
- HW2???
- recorded lecture audio not happening :(
- midterm???
  - will cover text classification / language modeling / word embeddings / sequence labeling / machine translation (including today's lecture)
  - will **not** cover CFGs / parsing.
  - 20% multiple choice, 80% short answer/computational qs
  - 1-page "cheat sheet" allowed, must be hand-written
- Mohit out next lecture and 11/1

# limitations of IBM models

- *discrete* alignments
- all alignments equally likely (model 1 only)
- translation of each *f* word depends only on aligned *e* word!

# Recap: The Noisy Channel Model

- Goal: translation system from French to English
- Have a model $p(e \mid f)$ which estimates conditional probability of any English sentence $e$ given the French sentence $f$. Use the training corpus to set the parameters.
- A Noisy Channel Model has two components:

$$p(e) \quad \textbf{the language model}$$

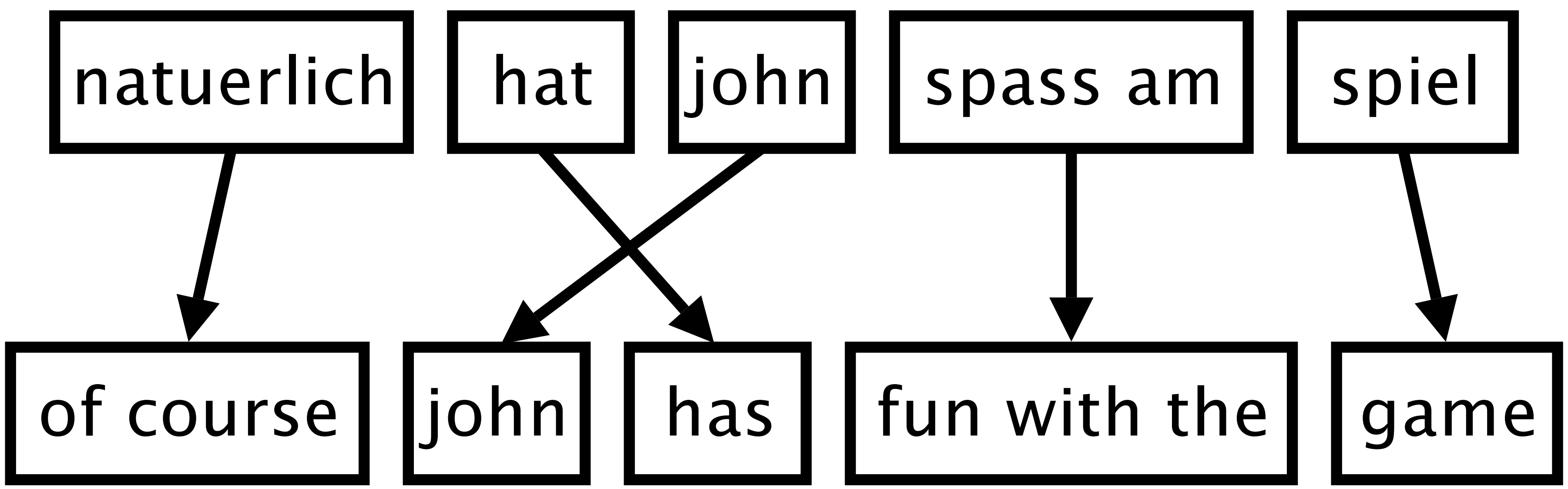
$$p(f \mid e) \quad \textbf{the translation model}$$

- Giving:

$$p(e \mid f) = \frac{p(e, f)}{p(f)} = \frac{p(e)p(f \mid e)}{\sum_e p(e)p(f \mid e)}$$

and

$$\text{argmax}_e p(e \mid f) = \text{argmax}_e p(e)p(f \mid e)$$


natuerlich
hat
john
spass am
spiel
of course
john
has
fun with the
game

# phrase-based MT

- better way of modeling $p(f \mid e)$: *phrase* alignments instead of word alignments

set of phrases in *f*

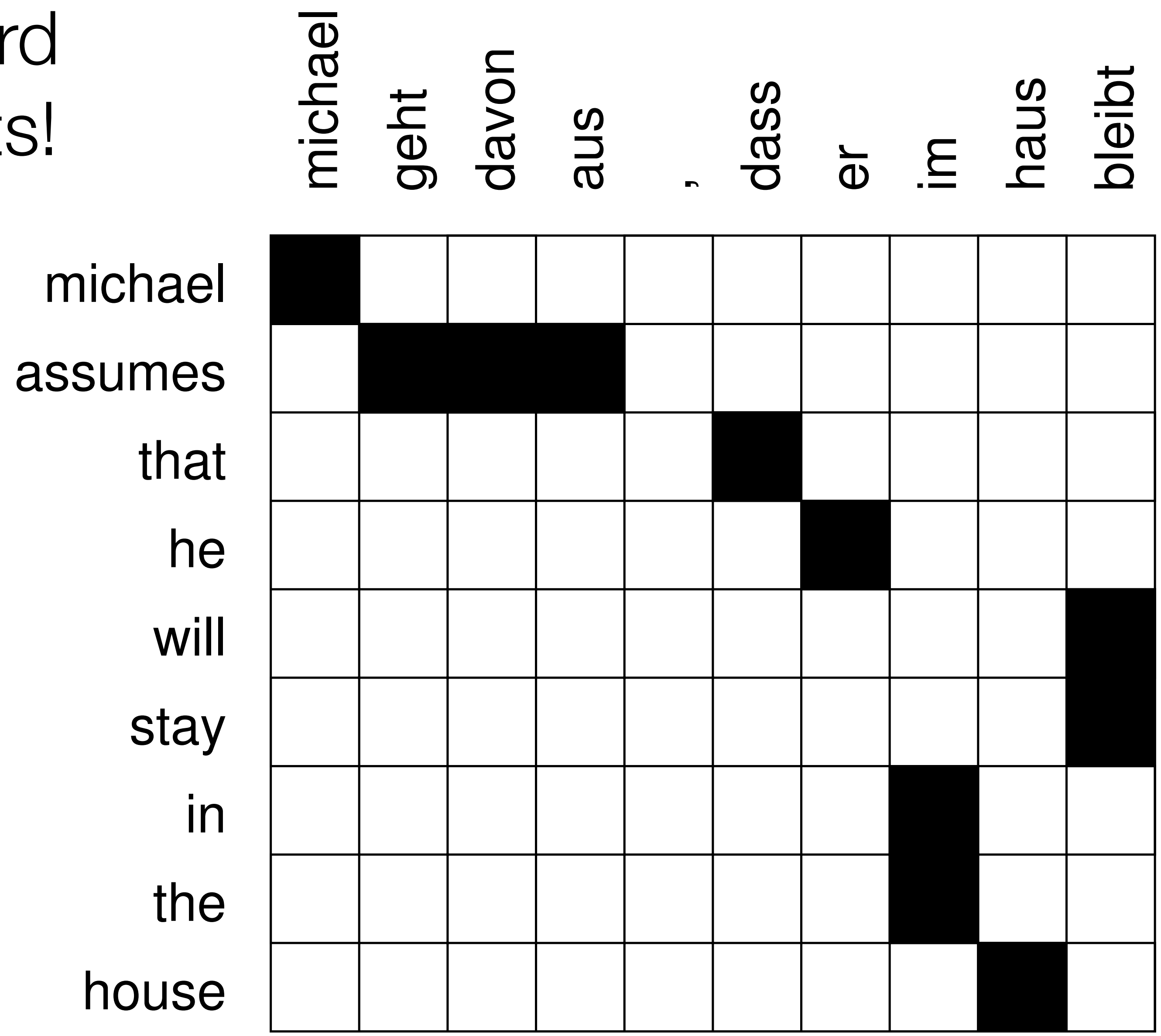
$$p(f \mid e) = \prod_{i=1}^{I} \phi(\bar{f}_i, \bar{e}_i) d(start_i - end_{i-1} - 1)$$

phrase translation probability

reordering probability

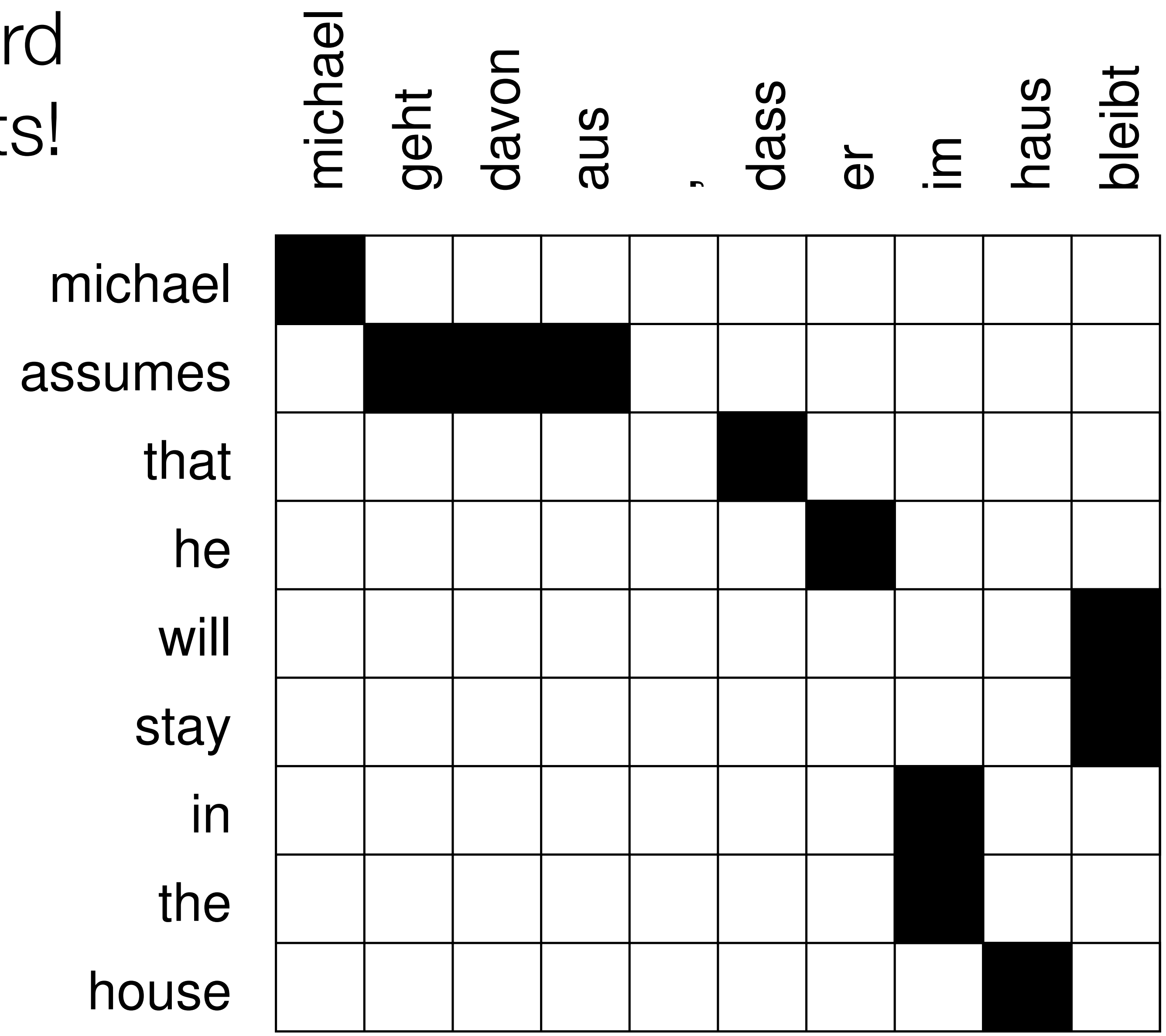
# Phrase alignment from word alignment!

use IBM models
to get word
alignments!



