Sequence to Sequence Learning: CNNs, Training and Uncertainty

Michael Auli

with Sergey Edunov, Myle Ott, Jonas Gehring, Angela Fan, Denis Yarats, Yann Dauphin, David Grangier, Marc'Aurelio Ranzato

http://github.com/facebookresearch/fairseq-py

Facebook AI Research (FAIR)

Overview

- Sequence to Sequence Learning & NLP
- Architecture: Sequence to Sequence Learning with CNNs.
- Exposure bias/Loss Mismatch: Training at the Sequence Level.
- Analyzing Uncertainty: model fitting and effects on search.

Sequence to Sequence Learning & Natural Language Processing

Sequence to Sequence Learning

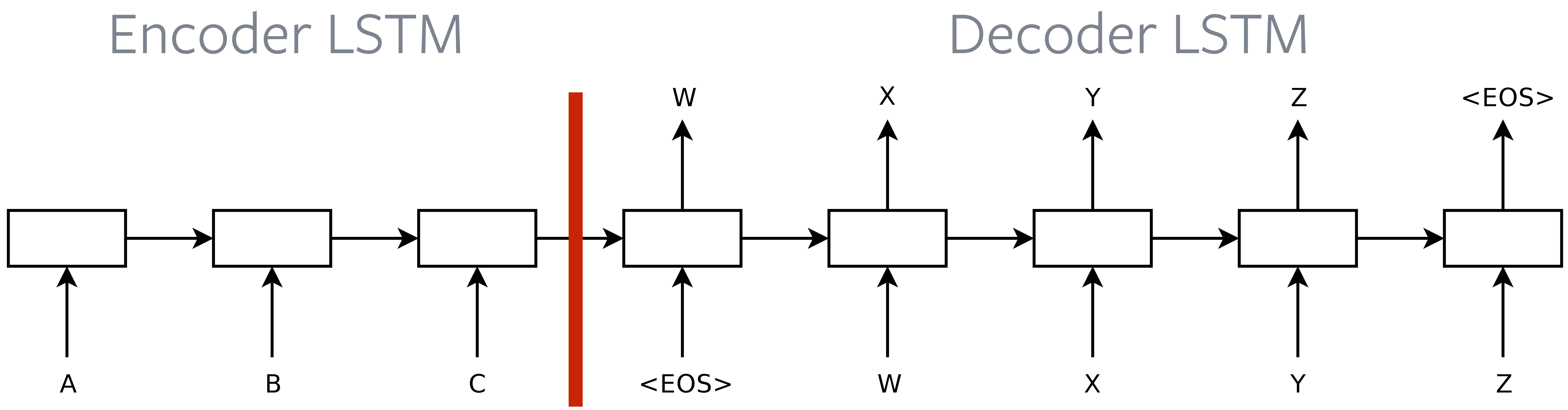


Figure 1: Our model reads an input sentence “ABC” and produces “WXYZ” as the output sentence. The model stops making predictions after outputting the end-of-sentence token. Note that the LSTM reads the input sentence in reverse, because doing so introduces many short term dependencies in the data that make the optimization problem much easier.

- **Encode** source sequence, and **decode** target sequence with **RNNs** (Sutksever et al., 2014)
- **Attention:** choose relevant encoder states (Bahdanau et al., 2014)

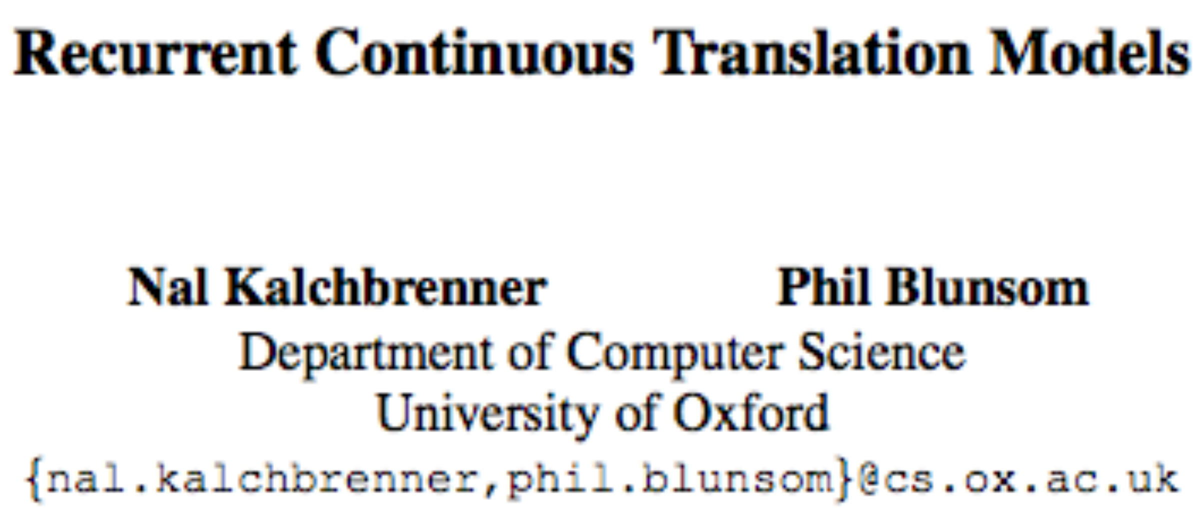
Figure from: Sutskever et al., 2014, ”Sequence to Sequence Learning with Neural Networks”

Sequence to Sequence Learning

- Applications: translation, summarization, parsing, dialogue, ...

