recurrent neural language models

CS 585, Fall 2019

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 from Richard Socher

stuff from last time…

- HW1 due next Monday, project proposal due next Friday
- project groups?
- can you post readings earlier?
- can you give us a timeline of all due dates?

language model review

- Goal: compute the probability of a sentence or sequence of words:

$$P(W) = P(w_1,w_2,w_3,w_4,w_5 \ldots w_n)$$

- Related task: probability of an upcoming word:

$$P(w_5|w_1,w_2,w_3,w_4)$$

- A model that computes either of these:

$P(W)$ or $P(w_n|w_1,w_2 \ldots w_{n-1})$ is called a **language model** or **LM**

$$p(w_j \mid \text{students opened their}) = \frac{\text{count(students opened their } w_j)}{\text{count(students opened their)}}$$

what is the order of this n-gram model? (i.e., what is n?)

Problems with n-gram Language Models

Sparsity Problem 1

Problem: What if *"students opened their* $\boldsymbol{w}_j$*"* never occurred in data? Then $\boldsymbol{w}_j$ has probability 0!

$$p(w_j \mid \text{students opened their}) = \frac{\text{count(students opened their } w_j)}{\text{count(students opened their)}}$$

Problems with n-gram Language Models

Sparsity Problem 1

Problem: What if *"students opened their* w_j*"* never occurred in data? Then w_j has probability 0!

(Partial) Solution: Add small δ to count for every $w_j \in V$. This is called *smoothing*.

$$p(w_j \mid \text{students opened their}) = \frac{\text{count}(\text{students opened their } w_j)}{\text{count}(\text{students opened their})}$$

Problems with n-gram Language Models

Sparsity Problem 1

Problem: What if *"students opened their* $\boldsymbol{w}_j$*"* never occurred in data? Then $\boldsymbol{w}_j$ has probability 0!

(Partial) Solution: Add small δ to count for every $\boldsymbol{w}_j \in V$. This is called *smoothing*.

$$P(\boldsymbol{w}_j|\text{students opened their}) = \frac{\text{count}(\text{students opened their } \boldsymbol{w}_j)}{\text{count}(\text{students opened their})}$$

Sparsity Problem 2

Problem: What if *"students opened their"* never occurred in data? Then we can't calculate probability for *any* $\boldsymbol{w}_j$!

Problems with n-gram Language Models

Sparsity Problem 1

Problem: What if *"students opened their* $\boldsymbol{w}_j$*"* never occurred in data? Then $\boldsymbol{w}_j$ has probability 0!

(Partial) Solution: Add small δ to count for every $\boldsymbol{w}_j \in V$. This is called *smoothing*.

$$P(\boldsymbol{w}_j|\text{students opened their}) = \frac{\text{count}(\text{students opened their } \boldsymbol{w}_j)}{\text{count}(\text{students opened their})}$$

Sparsity Problem 2

Problem: What if *"students opened their"* never occurred in data? Then we can't calculate probability for *any* $\boldsymbol{w}_j$!

(Partial) Solution: Just condition on *"opened their"* instead. This is called *backoff*.

Problems with n-gram Language Models

Storage: Need to store count for all possible *n*-grams. So model size is O(exp(*n*)).

$$P(\boldsymbol{w}_j|\text{students opened their}) = \frac{\text{count(students opened their } \boldsymbol{w}_j\text{)}}{\text{count(students opened their)}}$$

Increasing *n* makes model size huge!

How to build a *neural* Language Model?

- Recall the Language Modeling task:
  - Input: sequence of words $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(t)}$
  - Output: prob dist of the next word $P(\boldsymbol{x}^{(t+1)} = \boldsymbol{w}_j \mid \boldsymbol{x}^{(t)}, \ldots, \boldsymbol{x}^{(1)})$

- How about a window-based neural model?

A fixed-window neural Language Model

as the proctor started the clock *the students opened their* _____

discard

fixed window

A fixed-window neural Language Model

output distribution

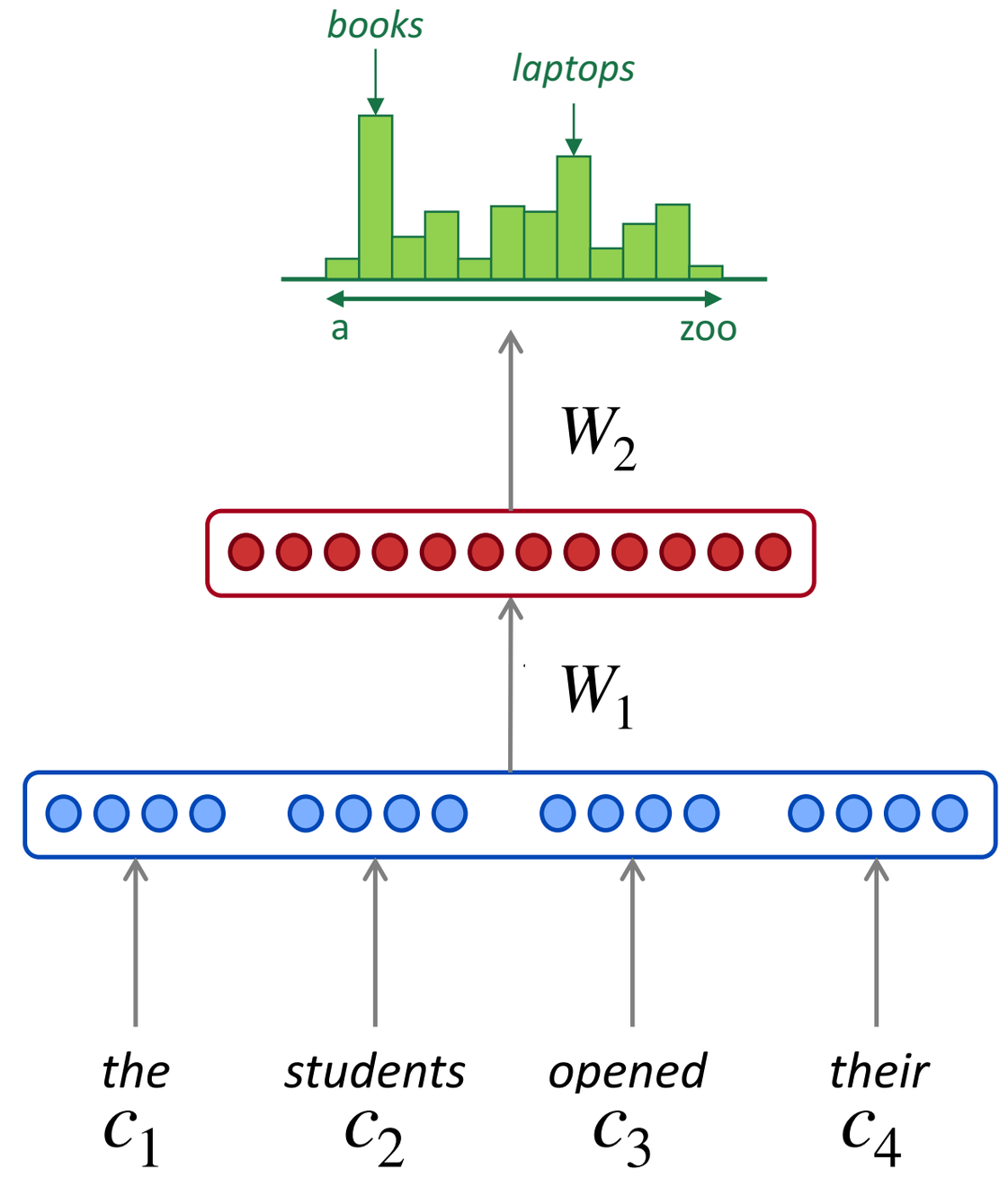
$$\hat{y} = \text{softmax}(W_2 h + b_2)$$

hidden layer

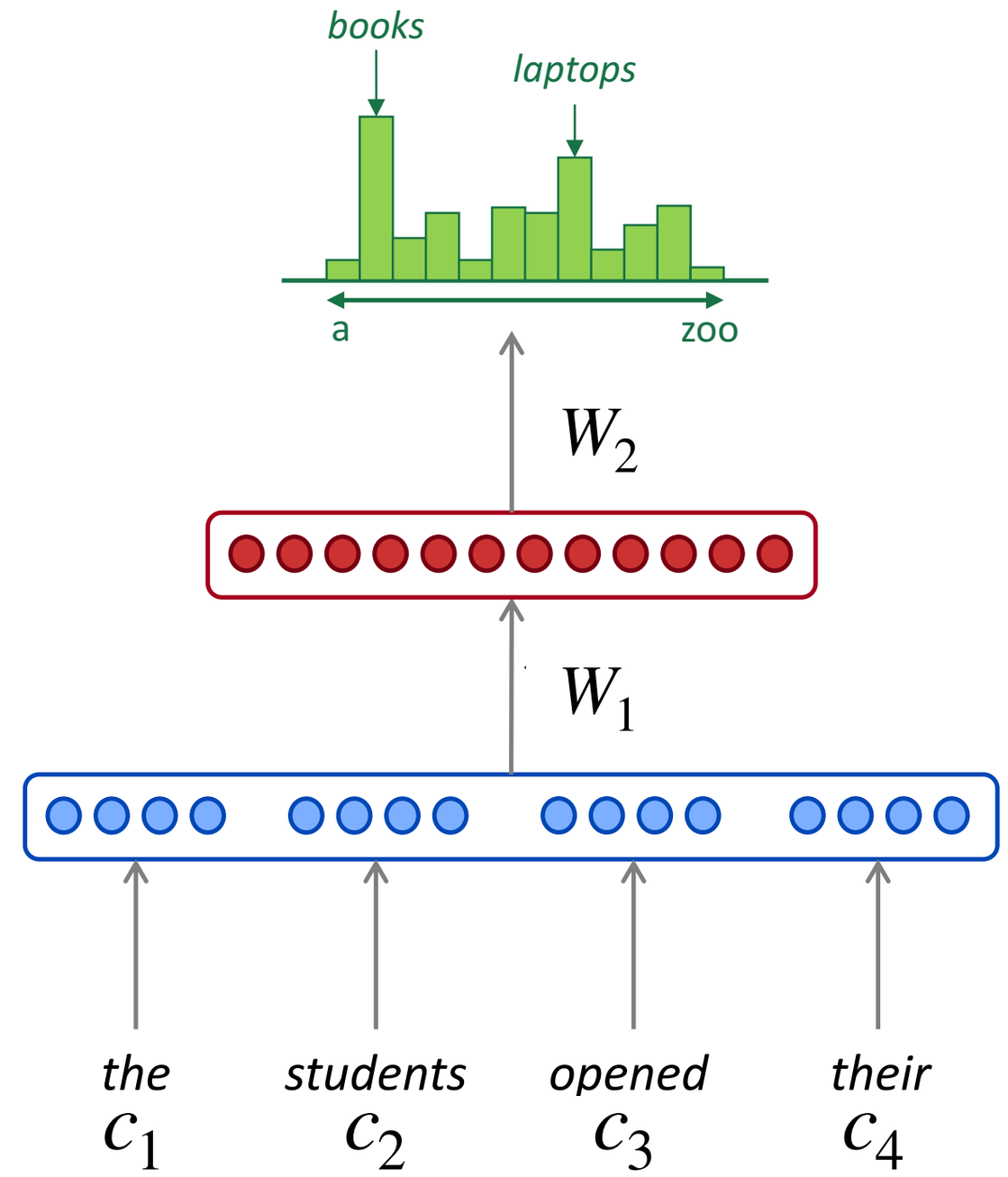
$$h = f(W_1 c + b_1)$$

concatenated word embeddings

$$c = [c_1; c_2; c_3; c_4]$$



how does this compare to a normal n-gram model?