# Phrase alignment from word alignment!

use IBM models
to get word
alignments!




assumes / geht davon aus
assumes that / geht davon aus , dass

- Phrase translations for den Vorschlag learned from the Europarl corpus:

| **English** | $\phi(\bar{e}\|\bar{f})$ | **English** | $\phi(\bar{e}\|\bar{f})$ |
|---|---|---|---|
| the proposal | 0.6227 | the suggestions | 0.0114 |
| 's proposal | 0.1068 | the proposed | 0.0114 |
| a proposal | 0.0341 | the motion | 0.0091 |
| the idea | 0.0250 | the idea of | 0.0091 |
| this proposal | 0.0227 | the proposal , | 0.0068 |
| proposal | 0.0205 | its proposal | 0.0068 |
| of the proposal | 0.0159 | it | 0.0068 |
| the proposals | 0.0159 | ... | ... |

in general, we learn a *phrase table* to store these translation probabilities. what are some limitations of phrase-based MT?

# today: neural MT

- instead of using the noisy channel model to decompose $p(e \mid f) \propto p(f \mid e)p(e)$, let's directly model $p(e \mid f)$

$$p(e \mid f) = p(e_1, e_2, \ldots, e_l \mid f)$$

$$= p(e_1 \mid f) \cdot p(e_2 \mid e_1, f) \cdot p(e_3 \mid e_2, e_1, f) \cdot \ldots$$

$$= \prod_{i=1}^{L} p(e_i \mid e_1, \ldots, e_{i-1}, f)$$

this is a *conditional language model*. how is this different than the LMs we saw in the IBM models?

# seq2seq models

- use two different RNNs to model $\prod_{i=1}^{L} p(e_i \,|\, e_1, \ldots, e_{i-1}, f)$
- first we have the *encoder*, which encodes the foreign sentence *f*
- then, we have the *decoder,* which produces the English sentence *e*

# Reminder: RNN language models!

output distribution

$$\hat{y} = \text{softmax}(W_2 h^{(t)} + b_2)$$

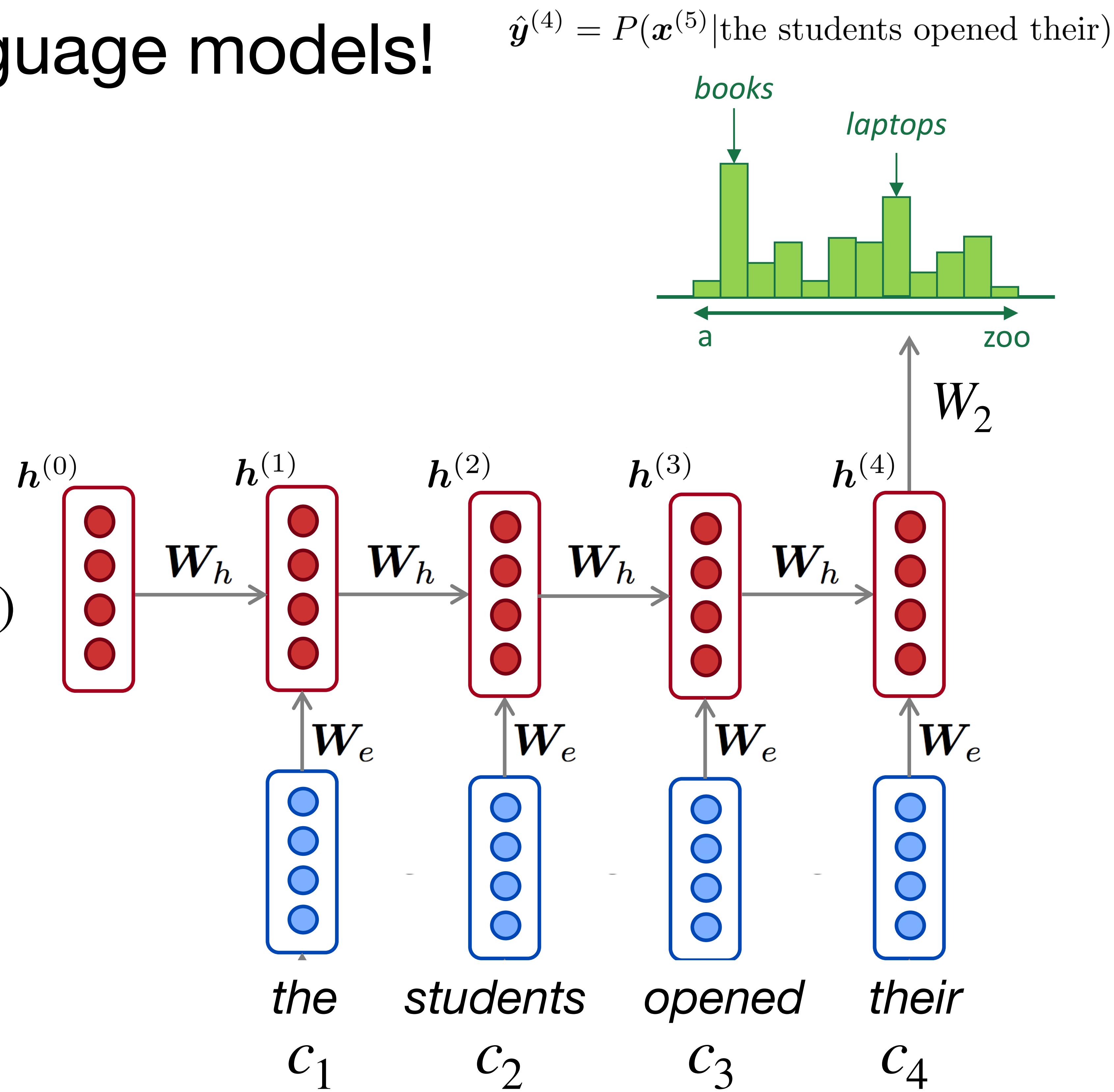
hidden states

$$h^{(t)} = f(W_h h^{(t-1)} + W_e c_t + b_1)$$

$h^{(0)}$ is initial hidden state!