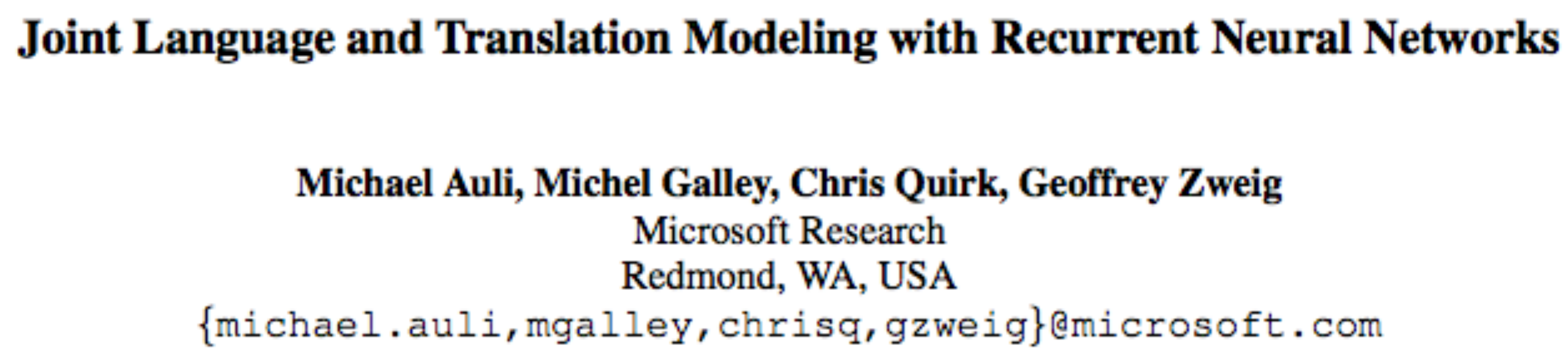
- ″... basis for 25% of papers at ACL.″,
  Mirella Lapata at ACL′17 keynote

Sequence to Sequence Learning



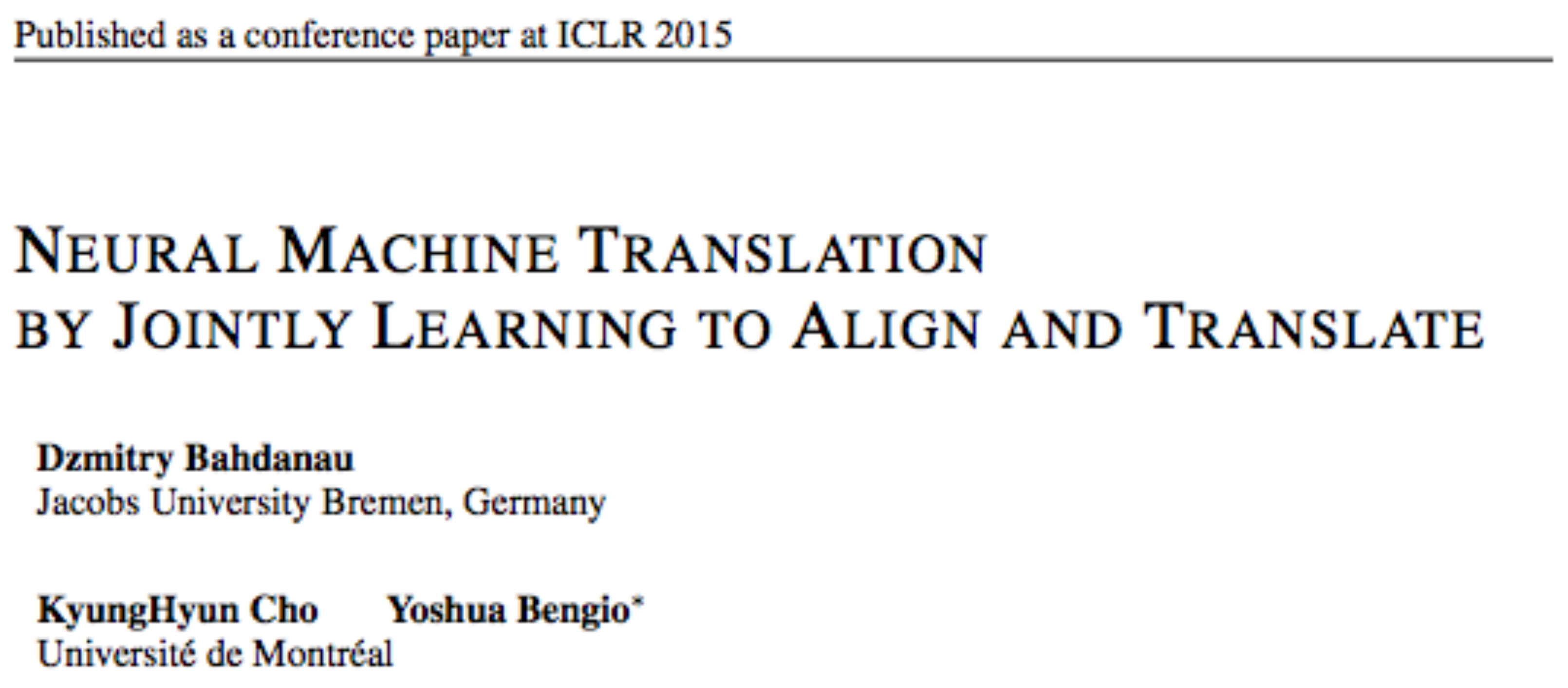
Recurrent Continuous Translation Models

Nal Kalchbrenner **Phil Blunsom**
Department of Computer Science
University of Oxford
{nal.kalchbrenner,phil.blunsom}@cs.ox.ac.uk