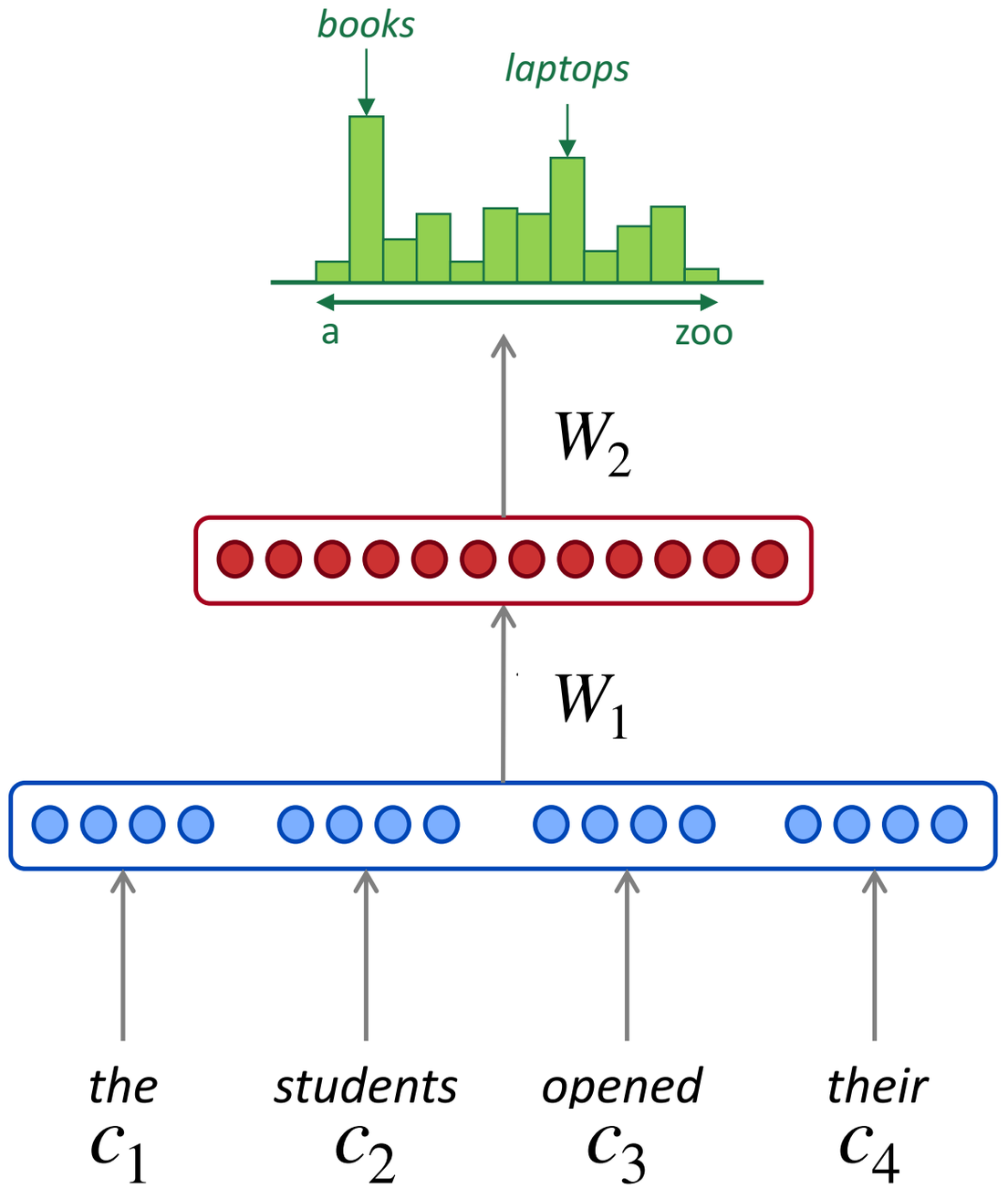
how does this compare to a normal n-gram model?

Improvements over *n*-gram LM:

- No sparsity problem
- Model size is O(*n*) not O(exp(*n*))

Remaining **problems**:

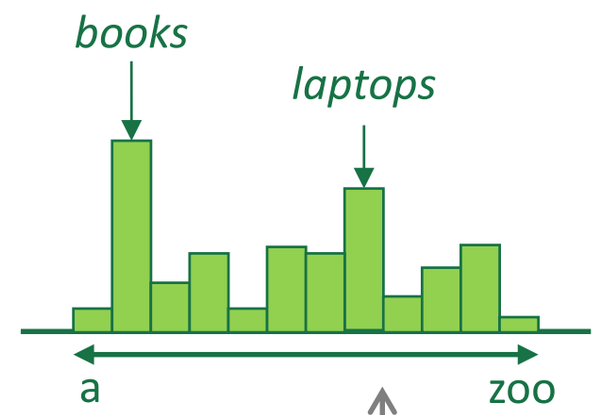
- Fixed window is too small
- Enlarging window enlarges $\boldsymbol{W}$
- Window can never be large enough!
- Each c_i uses different rows of $\boldsymbol{W}$. We don't share weights across the window.



Recurrent Neural Networks!

A RNN Language Model

$$\hat{\boldsymbol{y}}^{(4)} = P(\boldsymbol{x}^{(5)}|\text{the students opened their})$$



output distribution

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

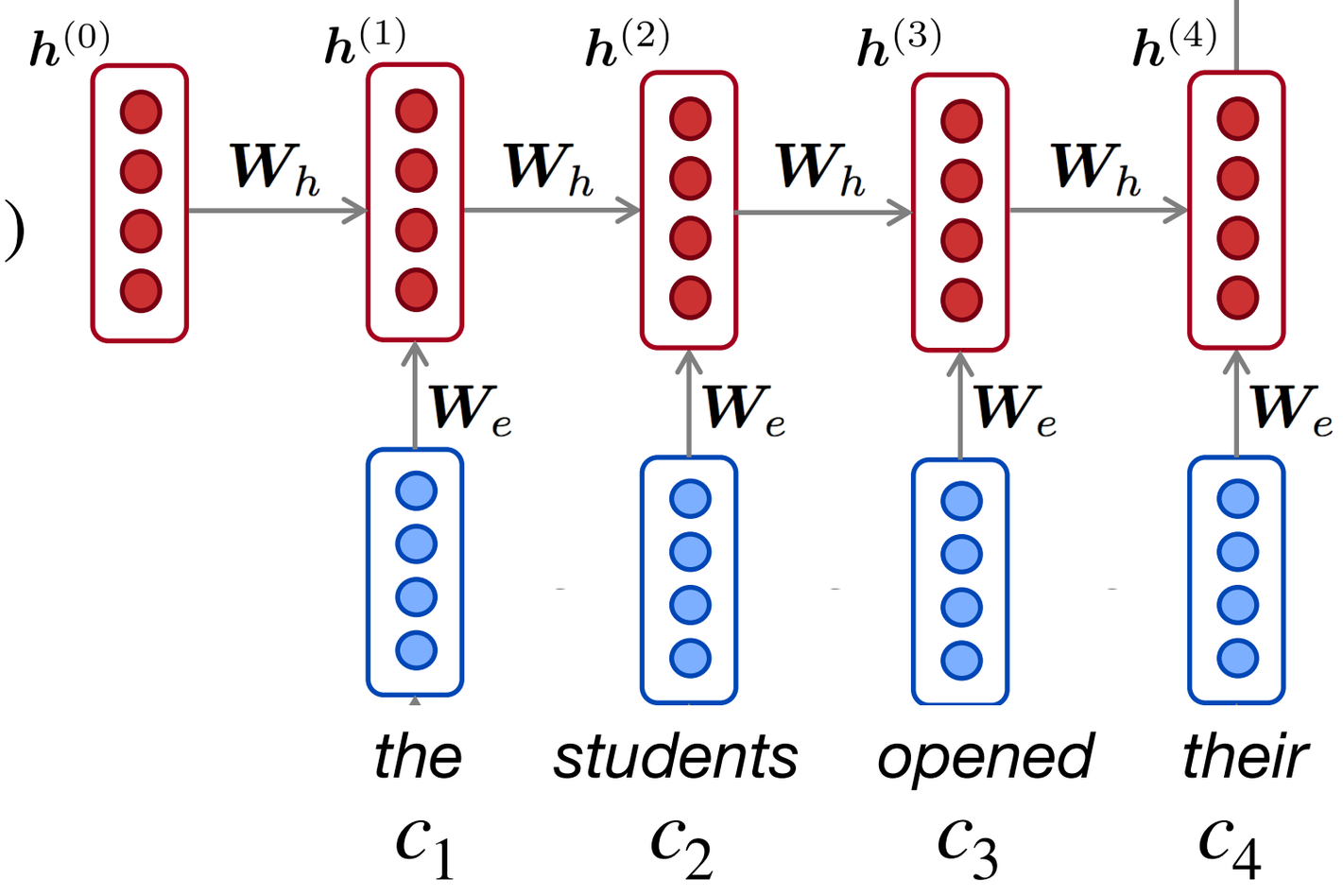
hidden states

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

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

word embeddings

$$c_1, c_2, c_3, c_4$$



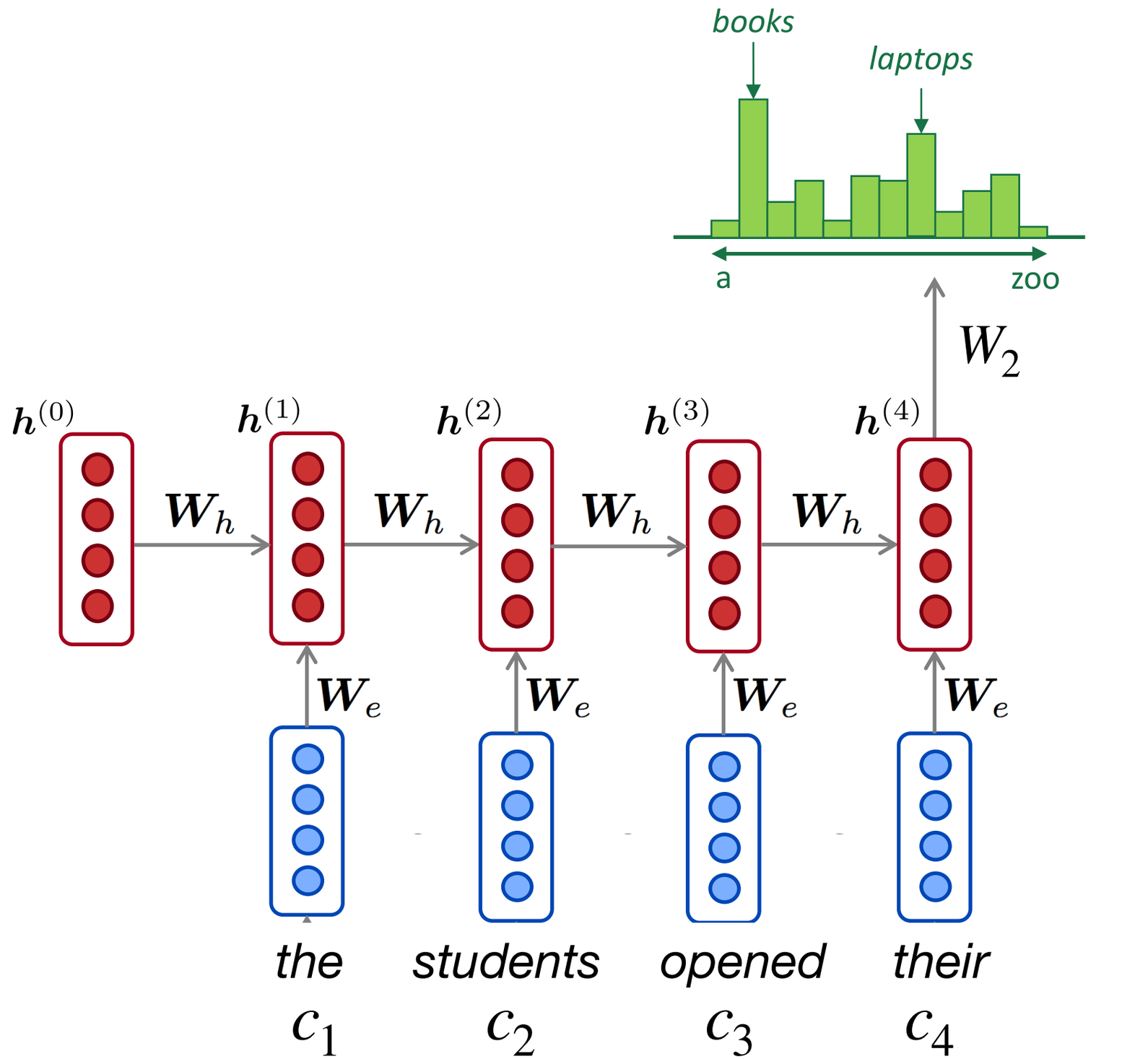
why is this good?