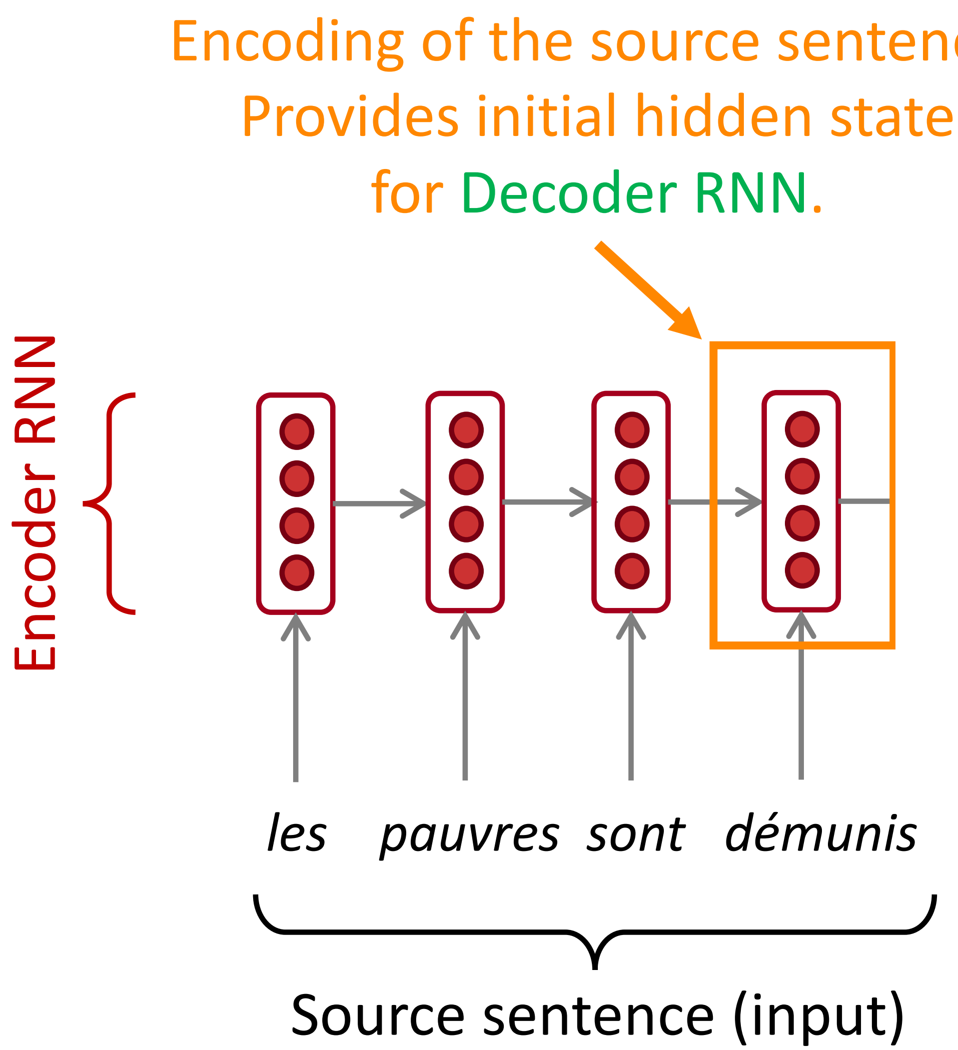
word embeddings

$$c_1, c_2, c_3, c_4$$



# Neural Machine Translation (NMT)

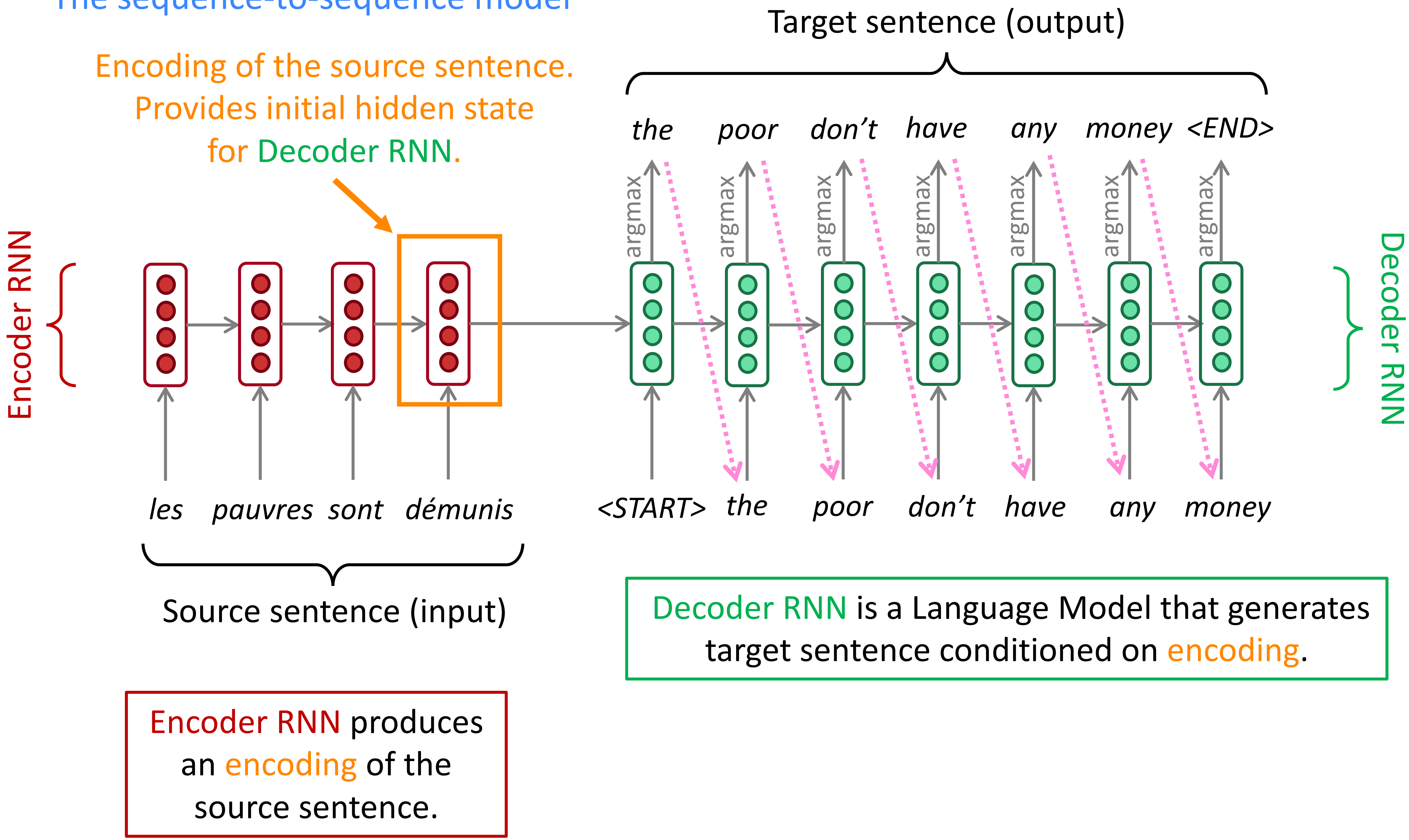
The sequence-to-sequence model



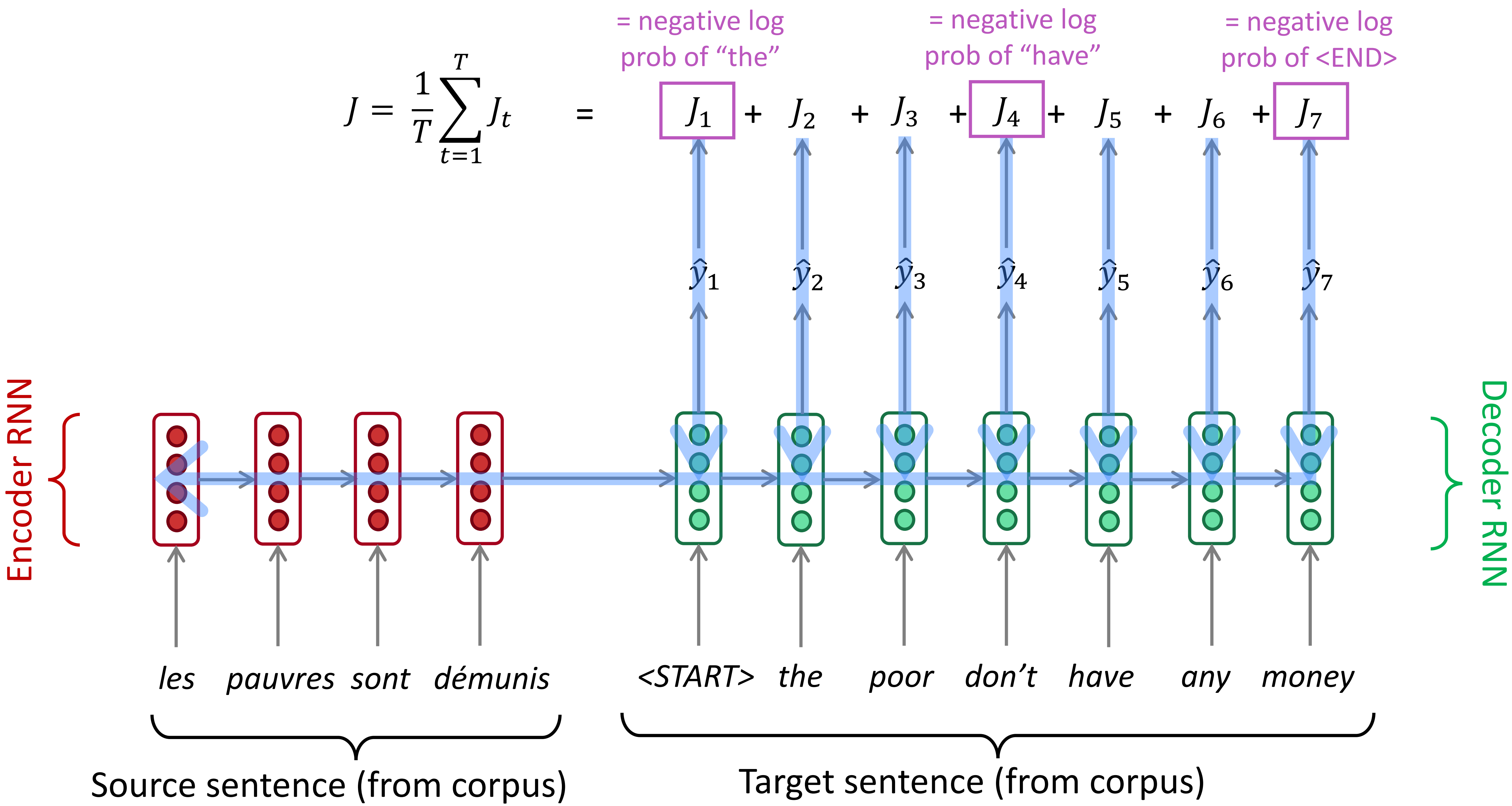

Encoder RNN produces an encoding of the source sentence.

# Neural Machine Translation (NMT)

The sequence-to-sequence model

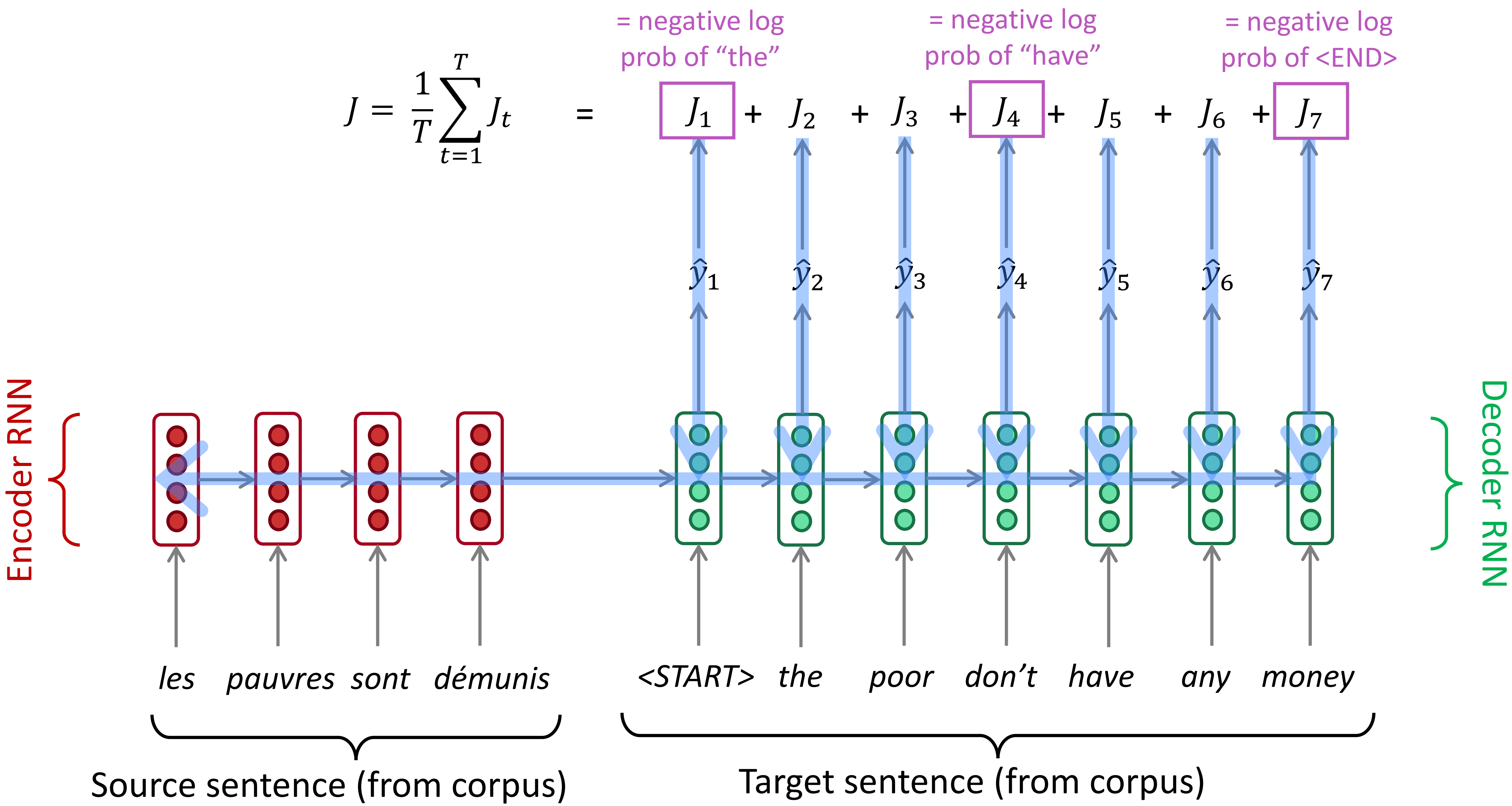


# Training a Neural Machine Translation system




what are the parameters of this model?

# Training a Neural Machine Translation system




what are the parameters of this model?

$$W_h^{enc}, W_e^{enc}, C^{enc}, W_h^{dec}, W_e^{dec}, C^{dec}, W_{out}$$



C is word embedding matrix

# decoding

- given that we trained a seq2seq model, how do we find the most probable English sentence?
- more concretely, how do we find

$$\arg\max \prod_{i=1}^{L} p(e_i \mid e_1, \ldots, e_{i-1}, f)$$

- can we enumerate all possible English sentences *e*?

# decoding

- given that we trained a seq2seq model, how do we find the most probable English sentence?
- more concretely, how do we find

$$\arg\max \prod_{i=1}^{L} p(e_i \mid e_1, \ldots, e_{i-1}, f)$$

- can we enumerate all possible English sentences *e*?
- can we use the Viterbi algorithm like we did for HMMs?

# decoding

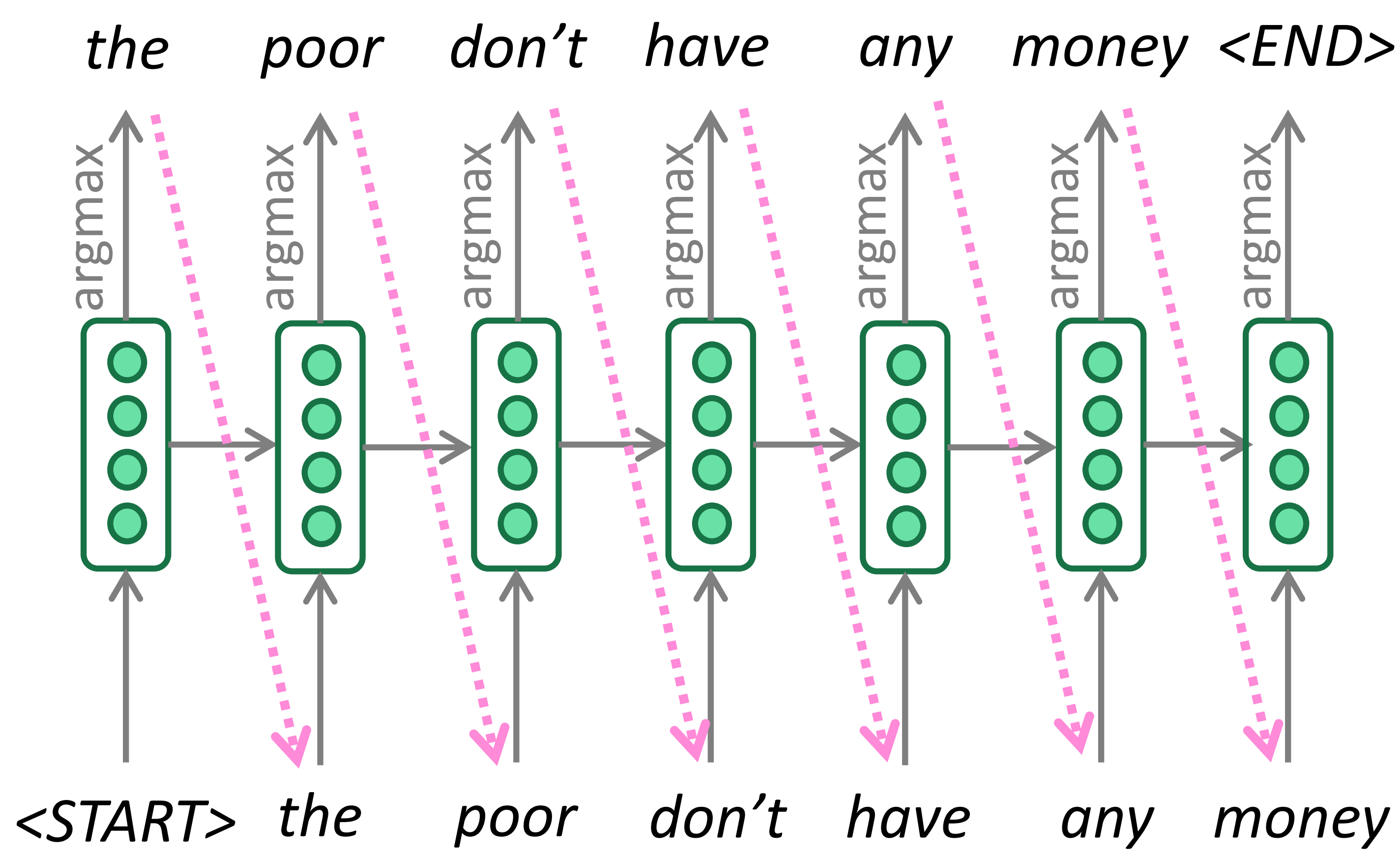
- given that we trained a seq2seq model, how do we find the most probable English sentence?
- more concretely, how do we find

$$\arg\max \prod_{i=1}^{L} p(e_i \mid e_1, \ldots, e_{i-1}, f)$$

- can we enumerate all possible English sentences *e*?
- can we use the Viterbi algorithm like we did for HMMs?