Joint Language and Translation Modeling with Recurrent Neural Networks

Michael Auli, Michel Galley, Chris Quirk, Geoffrey Zweig
Microsoft Research
Redmond, WA, USA
{michael.auli,mgalley,chrisq,gzweig}@microsoft.com

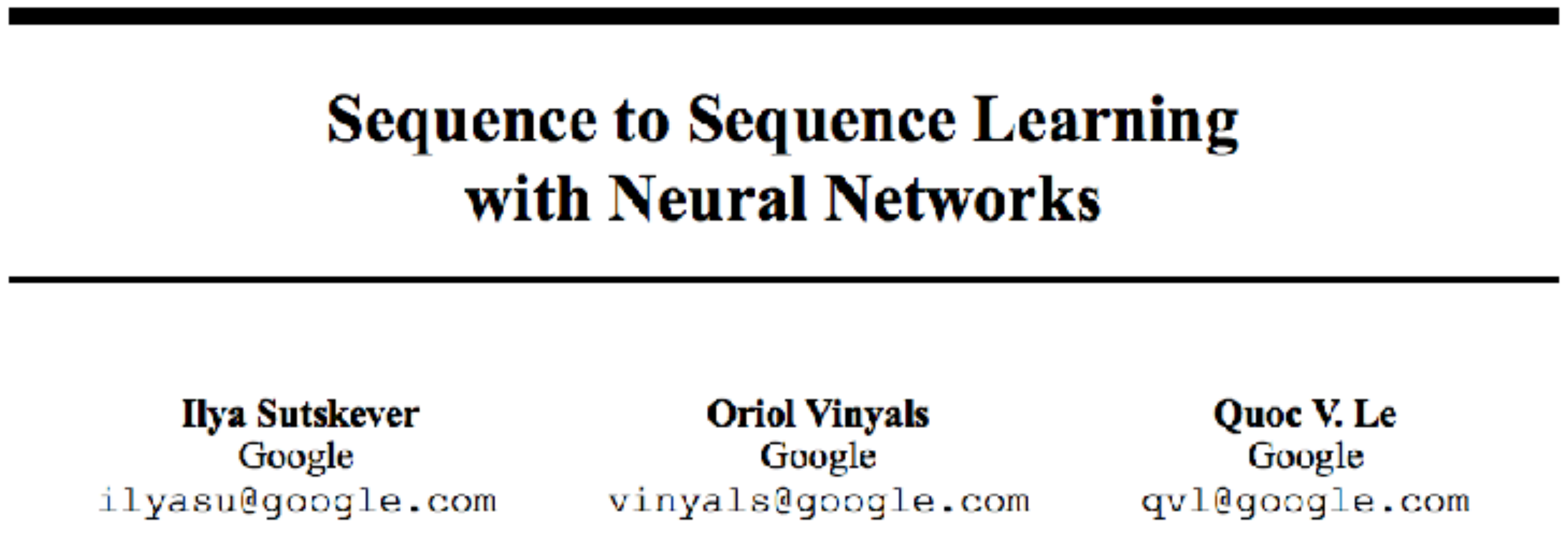


Published as a conference paper at ICLR 2015

NEURAL MACHINE TRANSLATION
BY JOINTLY LEARNING TO ALIGN AND TRANSLATE

Dzmitry Bahdanau
Jacobs University Bremen, Germany

KyungHyun Cho **Yoshua Bengio***
Université de Montréal



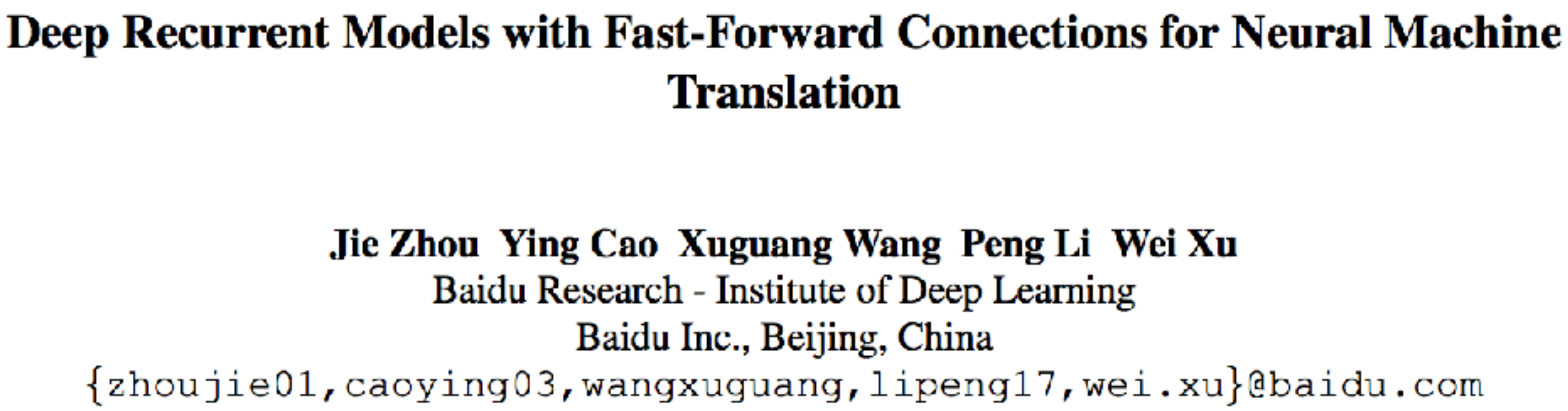
Sequence to Sequence Learning
with Neural Networks