RNN **Advantages**:

- Can process any length input
- Model size doesn't increase for longer input
- Computation for step *t* can (in theory) use information from many steps back
- Weights are shared across timesteps → representations are shared

RNN **Disadvantages**:

- Recurrent computation is slow
- In practice, difficult to access information from many steps back



let’s look at the derivatives!

Training a RNN Language Model

- Get a big corpus of text which is a sequence of words $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(T)}$
- Feed into RNN-LM; compute output distribution $\hat{\boldsymbol{y}}^{(t)}$ for *every step t*.
  - i.e. predict probability dist of *every word*, given words so far

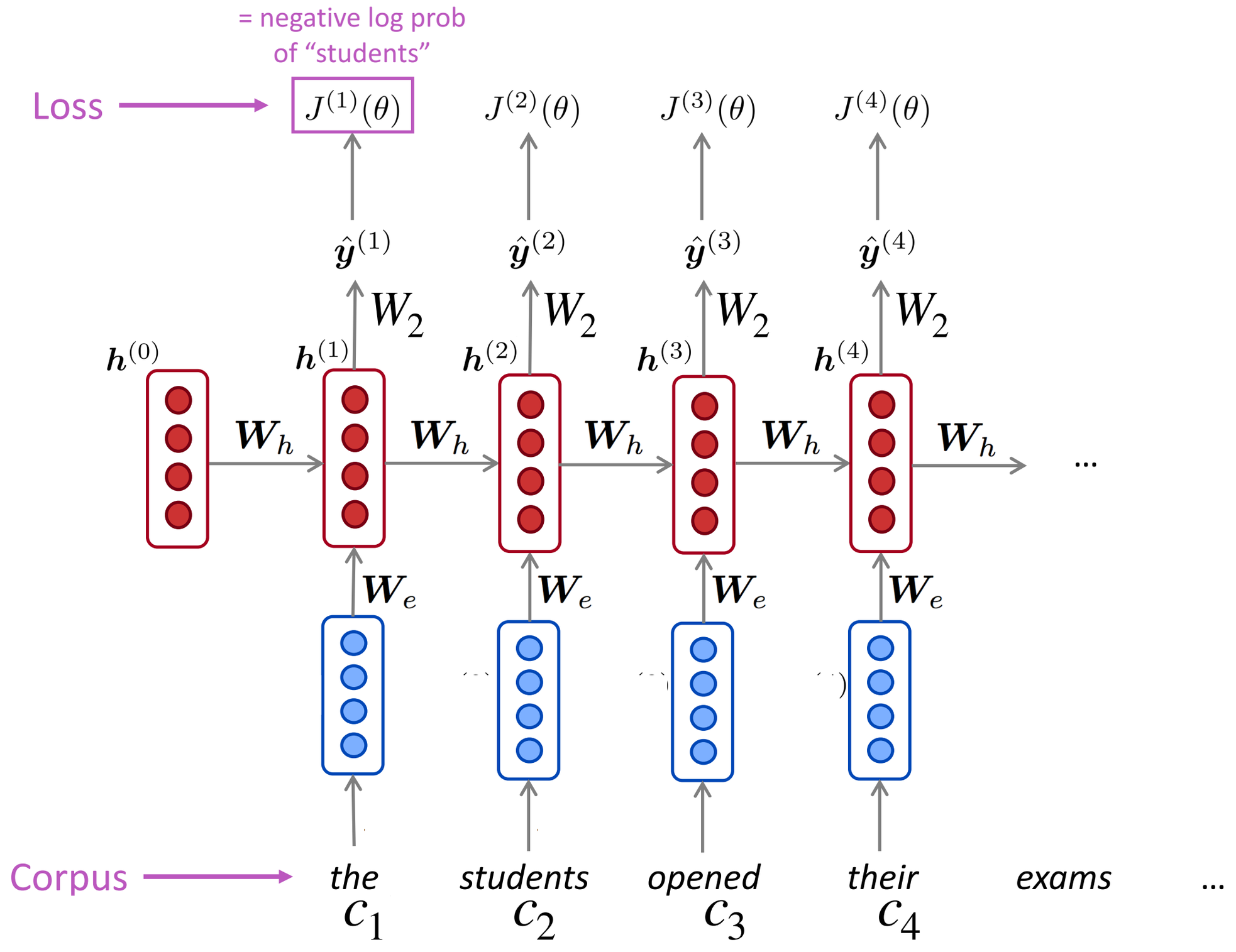
- Loss function on step *t* is usual cross-entropy between our predicted probability distribution $\hat{\boldsymbol{y}}^{(t)}$, and the true next word $\boldsymbol{y}^{(t)} = \boldsymbol{x}^{(t+1)}$:

$$J^{(t)}(\theta) = CE(\boldsymbol{y}^{(t)}, \hat{\boldsymbol{y}}^{(t)}) = -\sum_{j=1}^{|V|} y_j^{(t)} \log \hat{y}_j^{(t)}$$

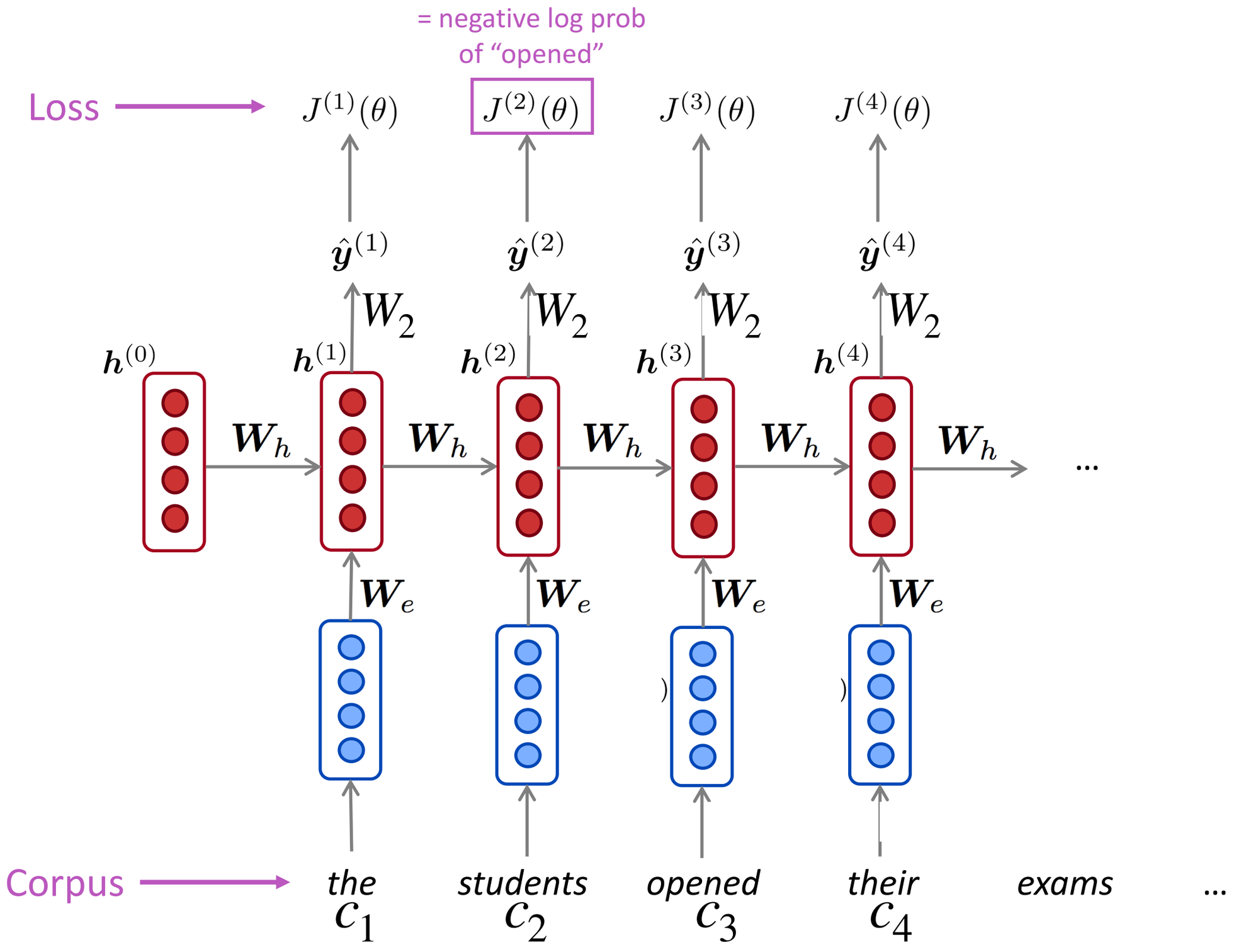
- Average this to get overall loss for entire training set:

$$J(\theta) = \frac{1}{T} \sum_{t=1}^{T} J^{(t)}(\theta)$$

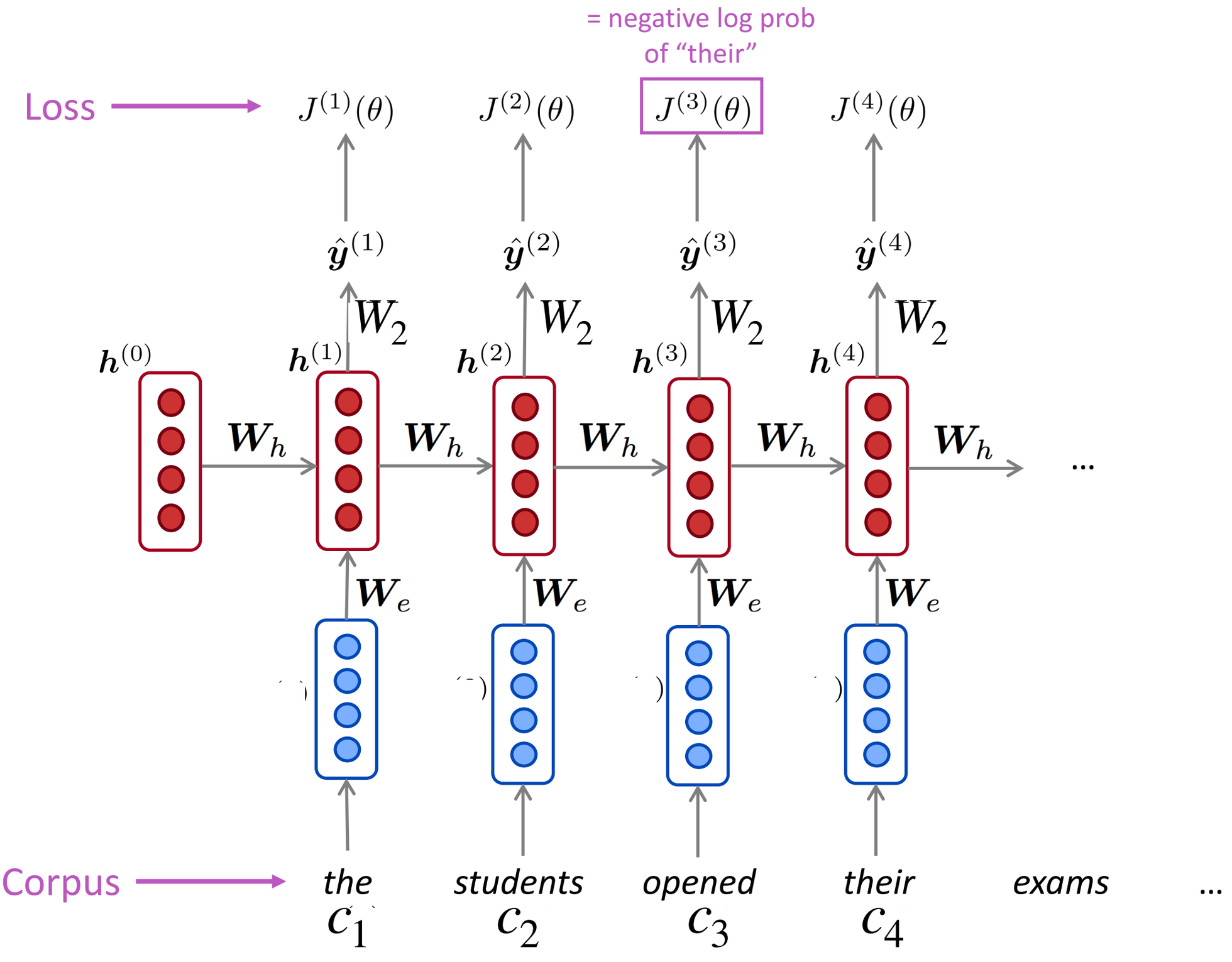
Training a RNN Language Model



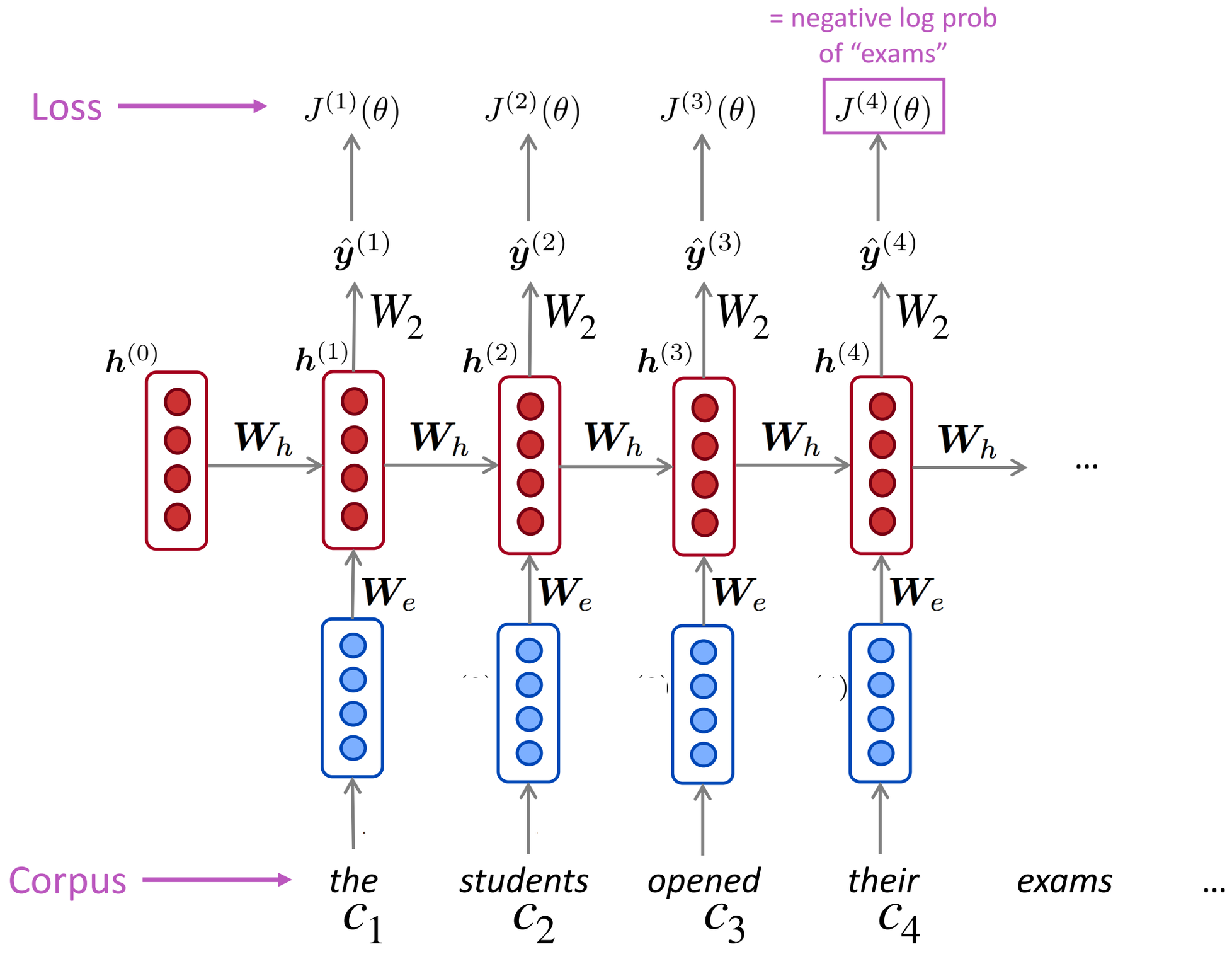
Training a RNN Language Model



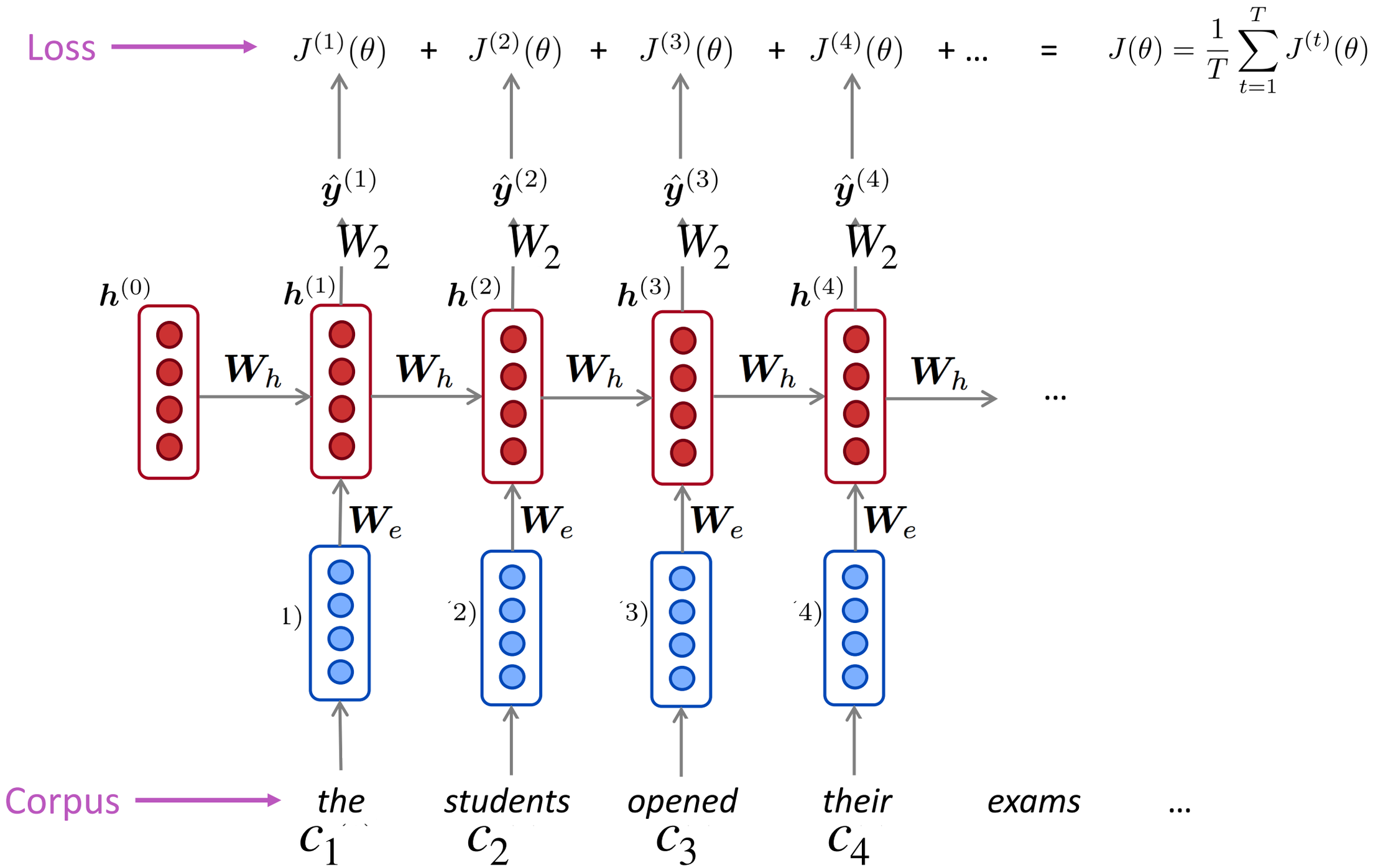
Training a RNN Language Model



Training a RNN Language Model



Training a RNN Language Model



Training a RNN Language Model

- However: Computing loss and gradients across entire corpus is too expensive!
- Recall: Stochastic Gradient Descent allows us to compute loss and gradients for small chunk of data, and update.

- → In practice, consider $x^{(1)}, \ldots, x^{(T)}$ as a sentence

$$J(\theta) = \frac{1}{T} \sum_{t=1}^{T} J^{(t)}(\theta)$$

- Compute loss $J(\theta)$ for a sentence (actually usually a batch of sentences), compute gradients and update weights. Repeat.

RNNs have greatly improved perplexity

n-gram model →

Increasingly complex RNNs ↓

Model	Perplexity
Interpolated Kneser-Ney 5-gram (Chelba et al., 2013)	67.6
RNN-1024 + MaxEnt 9-gram (Chelba et al., 2013)	51.3
RNN-2048 + BlackOut sampling (Ji et al., 2015)	68.3
Sparse Non-negative Matrix factorization (Shazeer et al., 2015)	52.9
LSTM-2048 (Jozefowicz et al., 2016)	43.7
2-layer LSTM-8192 (Jozefowicz et al., 2016)	30
Ours small (LSTM-2048)	43.9
Ours large (2-layer LSTM-2048)	39.8

Perplexity improves (lower is better) ↓

Source: https://research.fb.com/building-an-efficient-neural-language-model-over-a-billion-words/

okay… enough with the unconditional LMs. let’s switch to conditional LMs!

we’ll start with *machine translation*

MT goals

- Motivation: Human translation is expensive
- Rough translation vs. none
- Interactive assistance for human translators
  - e.g. Lilt
    - https://www.youtube.com/watch?v=YZ7G3gQgpfI
    - https://lilt.com/app/projects/details/1887/edit-document/2306
  - [compare to bilingual dictionary]

MT paradigms

- Rule-based *transfer rules*
  - Manually program lexicons/rules
  - SYSTRAN (AltaVista Babelfish; originally from 70s)

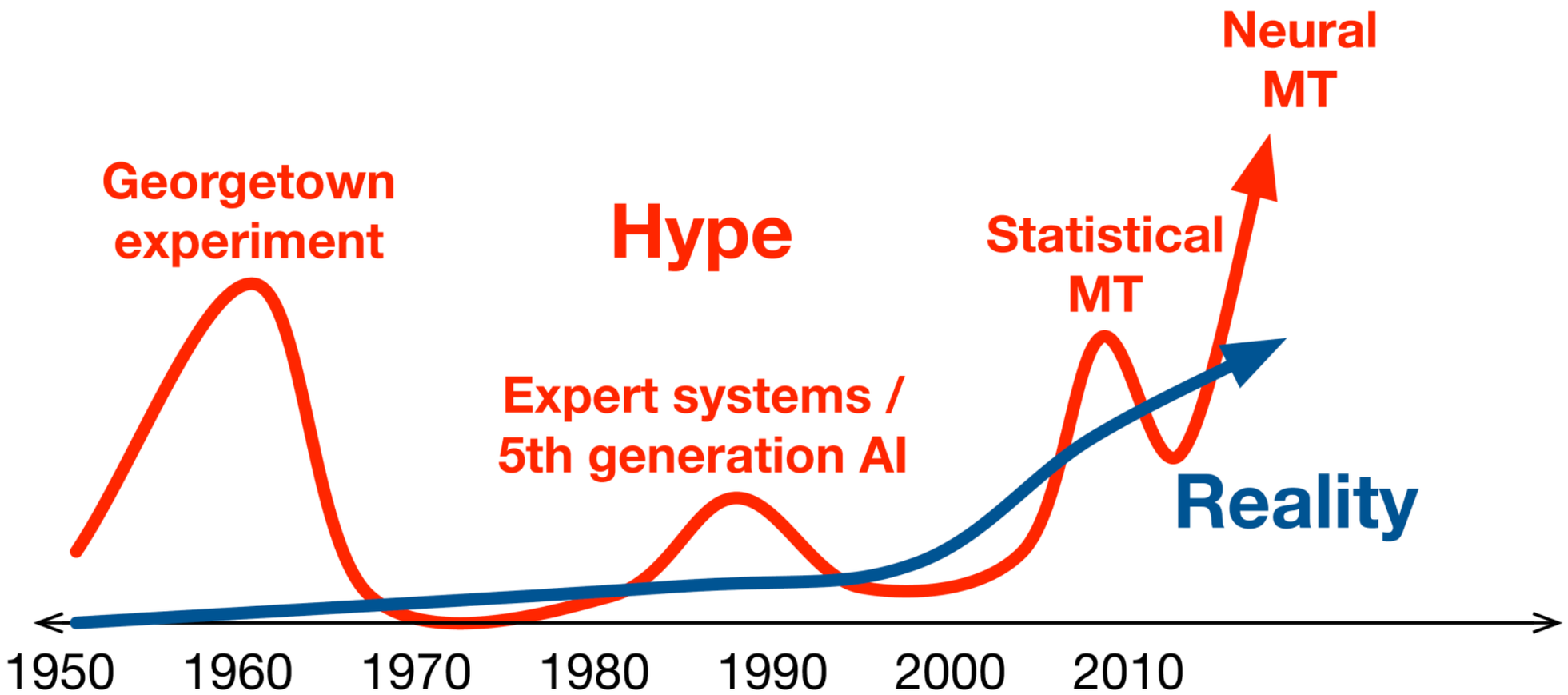
- Statistical MT
  - Word-to-word, phrase-to-phrase probs
  - Learn phrase- or syntax-tree translation rules from data, search for high-scoring translation outputs
  - Key research in the early 90s
  - Google Translate (mid 00s)
  - Open-source: Moses