# decoding

- given that we trained a seq2seq model, how do we find the most probable English sentence?
- easiest option: **greedy decoding**




issues?

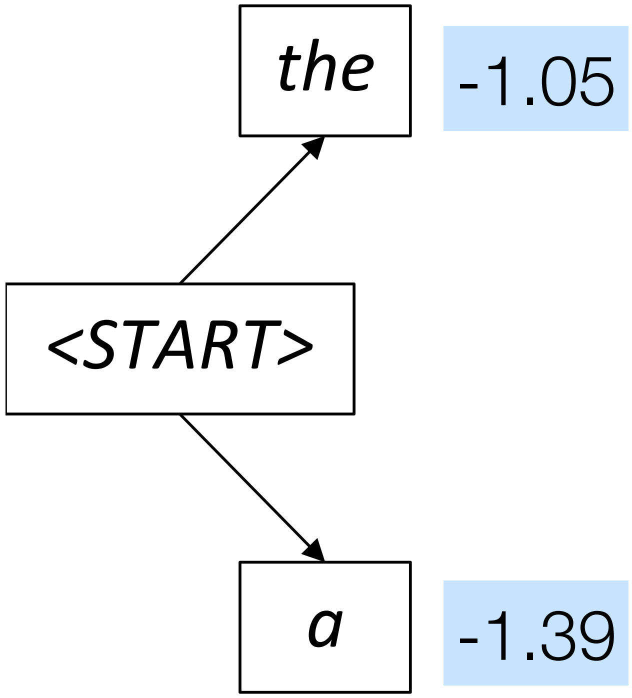
# Beam search

- in greedy decoding, we cannot go back and revise previous decisions!
  - *les pauvres sont démunis (the poor don't have any money)*
  - → *the* ____
  - → *the poor* ____
  - → *the poor are* ____
- fundamental idea of beam search: explore several different hypotheses instead of just a single one
  - keep track of *k* most probable partial translations at each decoder step instead of just one!