Ilya Sutskever
Google
ilyasu@google.com

Oriol Vinyals
Google
vinyals@google.com

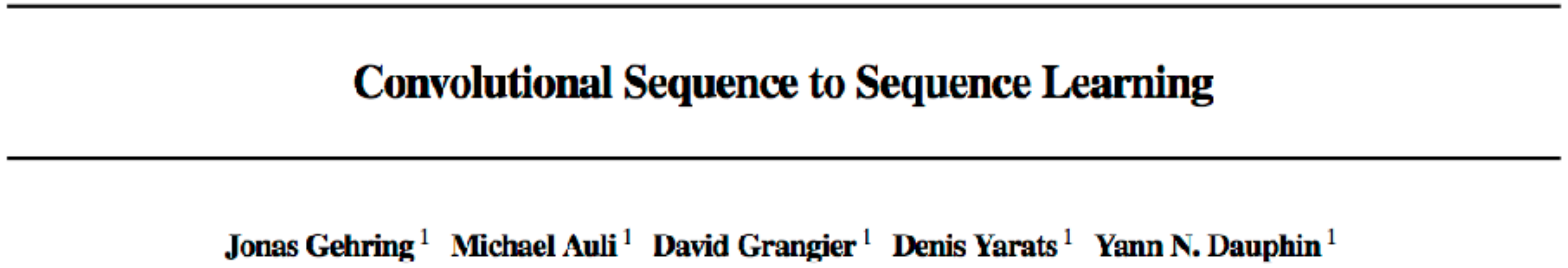
Quoc V. Le
Google
qvl@google.com

Sequence to Sequence Learning



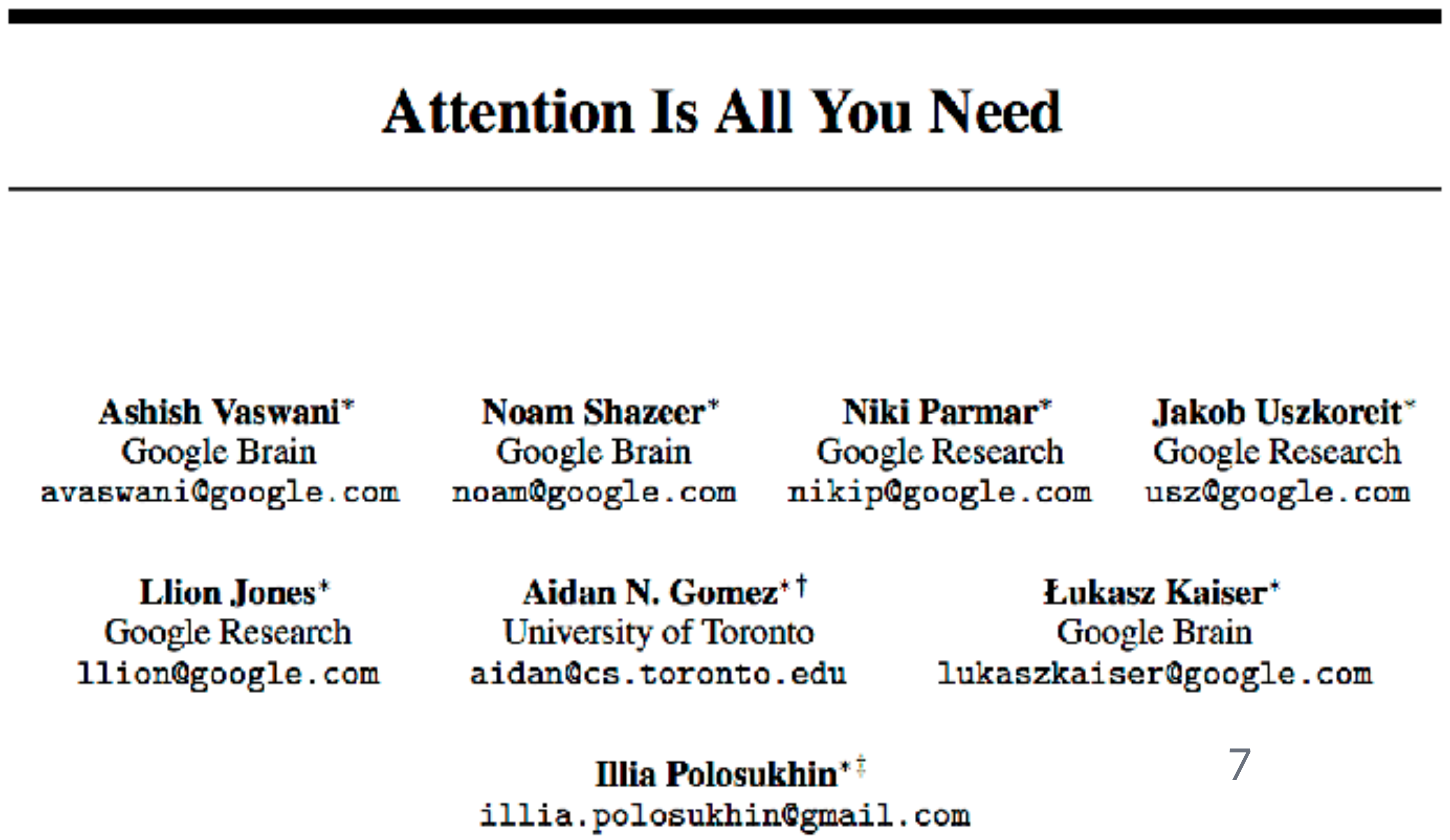
Deep Recurrent Models with Fast-Forward Connections for Neural Machine Translation