- Neural MT
  - Research in early 10s; very recently deployed
  - Latent representations of words/phrases

Machine learning for MT

- MT as ML: Translation is something people do naturally. Learn rules from data?
- Parallel data: (source, target) text pairs
  - E.g. 20 million words of European Parliament proceedings
    http://www.statmt.org/europarl/
- Training: learn parameters to predict {source => target}
- Test time: given source sentence, search for high-scoring target (e.g. beam search)

MT History: Hype vs. Reality



How Good is Machine Translation? Chinese > English

记者从环保部了解到，《水十条》要求今年年底前直辖市、省会城市、计划单列市建成区基本解决黑臭水体。截至目前，全国224个地级及以上城市共排查确认黑臭水体2082个，其中34.9%完成整治，28.4%正在整治，22.8%正在开展项目前期。

Reporters learned from the Ministry of Environmental Protection, "Water 10" requirements before the end of this year before the municipality, the provincial capital city, plans to build a separate city to solve the basic black and black water. Up to now, the country's 224 prefecture-level and above cities were identified to confirm the black and white water 2082, of which 34.9% to complete the renovation, 28.4% is remediation, 22.8% is carrying out the project early.

How Good is Machine Translation? French > English

A l'orée de ce débat télévisé inédit dans l'histoire de la Ve République, on attendait une forme de «Tous sur Macron» mais c'est la candidate du Front national qui s'est retrouvée au cœur des premières attaques de ses quatre adversaires d'un soir, favorisées par le premier thème abordé, les questions de société et donc de sécurité, d'immigration et de laïcité.

At the beginning of this televised debate, which was unheard of in the history of the Fifth Republic, a "Tous sur Macron" was expected, but it was the candidate of the National Front who found itself at the heart of the first attacks of its four Opponents of one evening, favored by the first theme tackled, the issues of society and thus security, immigration and secularism.

What is MT good (enough) for?

- **Assimilation:** reader initiates translation, wants to know content
  - User is tolerant of inferior quality
  - Focus of majority of research

- **Communication:** participants in conversation don't speak same language
  - Users can ask questions when something is unclear
  - Chat room translations, hand-held devices
  - Often combined with speech recognition

- **Dissemination:** publisher wants to make content available in other languages
  - High quality required
  - Almost exclusively done by human translators

today: neural MT

- we'll use French (*f*) to English (*e*) as a running example
- **goal**: given French sentence *f* with tokens f_1, f_2, … f_n produce English translation *e* with tokens e_1, e_2, … e_m

is *n* always equal to *m*?

today: neural MT

- we'll use French (*f*) to English (*e*) as a running example
- **goal**: given French sentence *f* with tokens f_1, f_2, … f_n produce English translation *e* with tokens e_1, e_2, … e_m

is *n* always equal to *m*?

- **real goal**: compute $\arg\max_e p(e \mid f)$

today: neural MT

- let's use an NN to directly model $p(e|f)$

$$p(e|f) = p(e_1, e_2, \ldots, e_m|f)$$
$$= p(e_1|f) \cdot p(e_2|e_1, f) \cdot p(e_3|e_2, e_1, f) \cdot \ldots$$
$$= \prod_{i=1}^{m} p(e_i|e_1, \ldots, e_{i-1}, f)$$

how does this formulation relate to the language models we discussed previously?

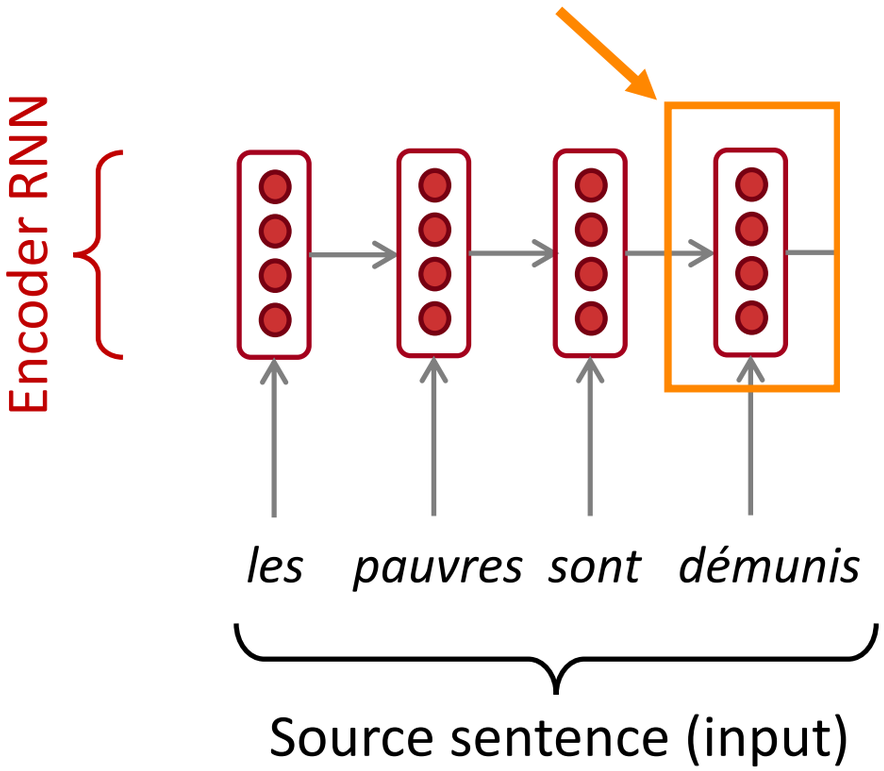
seq2seq models

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

Neural Machine Translation (NMT)

The sequence-to-sequence model

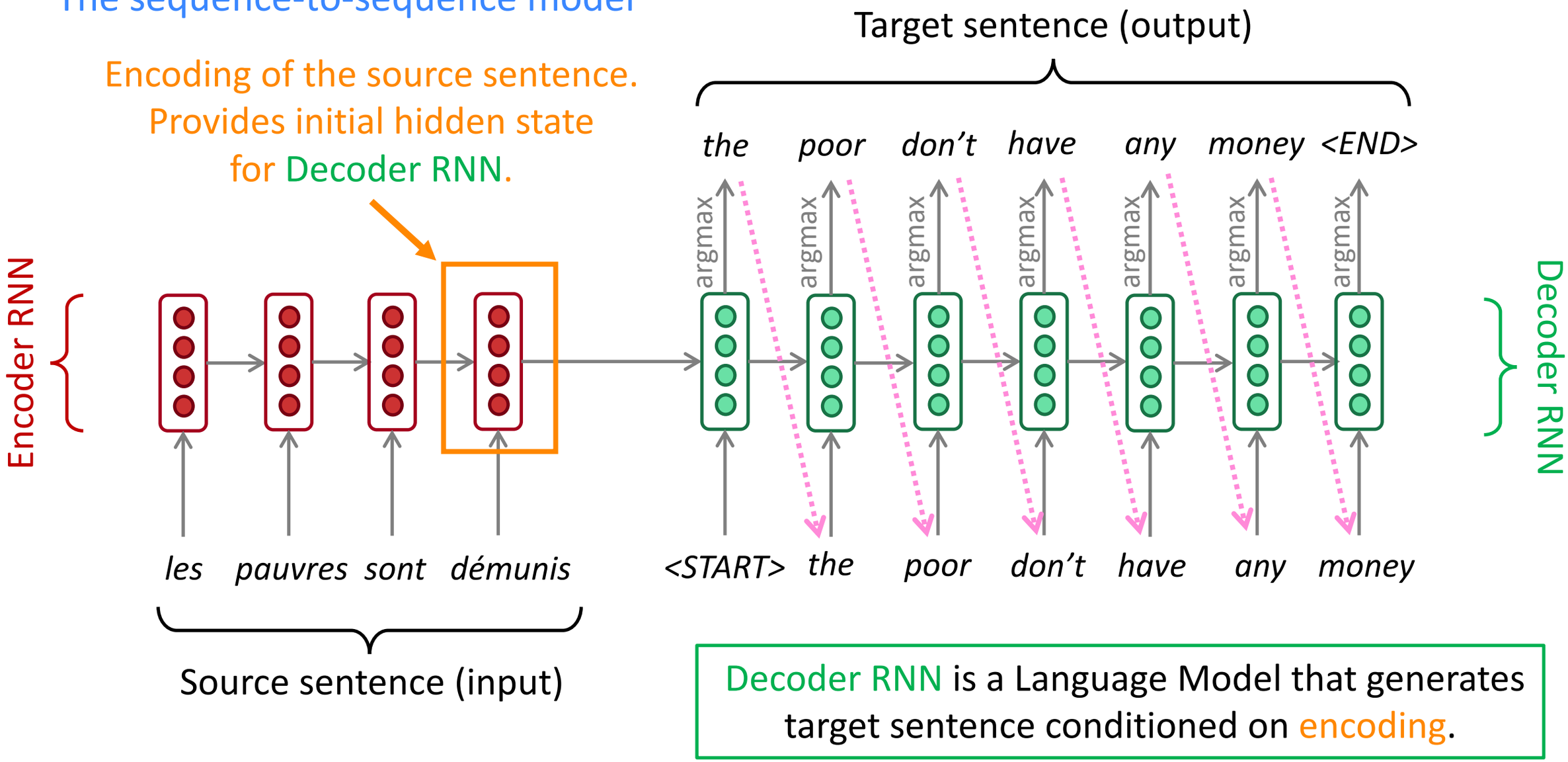
Encoding of the source sentence.
Provides initial hidden state
for Decoder RNN.



Encoder RNN produces an encoding of the source sentence.

Neural Machine Translation (NMT)

The sequence-to-sequence model



Encoder RNN produces an encoding of the source sentence.