    the beam size *k* is usually 5-10

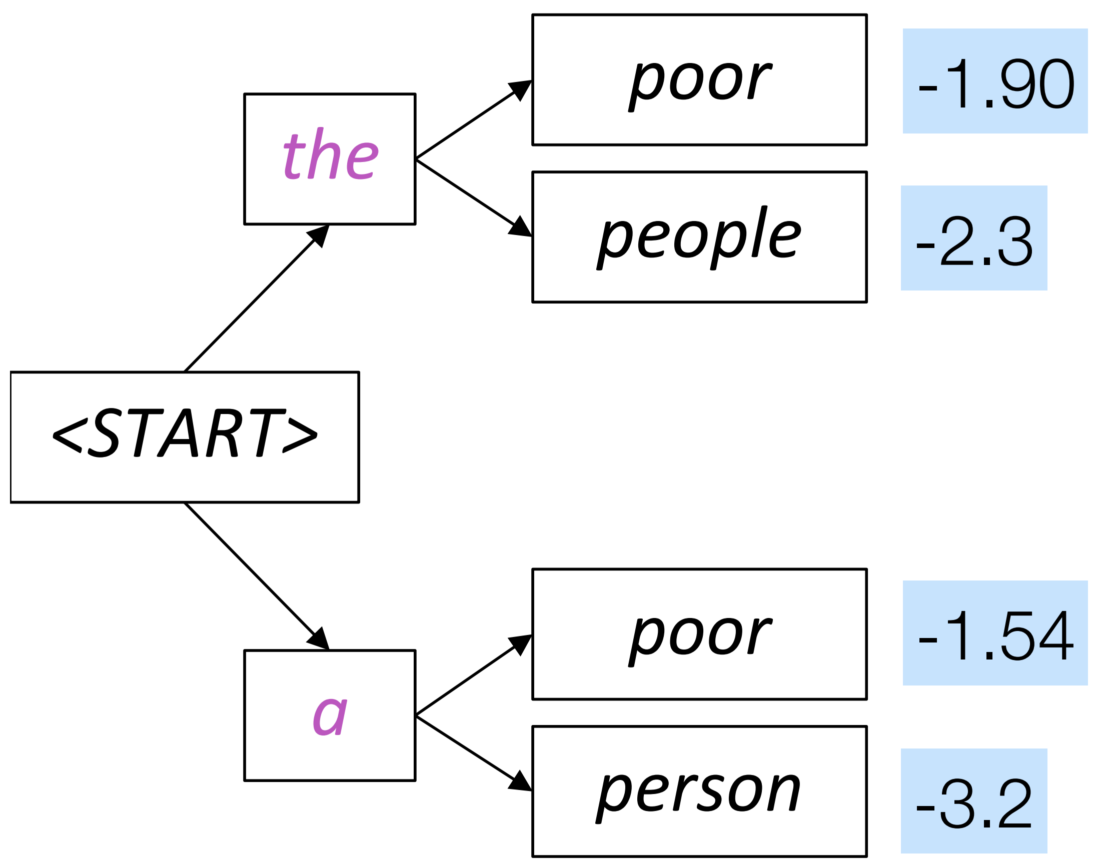
# Beam search decoding: example

Beam size = 2



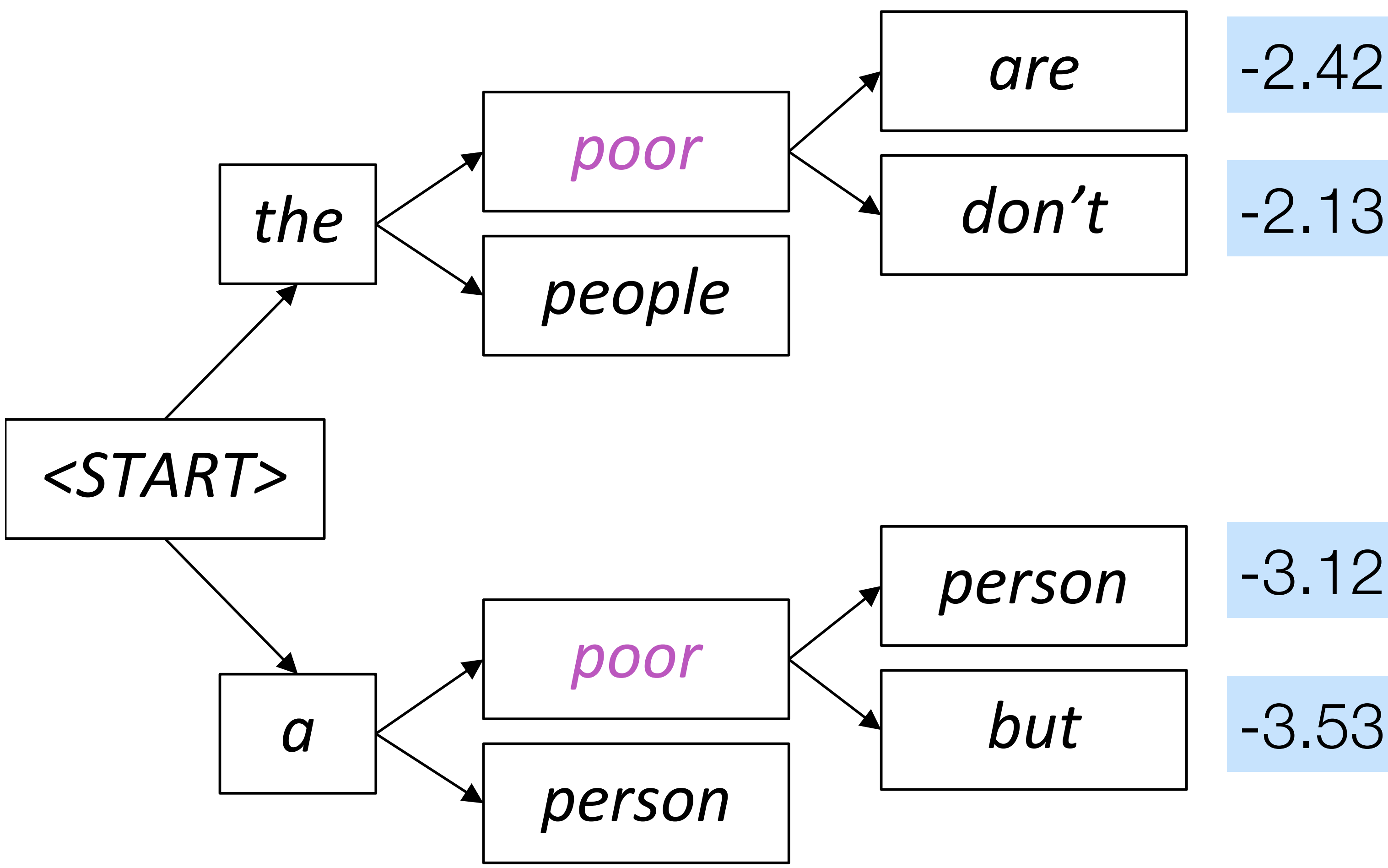
# Beam search decoding: example

Beam size = 2



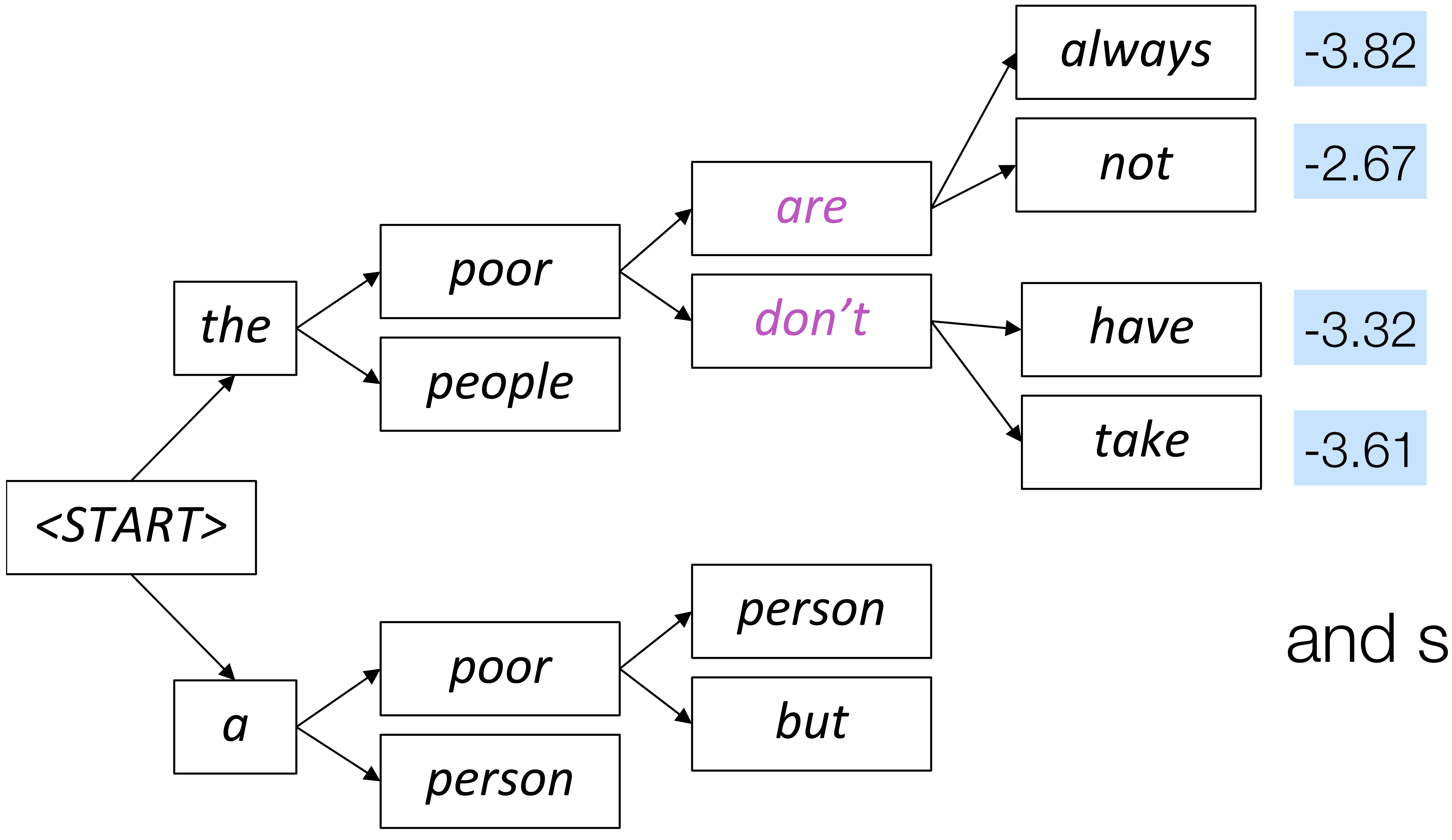
# Beam search decoding: example

Beam size = 2



# Beam search decoding: example

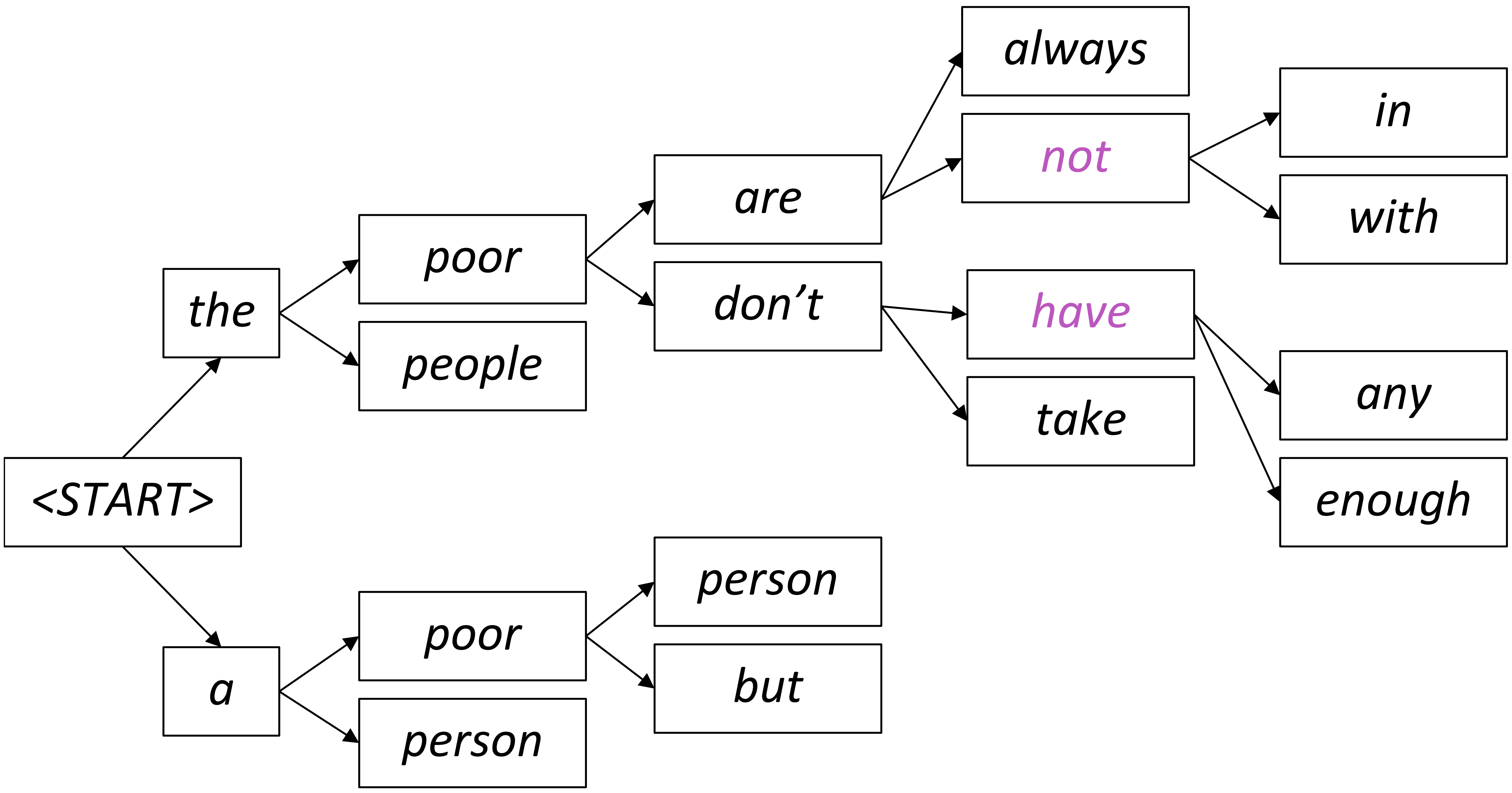
Beam size = 2




and so on…

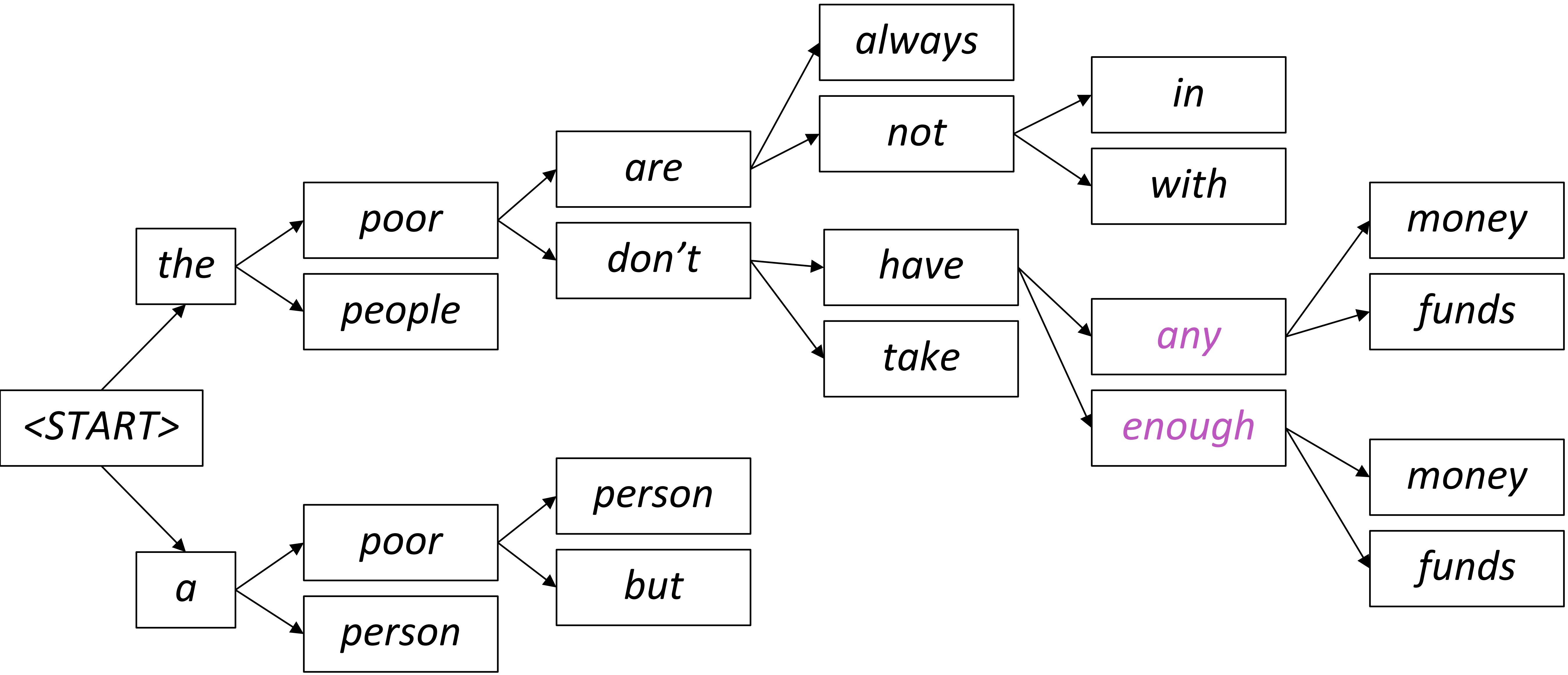
# Beam search decoding: example

Beam size = 2



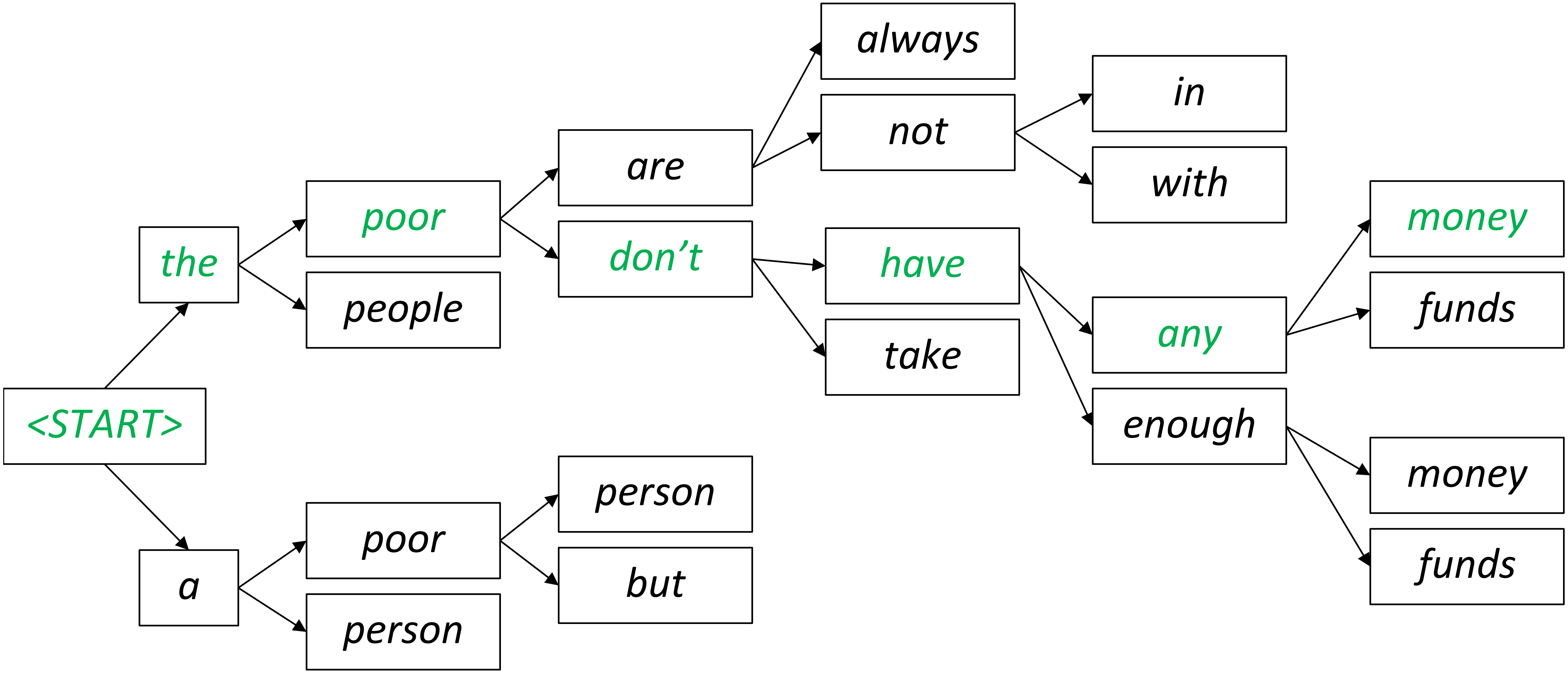
# Beam search decoding: example

Beam size = 2



# Beam search decoding: example

Beam size = 2



does beam search always produce the *best* translation (i.e., does it always find the argmax?)

how many probabilities do we need to evaluate at each time step with a beam size of $k$?

what are the termination conditions for beam search?