Jie Zhou Ying Cao Xuguang Wang Peng Li Wei Xu
Baidu Research - Institute of Deep Learning
Baidu Inc., Beijing, China
{zhoujie01,caoying03,wangxuguang,lipeng17,wei.xu}@baidu.com



Convolutional Sequence to Sequence Learning

Jonas Gehring[1] **Michael Auli**[1] **David Grangier**[1] **Denis Yarats**[1] **Yann N. Dauphin**[1]



Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar*
Google Research
nikip@google.com

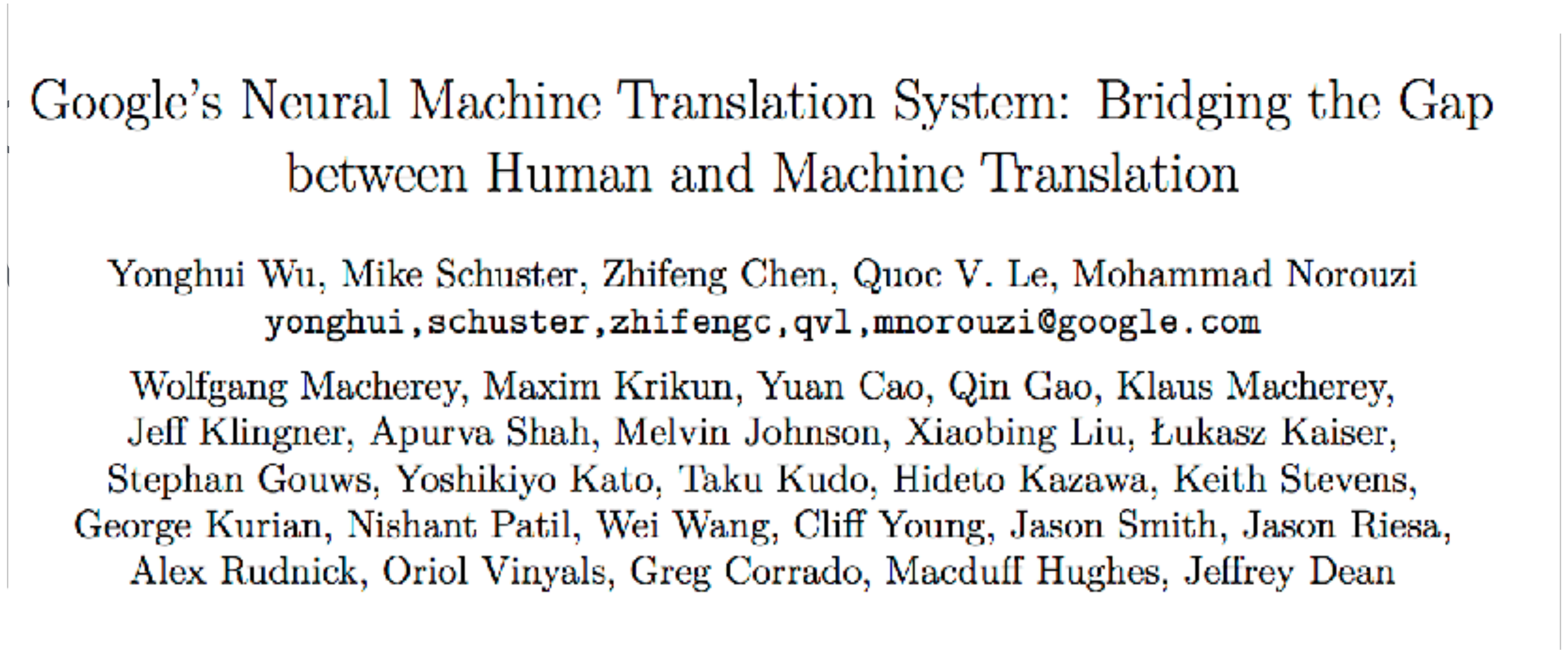
Jakob Uszkoreit*
Google Research
usz@google.com