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

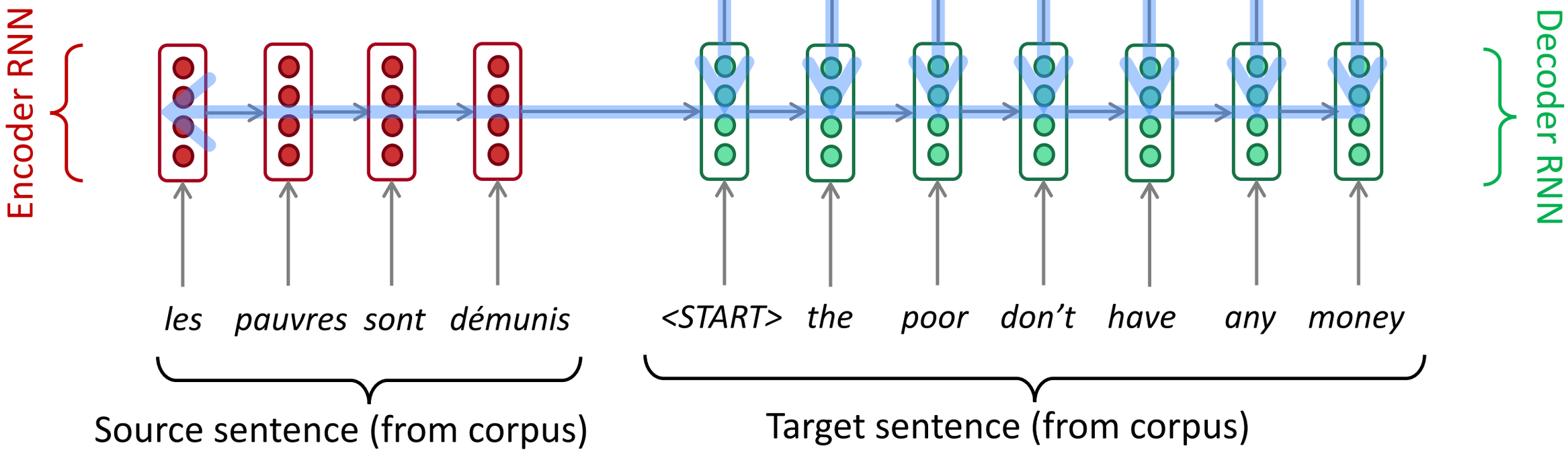
Training a Neural Machine Translation system

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

= negative log prob of "the"

= negative log prob of "have"

= negative log prob of <END>



what are the parameters of this model?

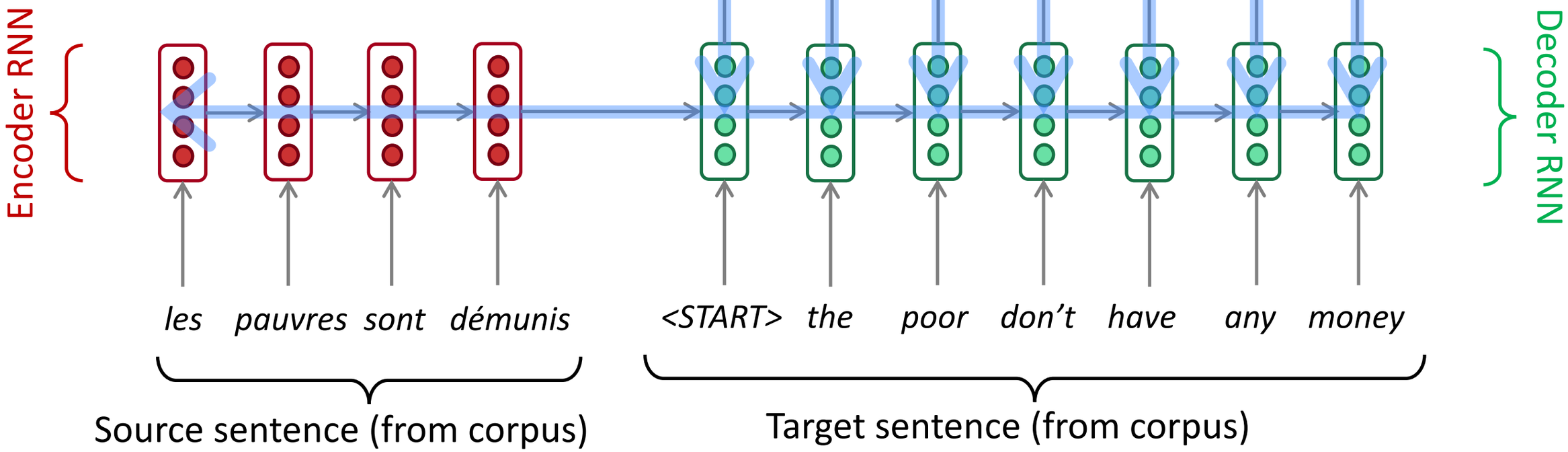
Training a Neural Machine Translation system

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

= negative log prob of "the"

= negative log prob of "have"

= negative log prob of <END>



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

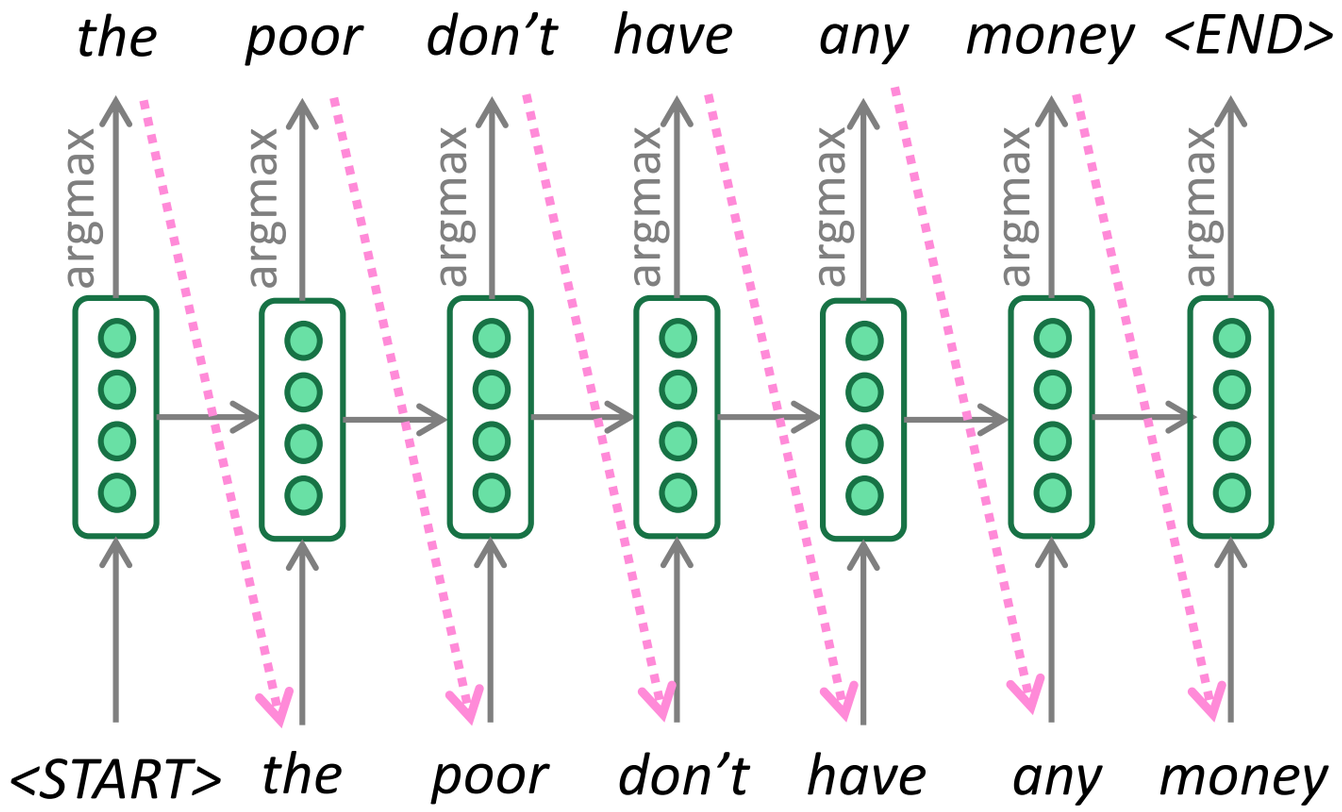
- given that we trained a seq2seq model, how do we find the most probable English sentence?
- more concretely, how do we find

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

- 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?
- easiest option: **greedy decoding**



issues?

Beam search

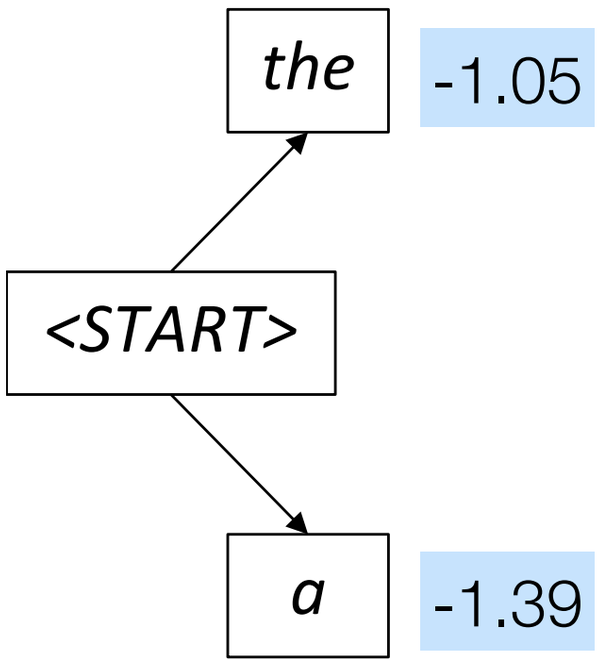
- in greedy decoding, we cannot go back and revise previous decisions!
  - *les pauvres sont démunis (the poor don't have any money)*
  - → *the* ____
  - → *the poor* ____
  - → *the poor are* ____