# Advantages of NMT

Compared to SMT, NMT has many advantages:

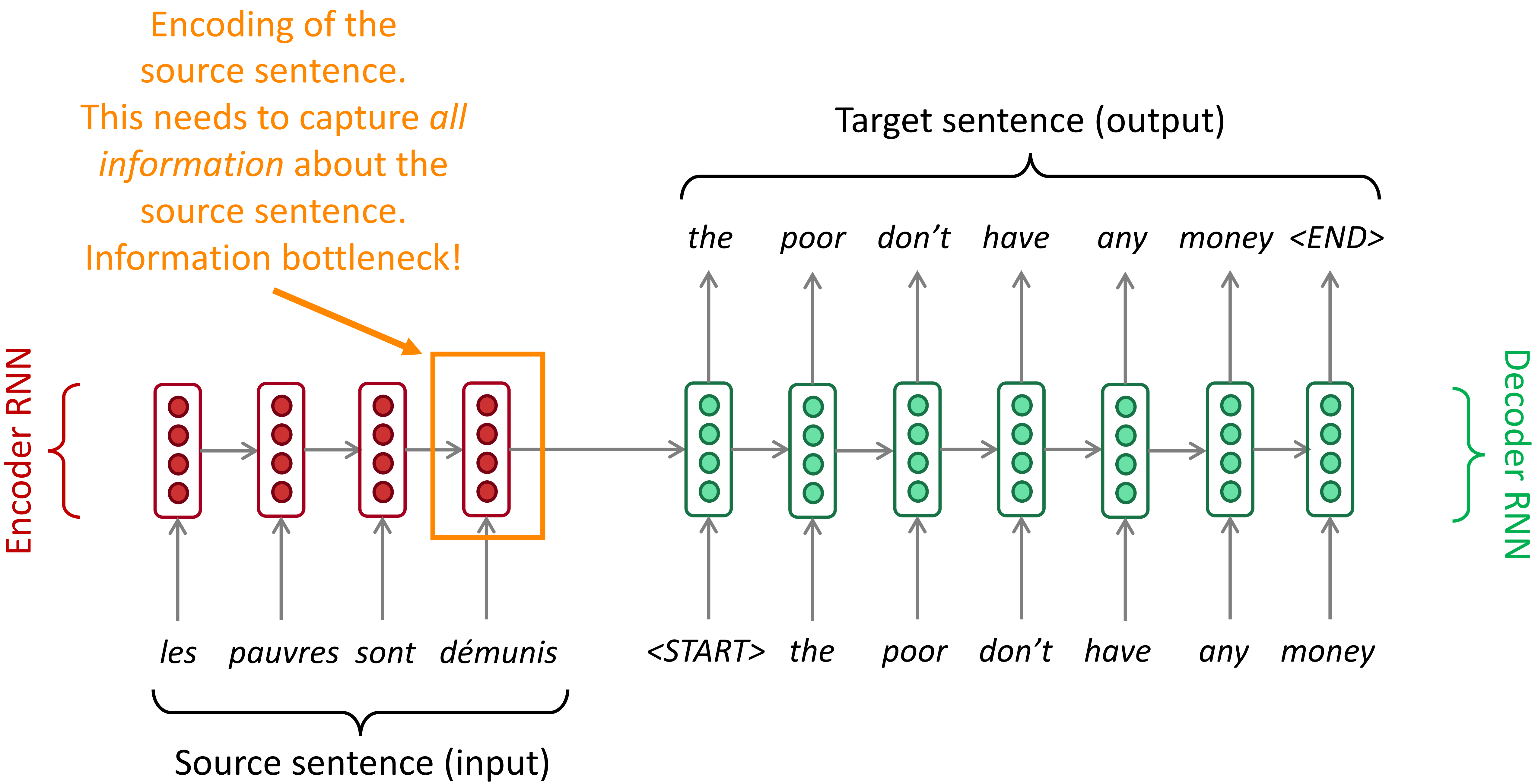
- Better performance
  - More fluent
  - Better use of context
  - Better use of phrase similarities
- A single neural network to be optimized end-to-end
  - No subcomponents to be individually optimized
- Requires much less human engineering effort
  - No feature engineering
  - Same method for all 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

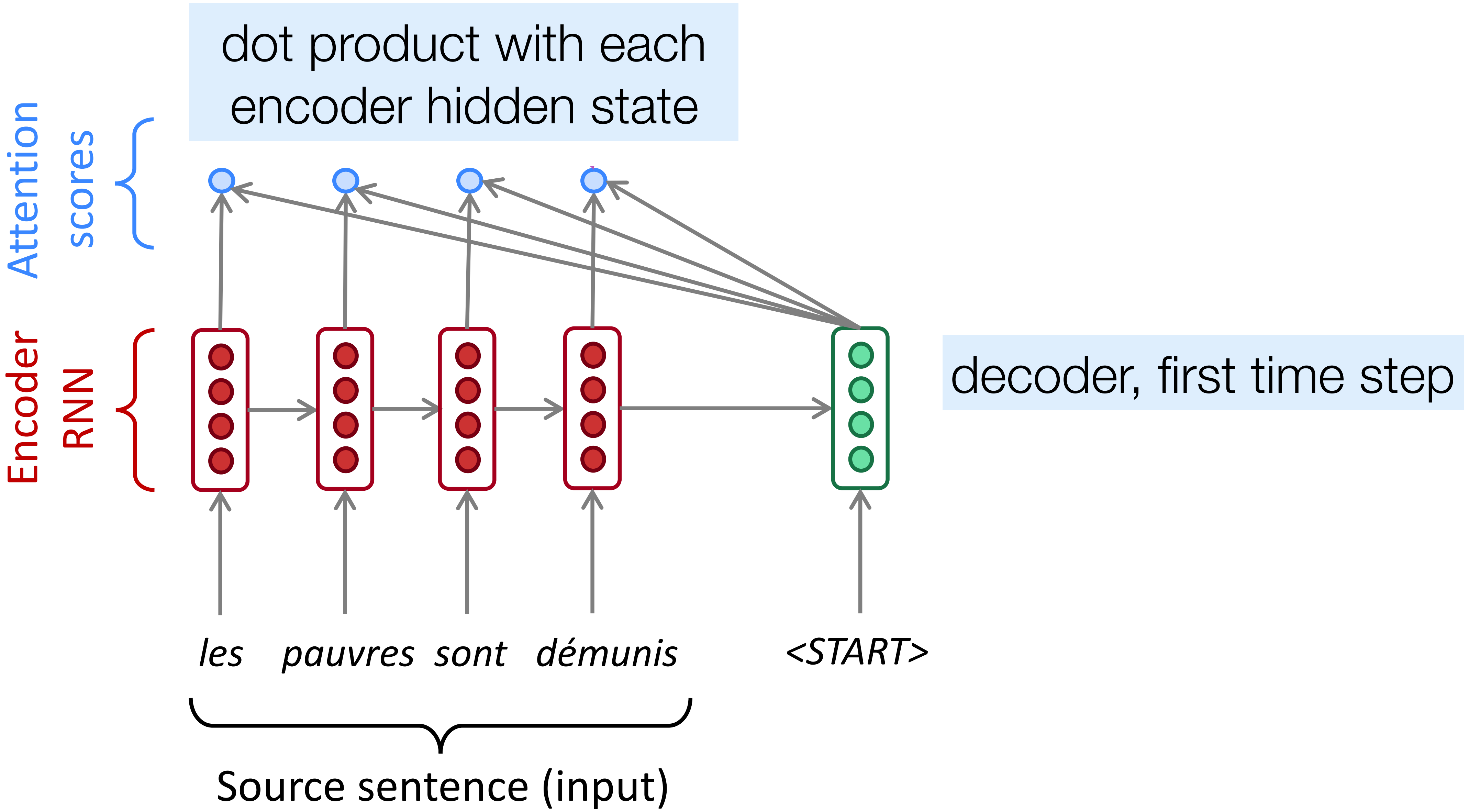
# Sequence-to-sequence: the bottleneck problem



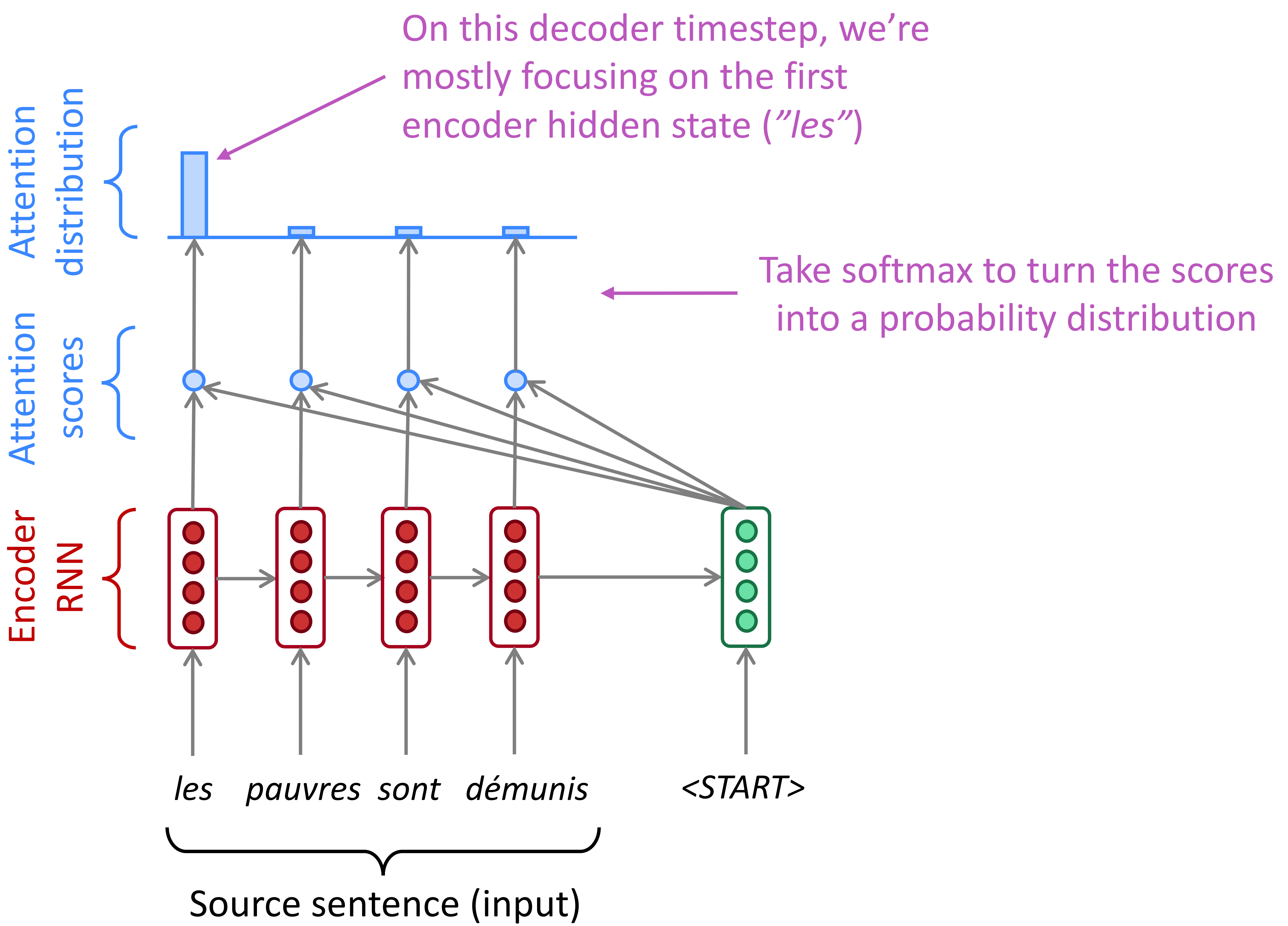
# The solution: **attention**

- **Attention mechanisms** allow the decoder to focus on a particular part of the source sequence at each time step
  - Conceptually similar to *alignments*