Llion Jones*
Google Research
llion@google.com

Aidan N. Gomez* †
University of Toronto
aidan@cs.toronto.edu

Łukasz Kaiser*
Google Brain
lukaszkaiser@google.com

Illia Polosukhin* ‡
illia.polosukhin@gmail.com

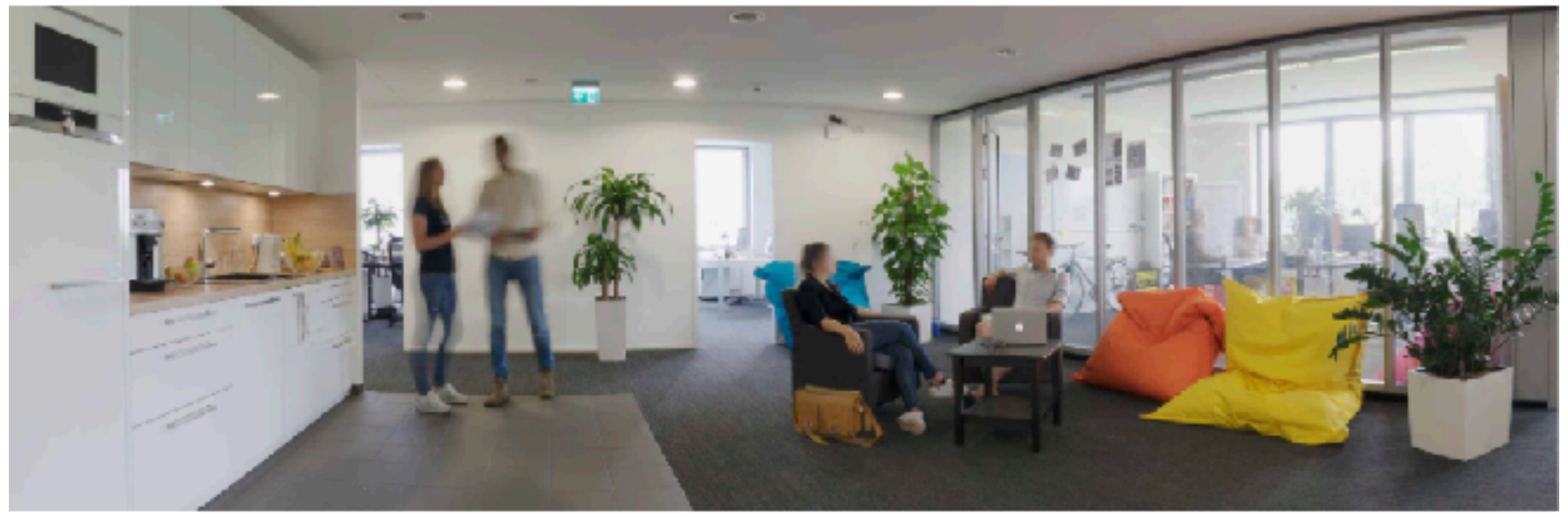


Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi
yonghui,schuster,zhifengc,qvl,mnorouzi@google.com

Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, Jeffrey Dean

DeepL



Press Information – DeepL Translator Launch

Architecture: Sequence to Sequence Learning with CNNs

Convolutional Sequence to Sequence Learning.
Jonas Gehring, Michael Auli, Yann Dauphin, David Grangier.
ICML 2017.
https://arxiv.org/abs/1711.04956

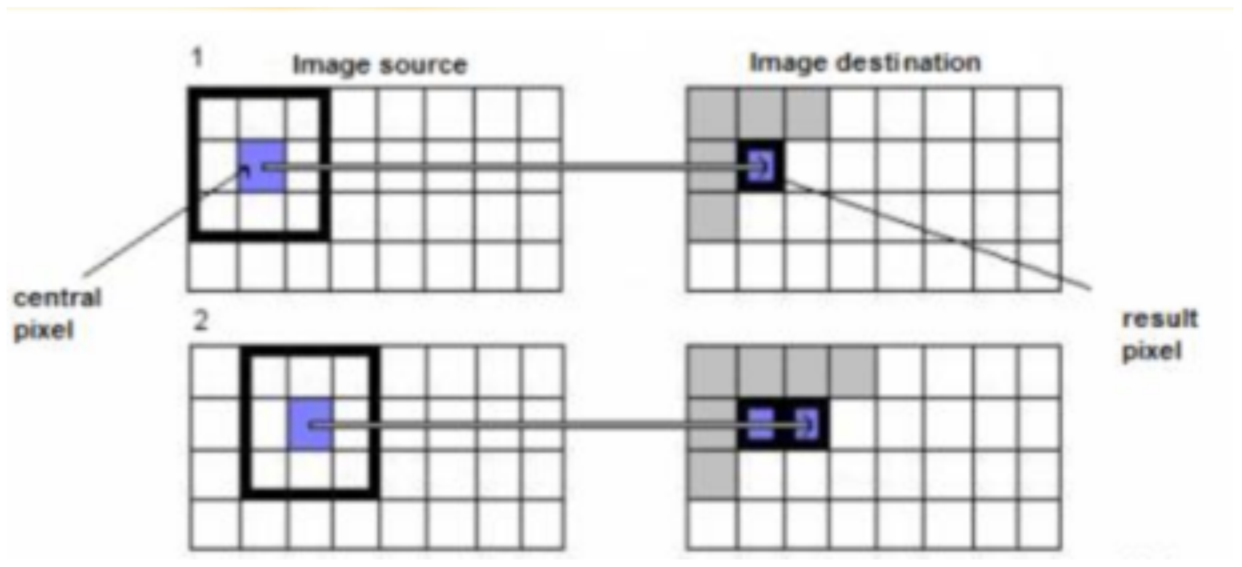
Convolutions vs Recurrent Networks

CNN

1d, 2d, 3d...