- 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

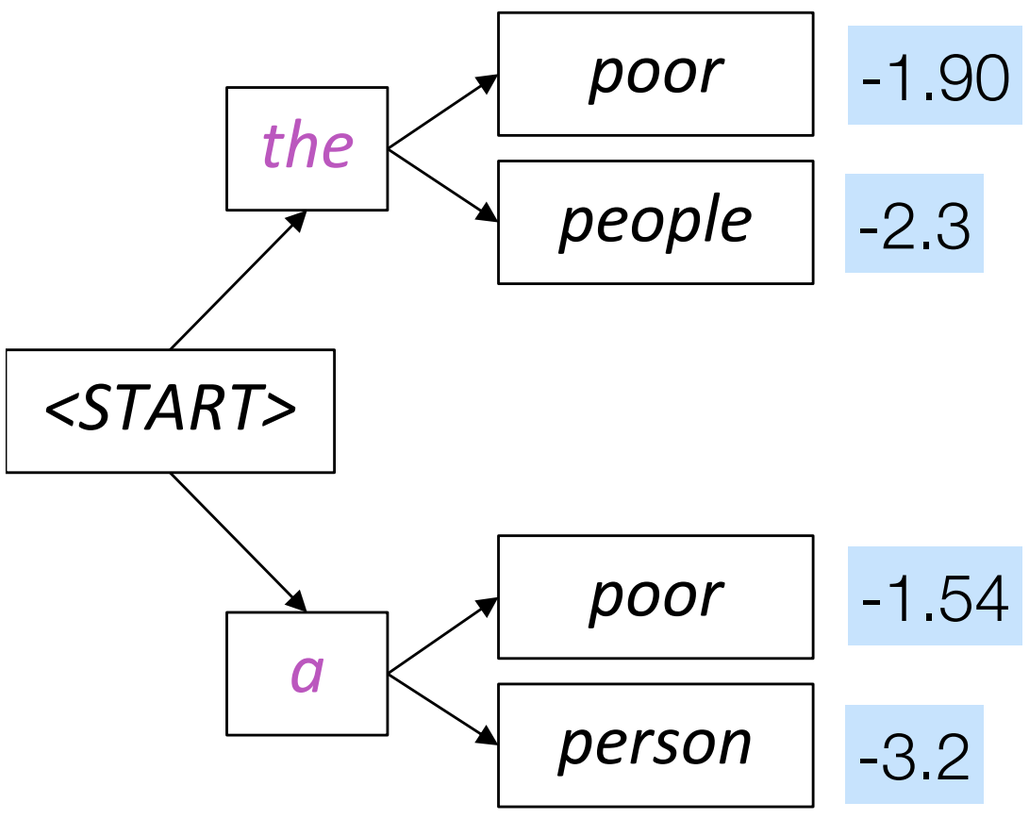
Beam search decoding: example

Beam size = 2



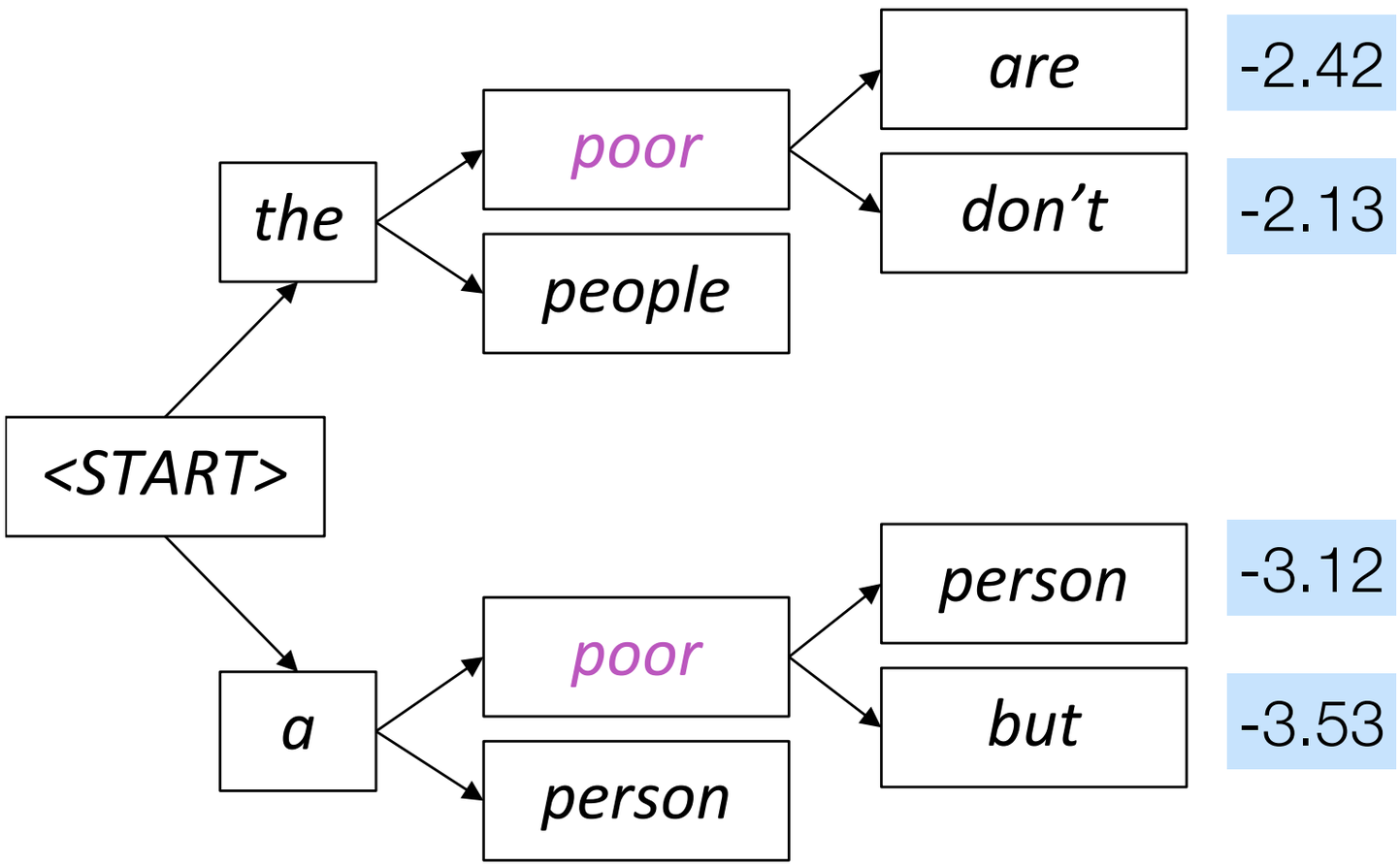
Beam search decoding: example

Beam size = 2



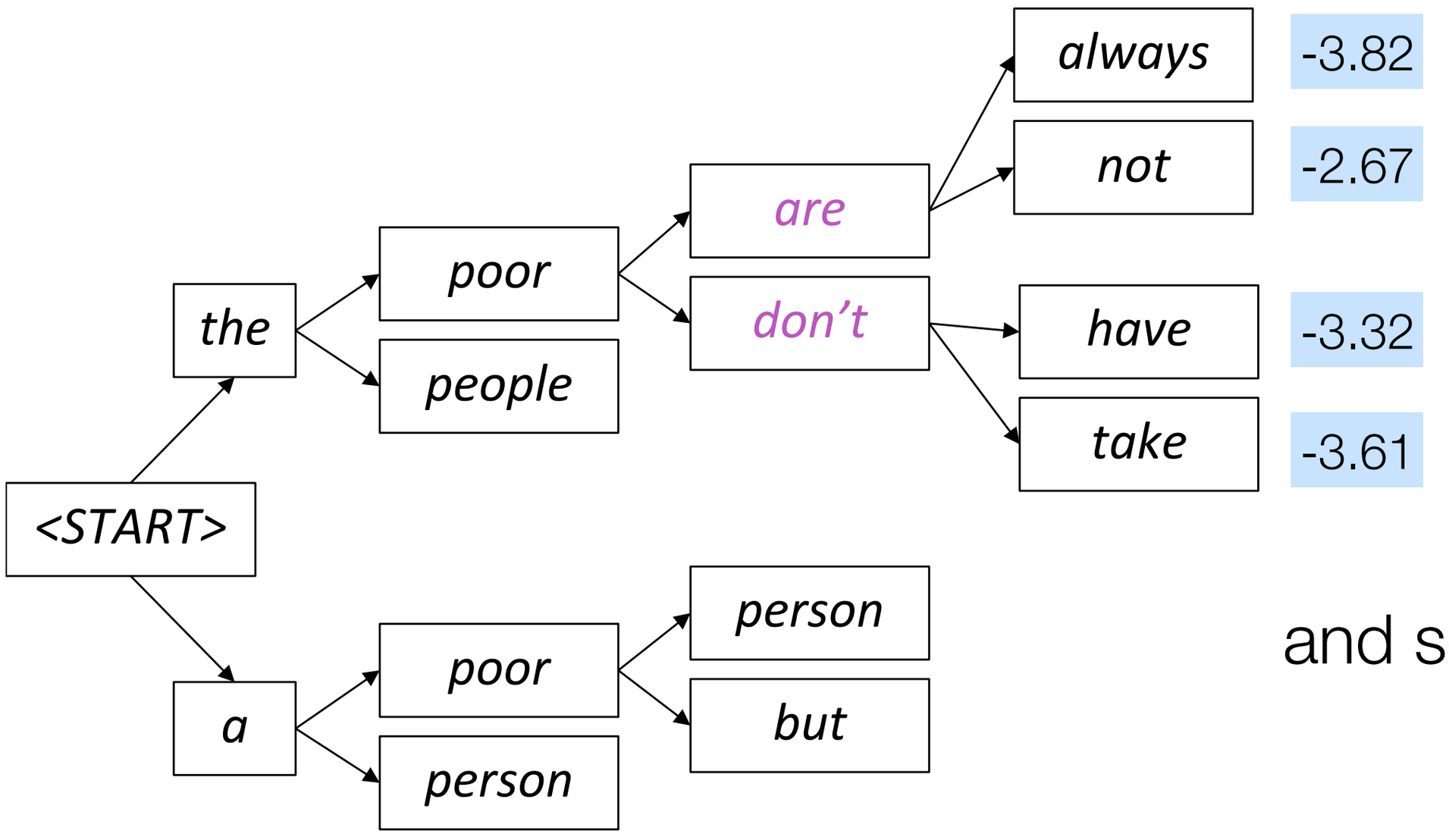
Beam search decoding: example

Beam size = 2



Beam search decoding: example

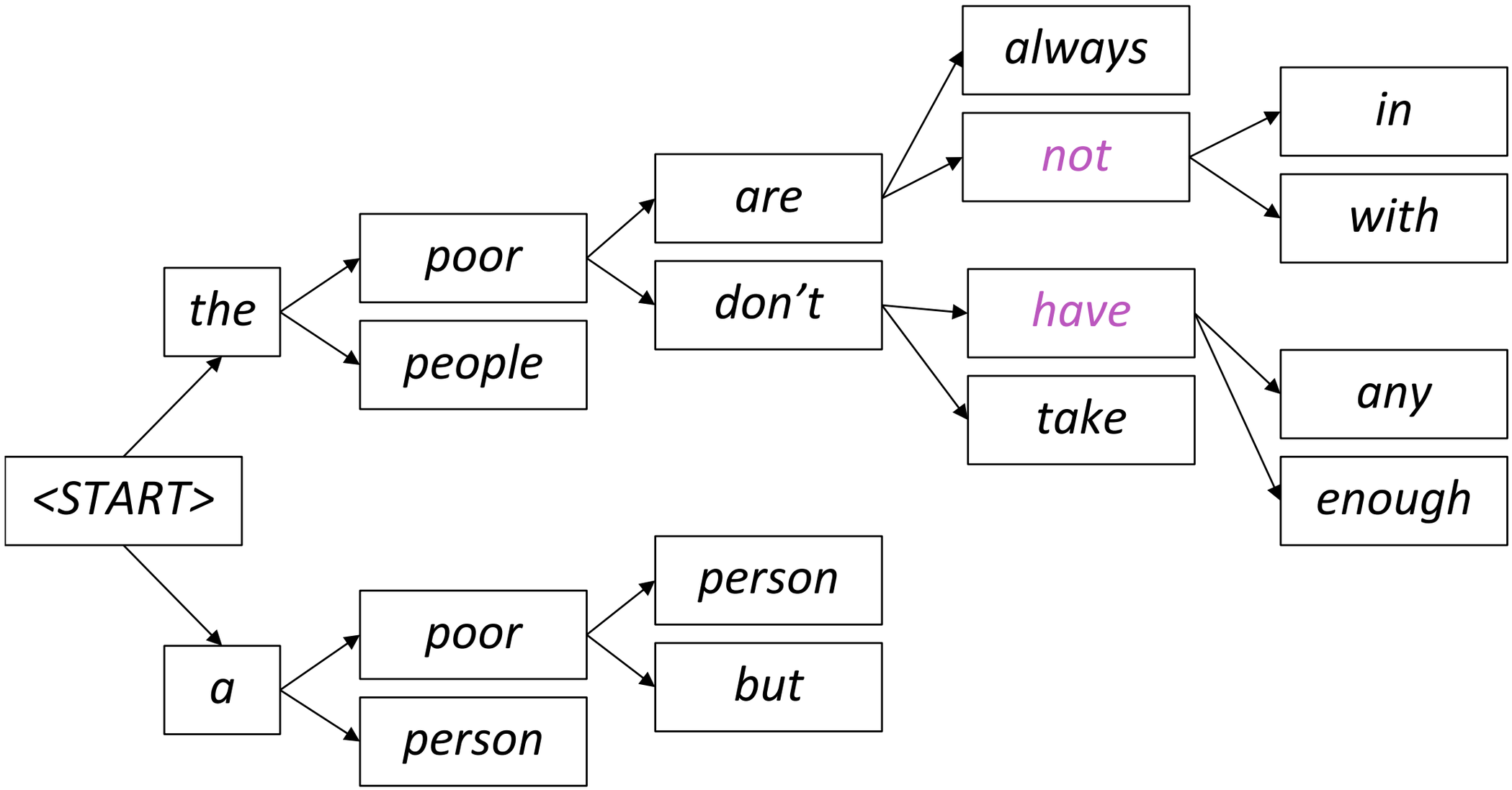
Beam size = 2



and so on…

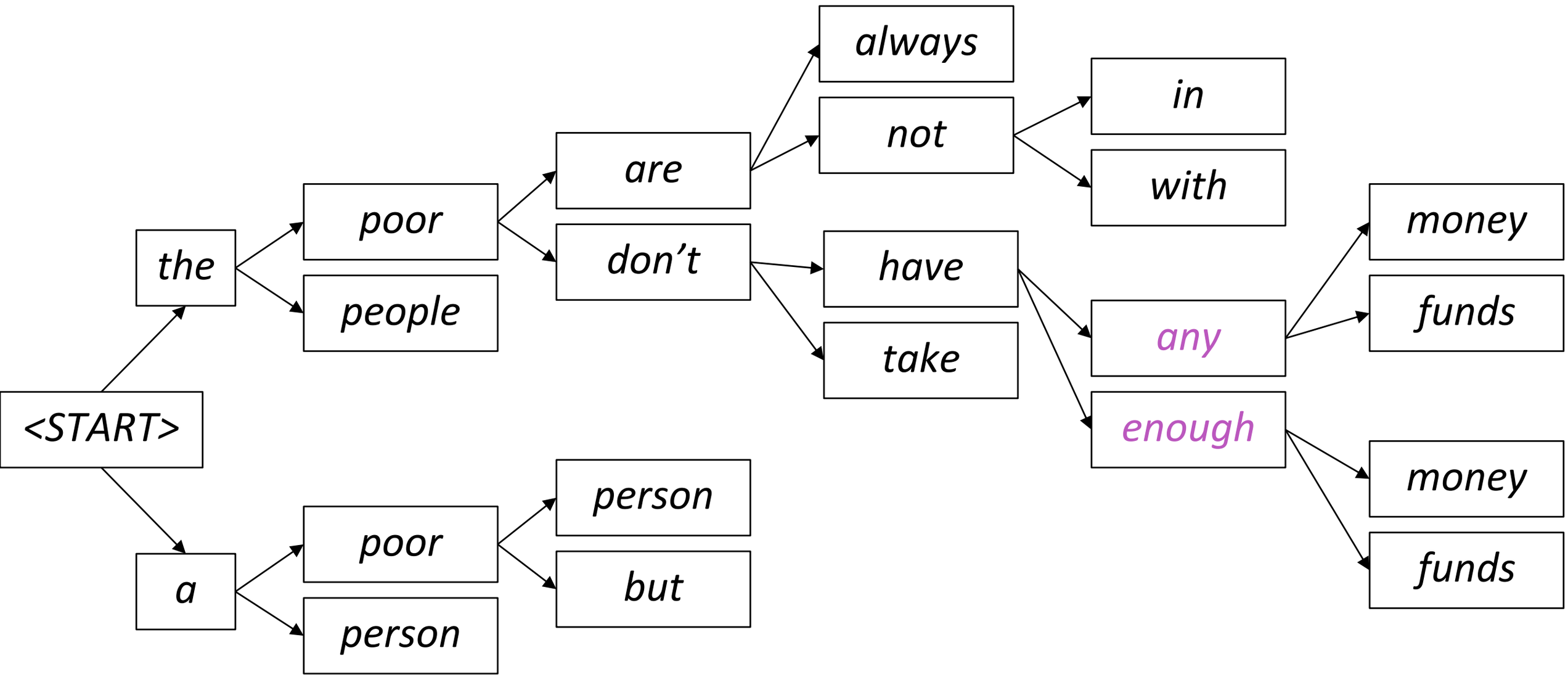
Beam search decoding: example

Beam size = 2



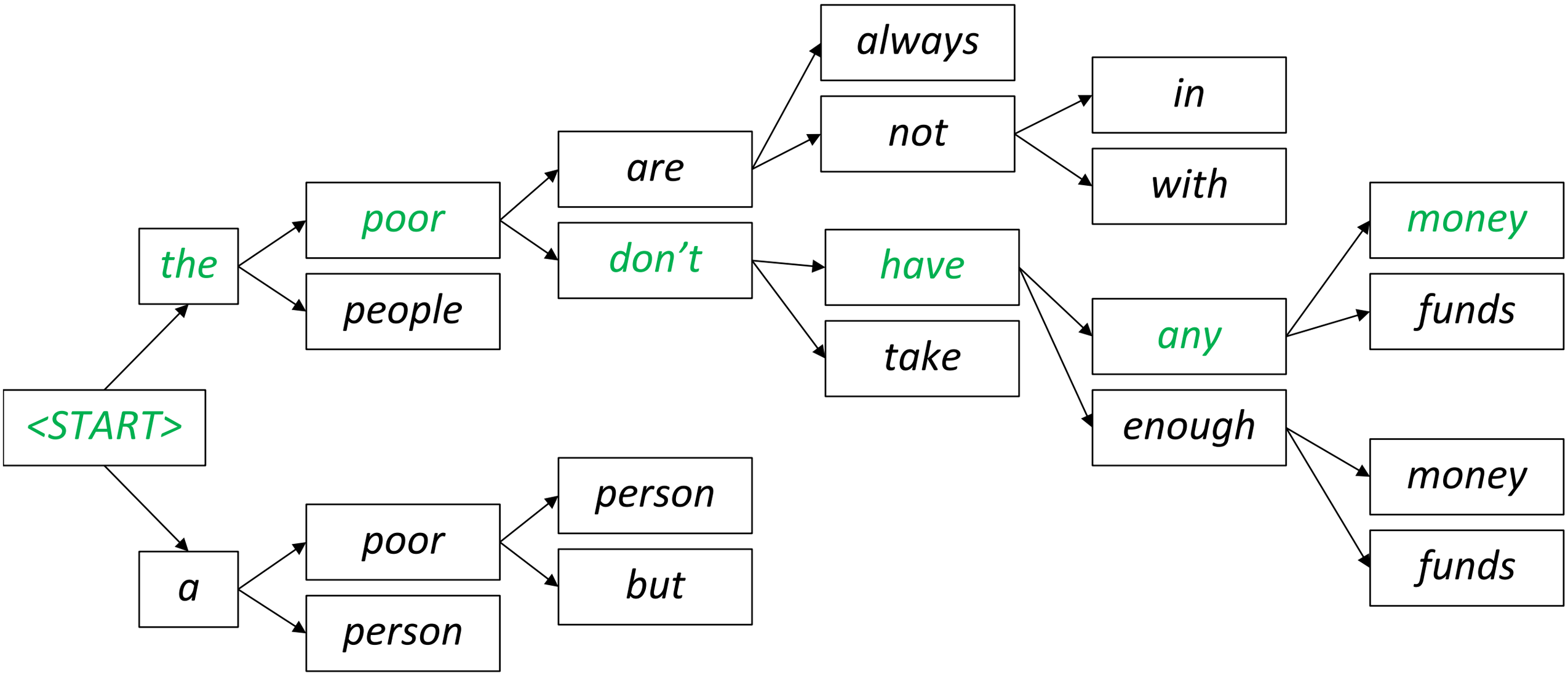
Beam search decoding: example

Beam size = 2



Beam search decoding: example

Beam size = 2



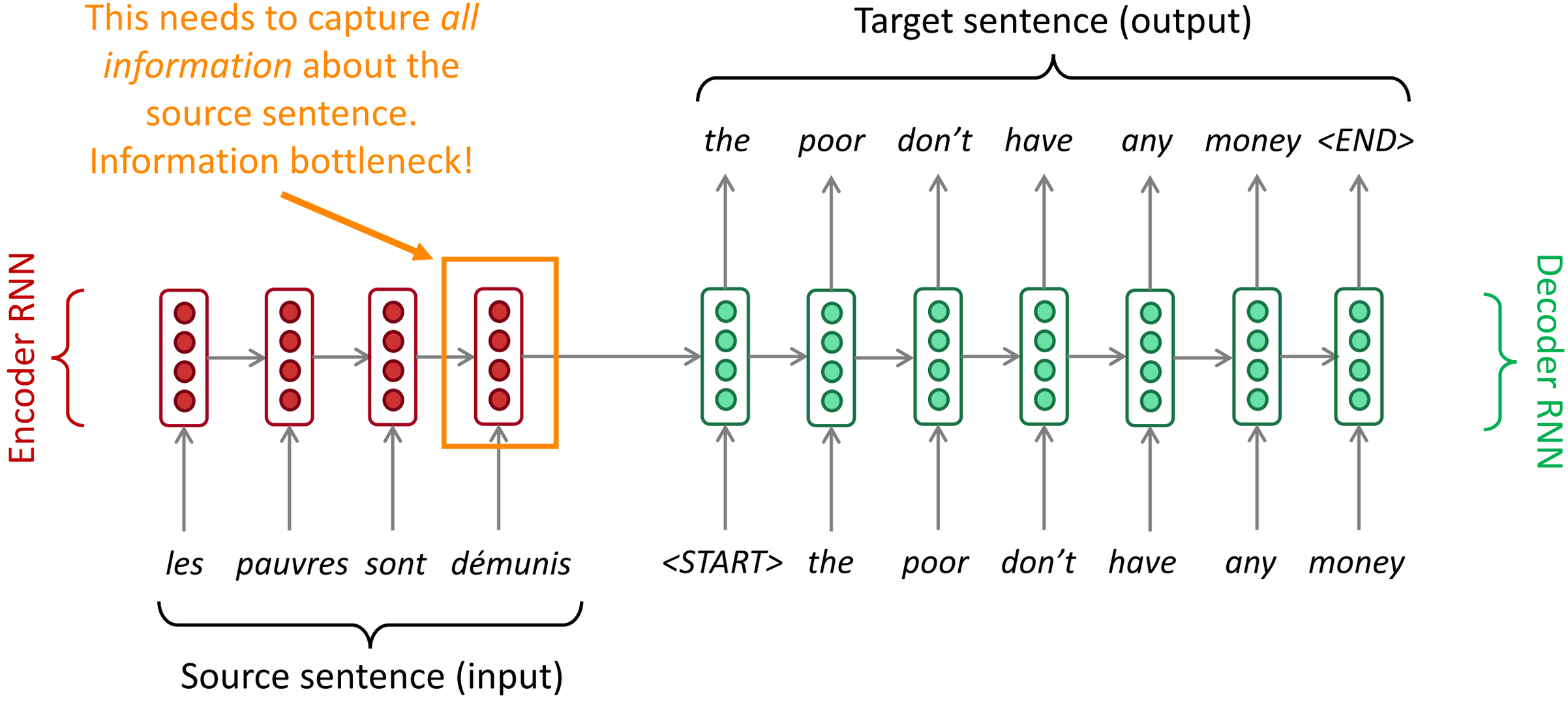
does beam search always produce the *best* translation (i.e., does it always find the argmax?)

what are the termination conditions for beam search?

next class preview: attention!

Sequence-to-sequence: the bottleneck problem

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



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

Adequacy and Fluency

- Human judgment
  - Given: machine translation output
  - Given: input and/or reference translation
  - Task: assess quality of MT output

- Metrics
  - **Adequacy:** does the output convey the meaning of the input sentence? Is part of the message lost, added, or distorted?
  - **Fluency:** is the output fluent? Involves both grammatical correctness and idiomatic word choices.

Fluency and Adequacy: Scales

Adequacy	
5	all meaning
4	most meaning
3	much meaning
2	little meaning
1	none

Fluency	
5	flawless English
4	good English
3	non-native English
2	disfluent English
1	incomprehensible

Judge Sentence

You have already judged 14 of 3064 sentences, taking 86.4 seconds per sentence.

Source: les deux pays constituent plutôt un laboratoire nécessaire au fonctionnement interne de l ' ue .

Reference: rather , the two countries form a laboratory needed for the internal working of the eu .

Translation	Adequacy	Fluency
both countries are rather a necessary laboratory the internal operation of the eu .	1 2 3 4 5	1 2 3 4 5