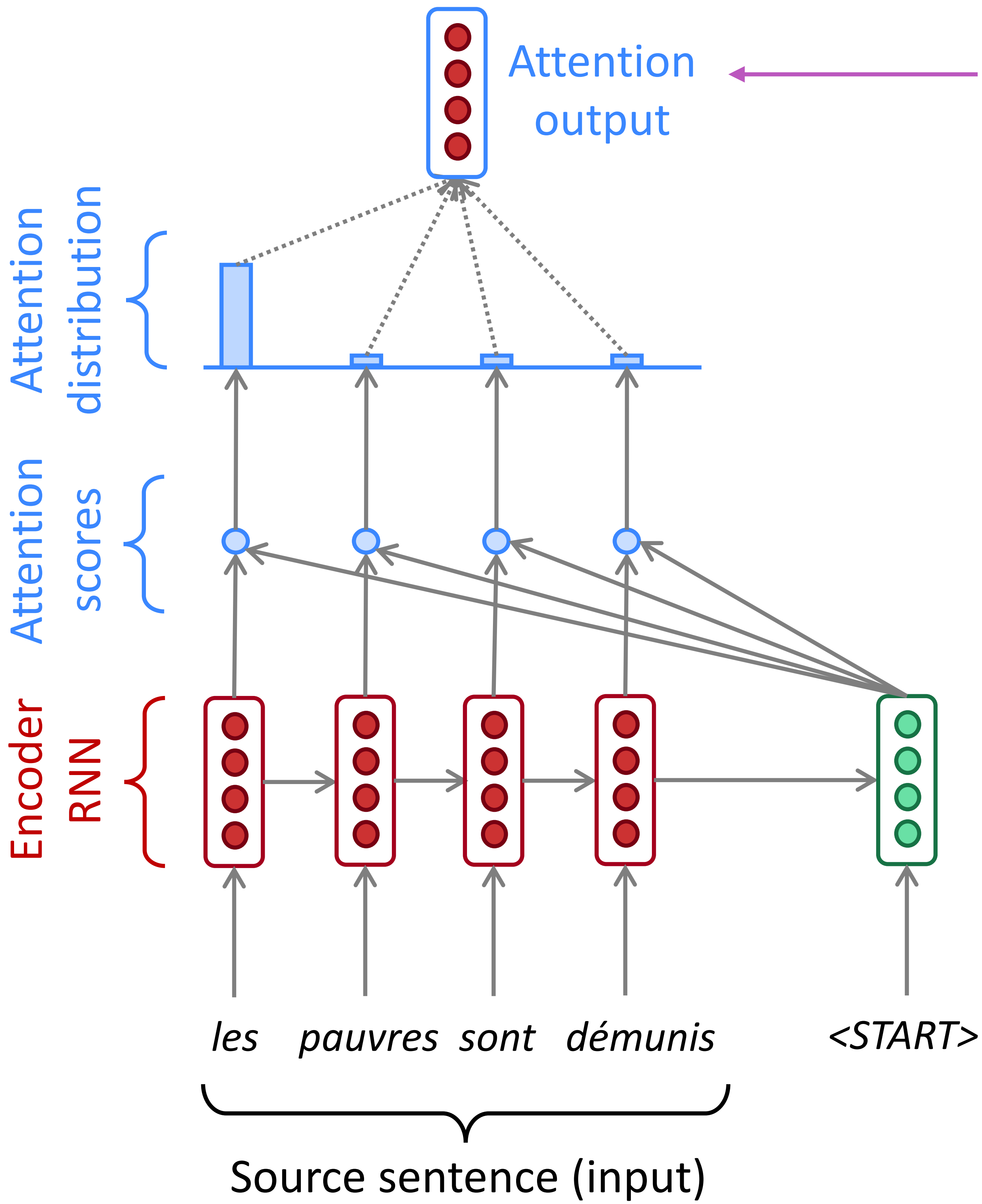
# Sequence-to-sequence with attention



# Sequence-to-sequence with attention



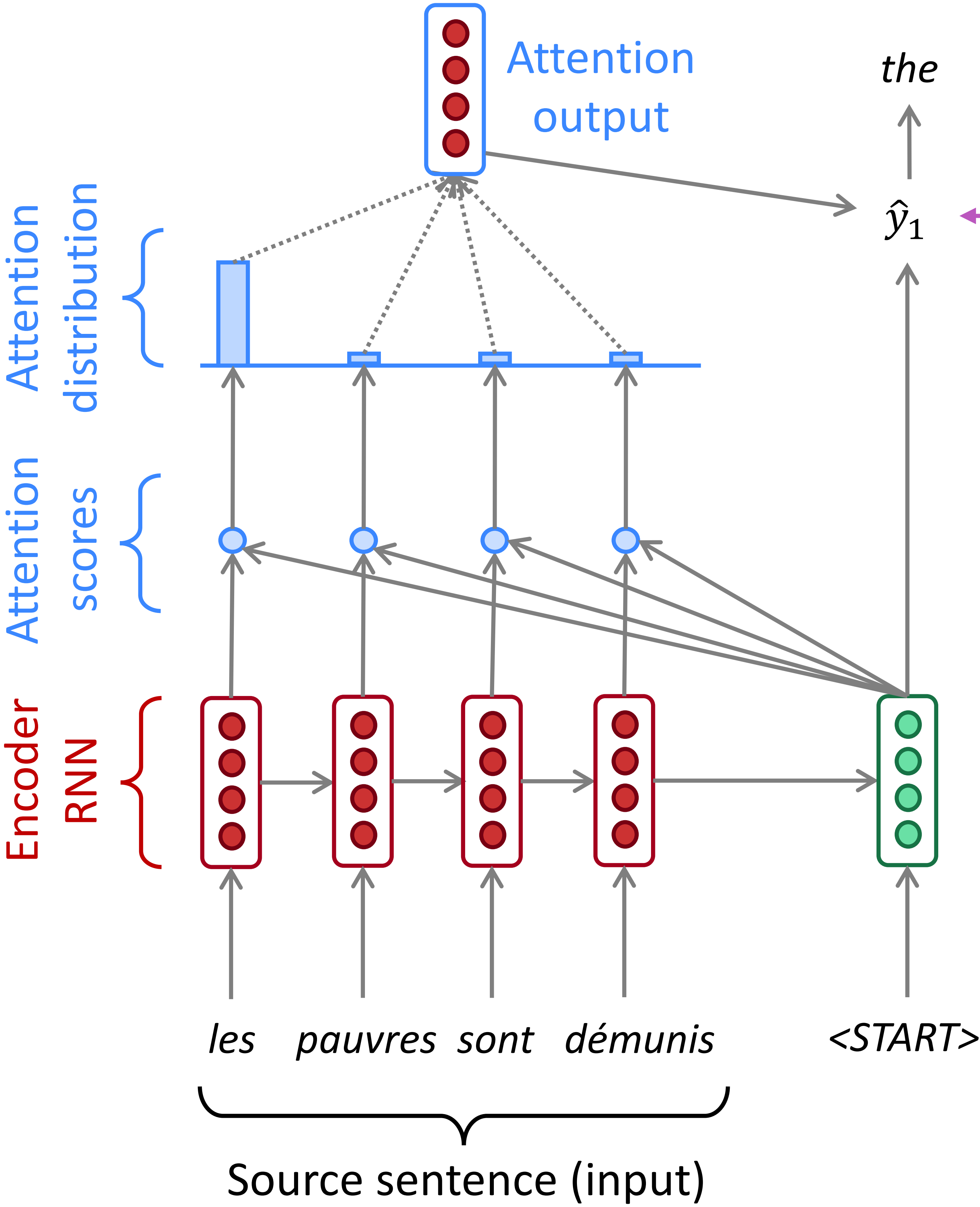
# Sequence-to-sequence with attention




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

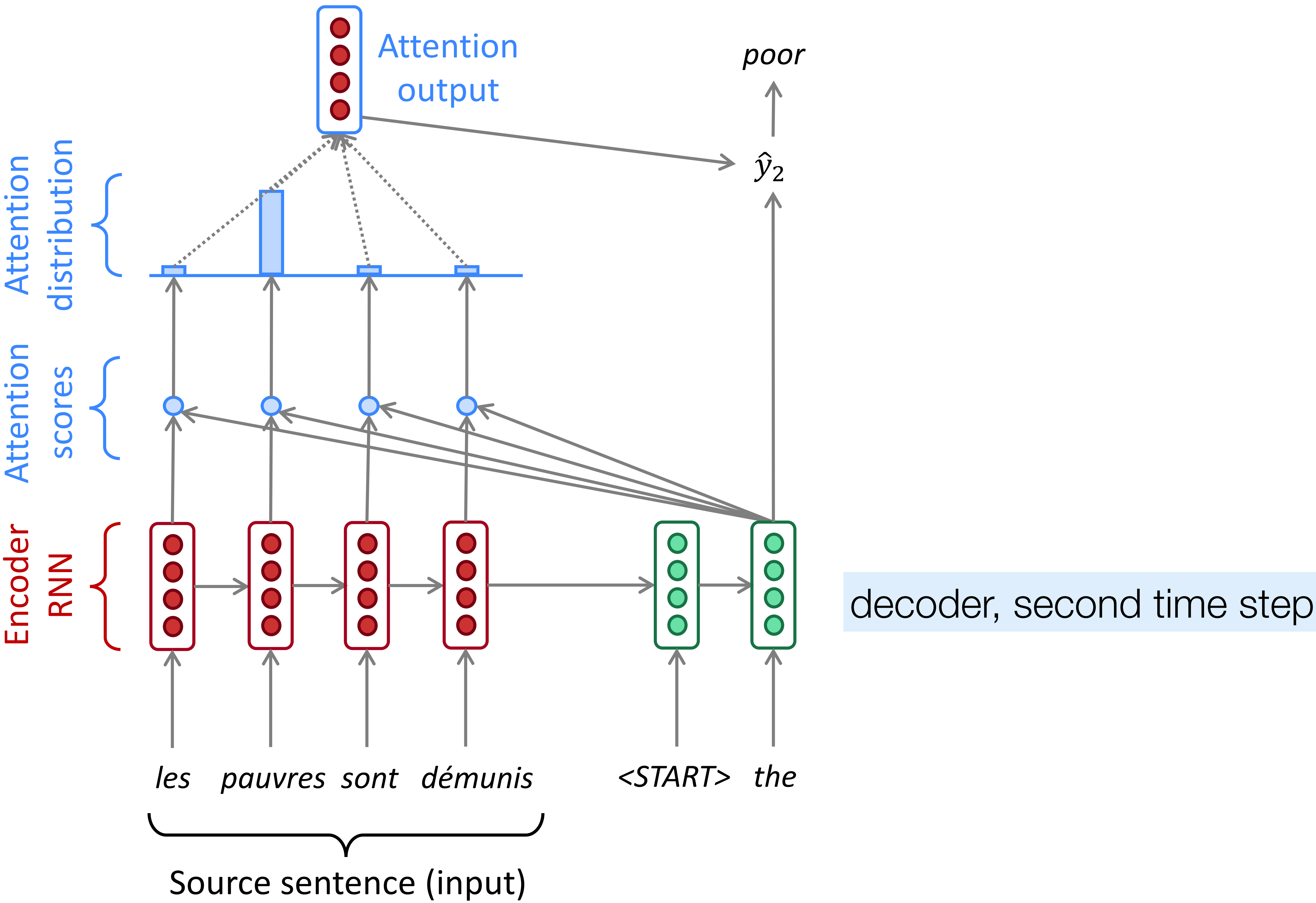
The attention output mostly contains information the hidden states that received high attention.

# Sequence-to-sequence with attention



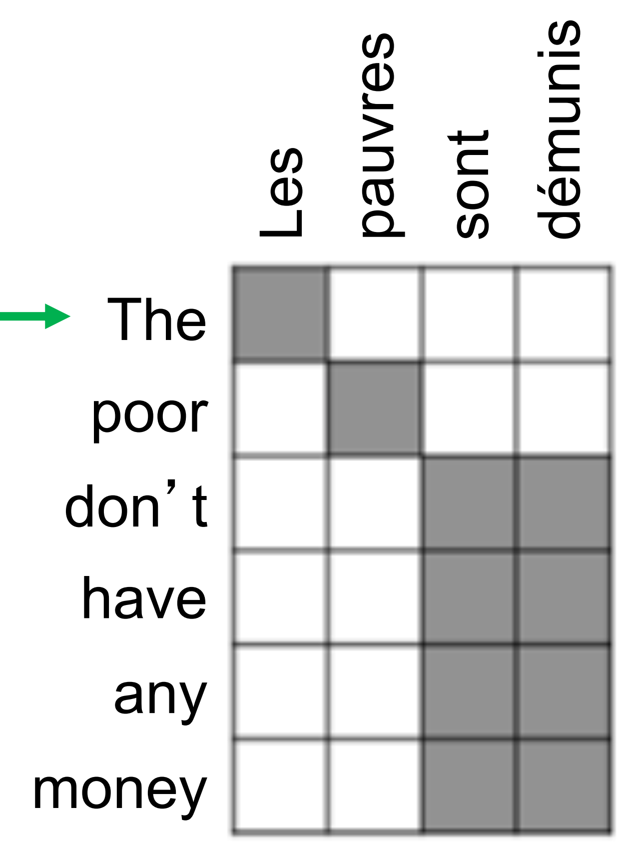

Concatenate attention output with decoder hidden state, then use to compute $\hat{y}_1$ as before

# Sequence-to-sequence with attention



# 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



onto evaluation…

# How good is a translation? Problem: no single right answer

这个 机场 的 安全 工作 由 以色列 方面 负责 .

Israeli officials are responsible for airport security.
Israel is in charge of the security at this airport.
The security work for this airport is the responsibility of the Israel government.
Israeli side was in charge of the security of this airport.
Israel is responsible for the airport's security.
Israel is responsible for safety work at this airport.
Israel presides over the security of the airport.
Israel took charge of the airport security.
The safety of this airport is taken charge of by Israel.
This airport's security is the responsibility of the Israeli security officials.

# Evaluation

- How good is a given machine translation system?
- Many different translations acceptable
- Evaluation metrics
  - Subjective judgments by human evaluators
  - Automatic evaluation metrics
  - Task-based evaluation

# Automatic Evaluation Metrics

- Goal: computer program that computes quality of translations
- Advantages: low cost, optimizable, consistent
- Basic strategy
  - Given: MT output
  - Given: human reference translation
  - Task: compute similarity between them

# Precision and Recall of Words

SYSTEM A: Israeli officials ~~responsibility~~ ~~of~~ airport ~~safety~~

REFERENCE: Israeli officials are responsible for airport security

Precision

$$\frac{\textit{correct}}{\textit{output-length}} = \frac{3}{6} = 50\%$$

Recall

$$\frac{\textit{correct}}{\textit{reference-length}} = \frac{3}{7} = 43\%$$

F-measure

$$\frac{\textit{precision} \times \textit{recall}}{(\textit{precision} + \textit{recall})/2} = \frac{.5 \times .43}{(.5 + .43)/2} = 46\%$$

# Precision and Recall of Words

SYSTEM A: Israeli officials ~~responsibility~~ ~~of~~ airport ~~safety~~

REFERENCE: Israeli officials are responsible for airport security

SYSTEM B: airport security Israeli officials are responsible

| Metric | System A | System B |
|---|---|---|
| precision | 50% | 100% |
| recall | 43% | 100% |
| f-measure | 46% | 100% |

flaw: no penalty for reordering

# BLEU
# Bilingual Evaluation Understudy

N-gram overlap between machine translation output and reference translation

Compute precision for n-grams of size 1 to 4

Add brevity penalty (for too short translations)

$$\textsf{BLEU} = \min\left(1, \frac{\textit{output-length}}{\textit{reference-length}}\right) \left(\prod_{i=1}^{4} \textit{precision}_i\right)^{\frac{1}{4}}$$

Typically computed over the entire corpus, not single sentences

# Multiple Reference Translations

To account for variability, use multiple reference translations

- n-grams may match in any of the references
- closest reference length used

Example

SYSTEM: $\boxed{\text{Israeli officials}}$ $\boxed{\text{responsibility of}}$ $\boxed{\text{airport}}$ safety

2-GRAM MATCH 2-GRAM MATCH 1-GRAM

REFERENCES:

<u>Israeli officials</u> are responsible for <u>airport</u> security
Israel is in charge <u>of</u> the security at this <u>airport</u>
The security work for this <u>airport</u> is the <u>responsibility of</u> the Israel government
<u>Israeli</u> side was in charge <u>of</u> the security of this <u>airport</u>

# BLEU examples

SYSTEM A: Israeli officials responsibility of airport safety
2-GRAM MATCH 1-GRAM MATCH

REFERENCE: Israeli officials are responsible for airport security

SYSTEM B: airport security Israeli officials are responsible
2-GRAM MATCH 4-GRAM MATCH

| Metric | System A | System B |
|---|---|---|
| precision (1gram) | 3/6 | 6/6 |
| precision (2gram) | 1/5 | 4/5 |
| precision (3gram) | 0/4 | 2/4 |
| precision (4gram) | 0/3 | 1/3 |
| brevity penalty | 6/7 | 6/7 |
| BLEU | 0% | 52% |

# neural MT usually > phrase-based MT!

| Novel | PBSMT | NMT | Relative improvement |
|---|---|---|---|
| Auster's *Sunset Park* (2010) | 0.3735 | 0.3851 | 3.11% |
| Collins' *Hunger Games #3* (2010) | 0.3322 | 0.3787 | 14.00% |
| Golding's *Lord of the Flies* (1954) | 0.2196 | 0.2451 | 11.61% |
| Hemingway's *The Old Man and the Sea* (1952) | 0.2559 | 0.2829 | 10.55% |
| Highsmith's *Ripley Under Water* (1991) | 0.2485 | 0.2762 | 11.15% |
| Hosseini's *A Thousand Splendid Suns* (2007) | 0.3422 | 0.3715 | 8.56% |
| Joyce's *Ulysses* (1922) | 0.1611 | 0.1794 | 11.36% |
| Kerouac's *On the Road* (1957) | 0.3248 | 0.3572 | 9.98% |
| Orwell's *1984* (1949) | 0.2978 | 0.3306 | 11.01% |
| Rowling's *Harry Potter #7* (2007) | 0.3558 | 0.3892 | 9.39% |
| Salinger's *The Catcher in the Rye* (1951) | 0.3255 | 0.3695 | 13.52% |
| Tolkien's *The Lord of the Rings #3* (1955) | 0.2537 | 0.2888 | 13.84% |
| Average | 0.2909 | 0.3212 | 10.67% |

English-to-Catalan novel translation



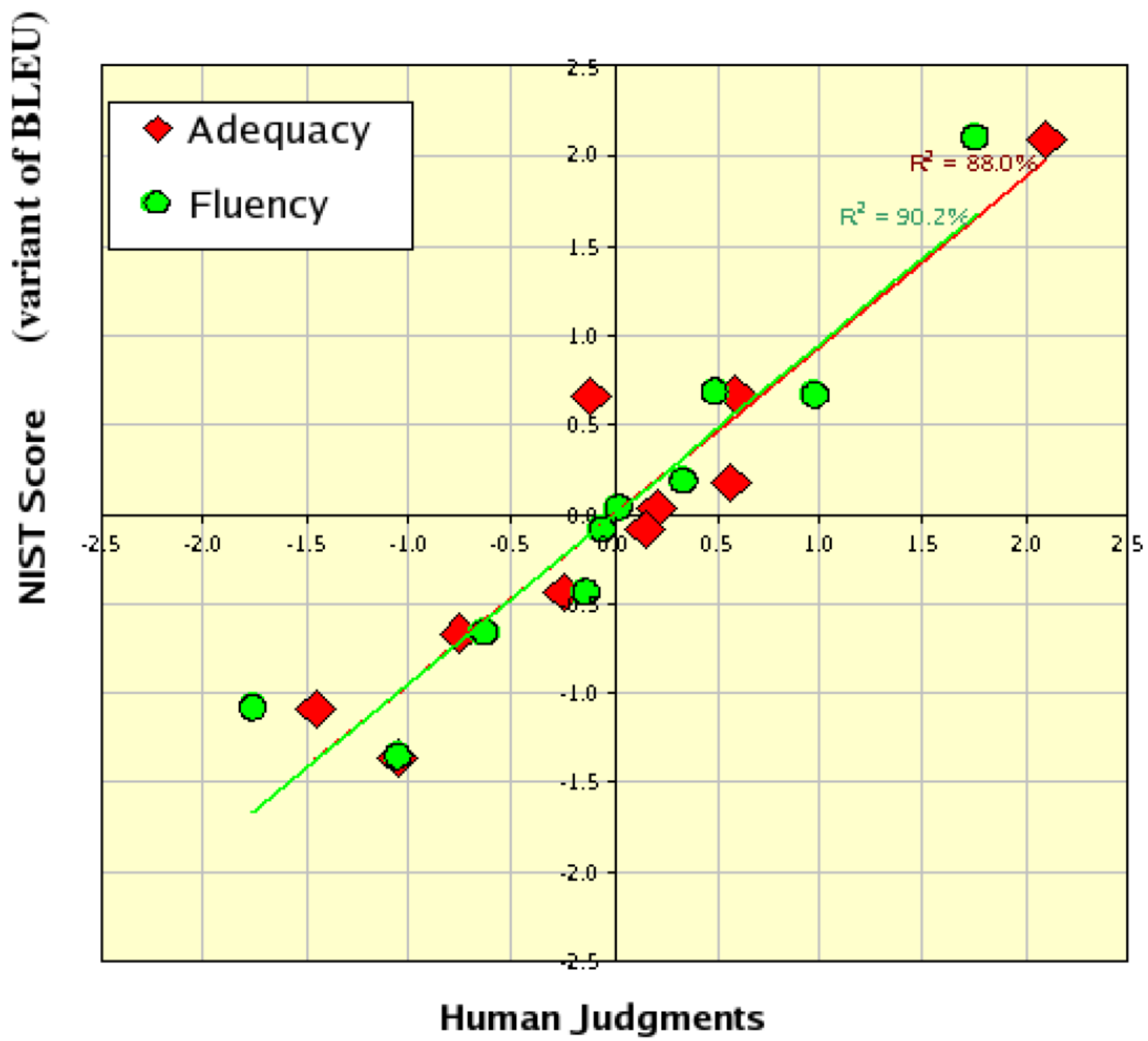
*from Toral & Way, 2018*

# what are some drawbacks of BLEU?

# what are some drawbacks of BLEU?

- all words/n-grams treated as equally relevant
- operates on local level
- scores are meaningless (absolute value not informative)
- human translators also score low on BLEU

# Yet automatic metrics such as BLEU correlate with human judgement



exercise!