both countries are a necessary laboratory at internal functioning of the eu .	1 2 3 4 5	1 2 3 4 5
the two countries are rather a laboratory necessary for the internal workings of the eu .	1 2 3 4 5	1 2 3 4 5
the two countries are rather a laboratory for the internal workings of the eu .	1 2 3 4 5	1 2 3 4 5
the two countries are rather a necessary laboratory internal workings of the eu .	1 2 3 4 5	1 2 3 4 5
Annotator: Philipp Koehn **Task:** WMT06 French-English	Annotate	
Instructions	5= All Meaning 4= Most Meaning 3= Much Meaning 2= Little Meaning 1= None	5= Flawless English 4= Good English 3= Non-native English 2= Disfluent English 1= Incomprehensible

Let's try:
rate fluency & adequacy on 1-5 scale

- Source:
  N'y aurait-il pas comme une vague hypocrisie de votre part ?
- Reference:
  Is there not an element of hypocrisy on your part?
- System1:
  Would it not as a wave of hypocrisy on your part?
- System2:
  Is there would be no hypocrisy like a wave of your hand?
- System3:
  Is there not as a wave of hypocrisy from you?

what are some issues with human 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

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

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)

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

Typically computed over the entire corpus, not single sentences

In the MT final project, we will use BLEU to evaluate models

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: Israeli officials (2-GRAM MATCH) responsibility of (2-GRAM MATCH) airport (1-GRAM) safety

REFERENCES:
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

BLEU examples

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

REFERENCE: Israeli officials are responsible for airport security

SYSTEM B: airport security Israeli officials are responsible
(airport security: 2-GRAM MATCH; Israeli officials are responsible: 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%

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

why does BLEU not account for recall?

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%

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