vision

convolutional filter

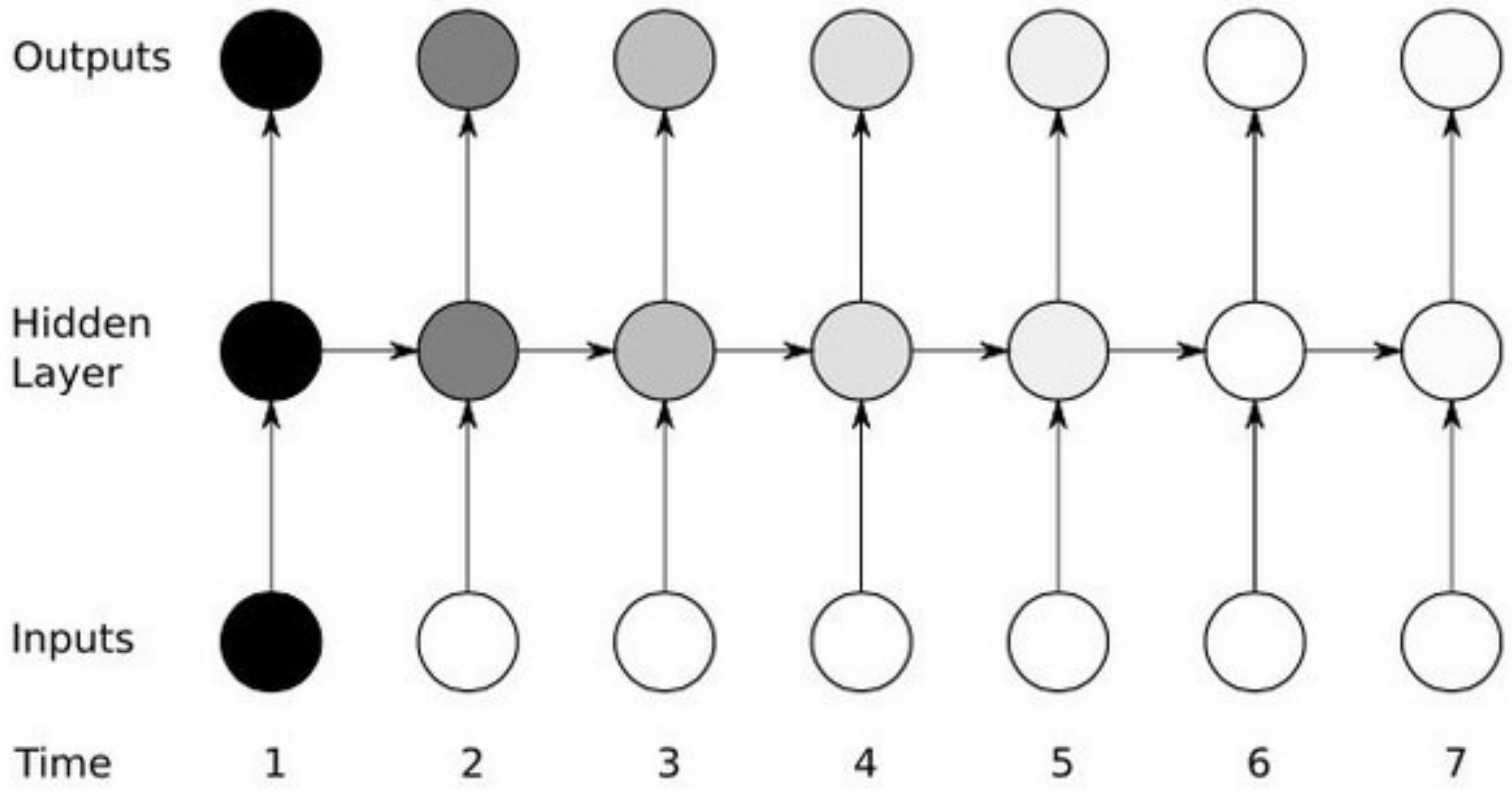


RNN

1d

language, speech

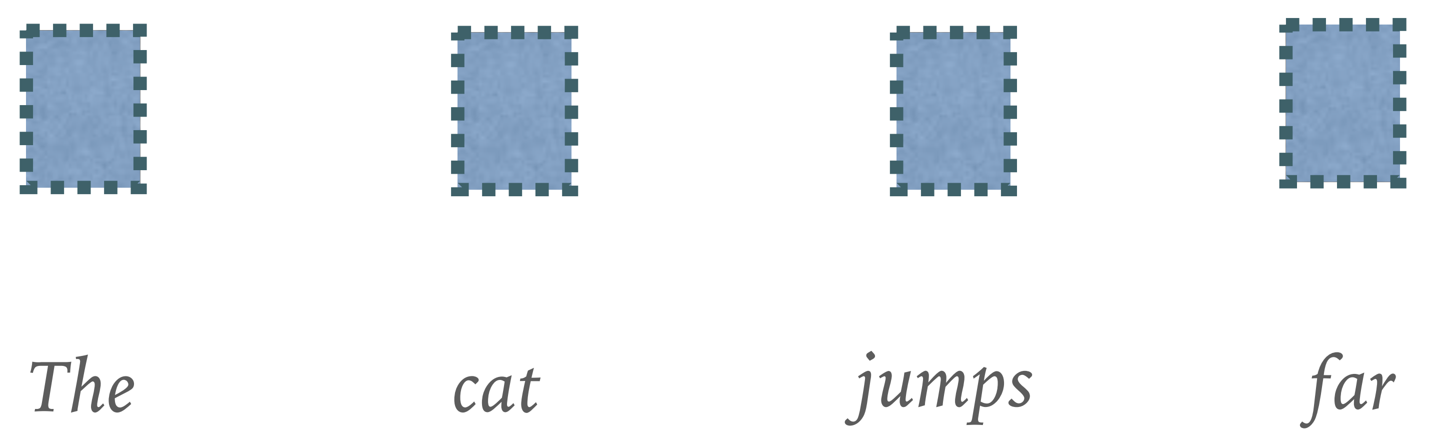
autoregressive filter



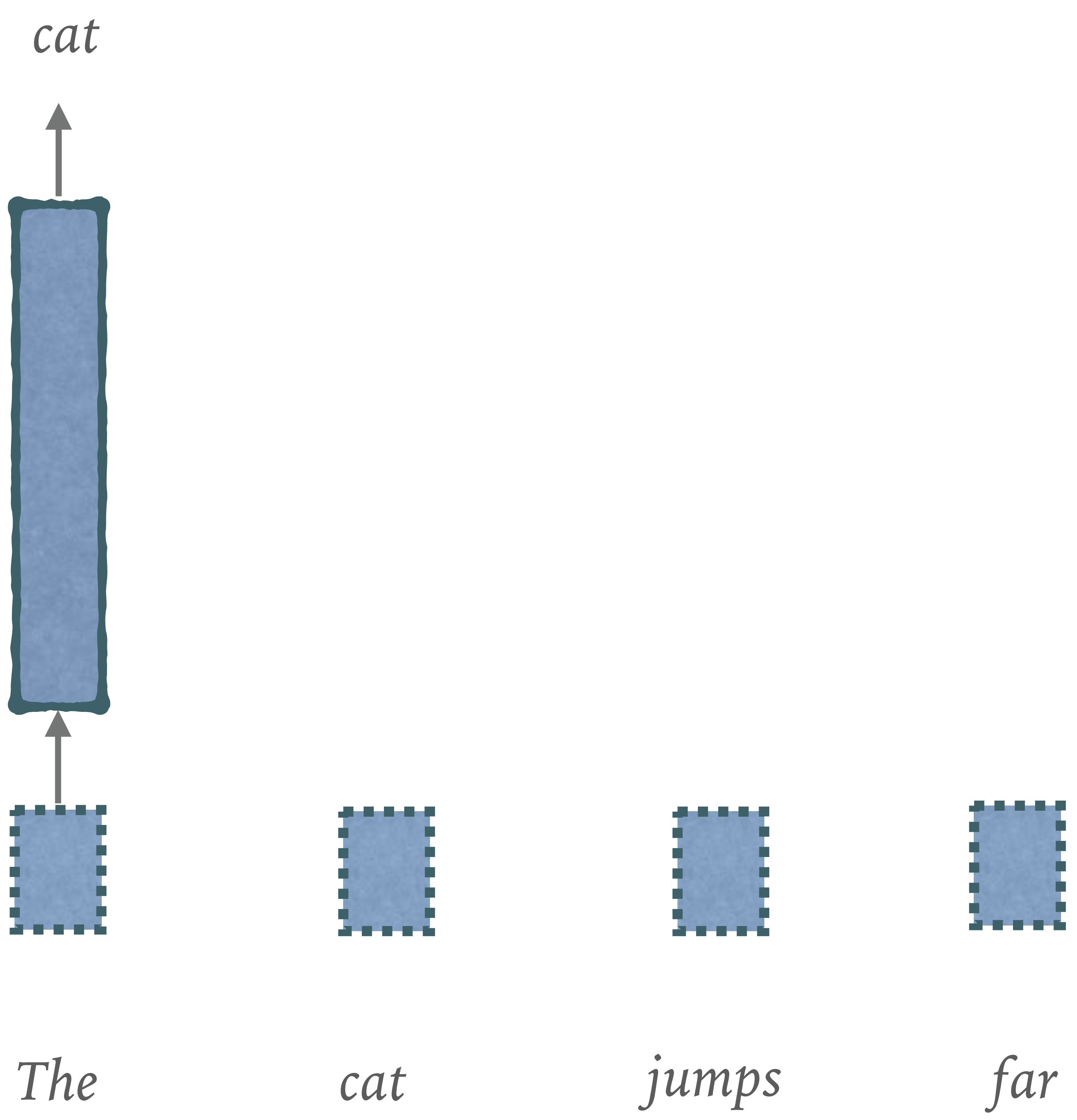
Convolutions vs Recurrent Networks

CNN	RNN
1d, 2d, 3d...	1d
vision	language, speech
convolutional filter	autoregressive filter
bounded dependencies	unbounded dep. (theory)
highly parallel	sequential

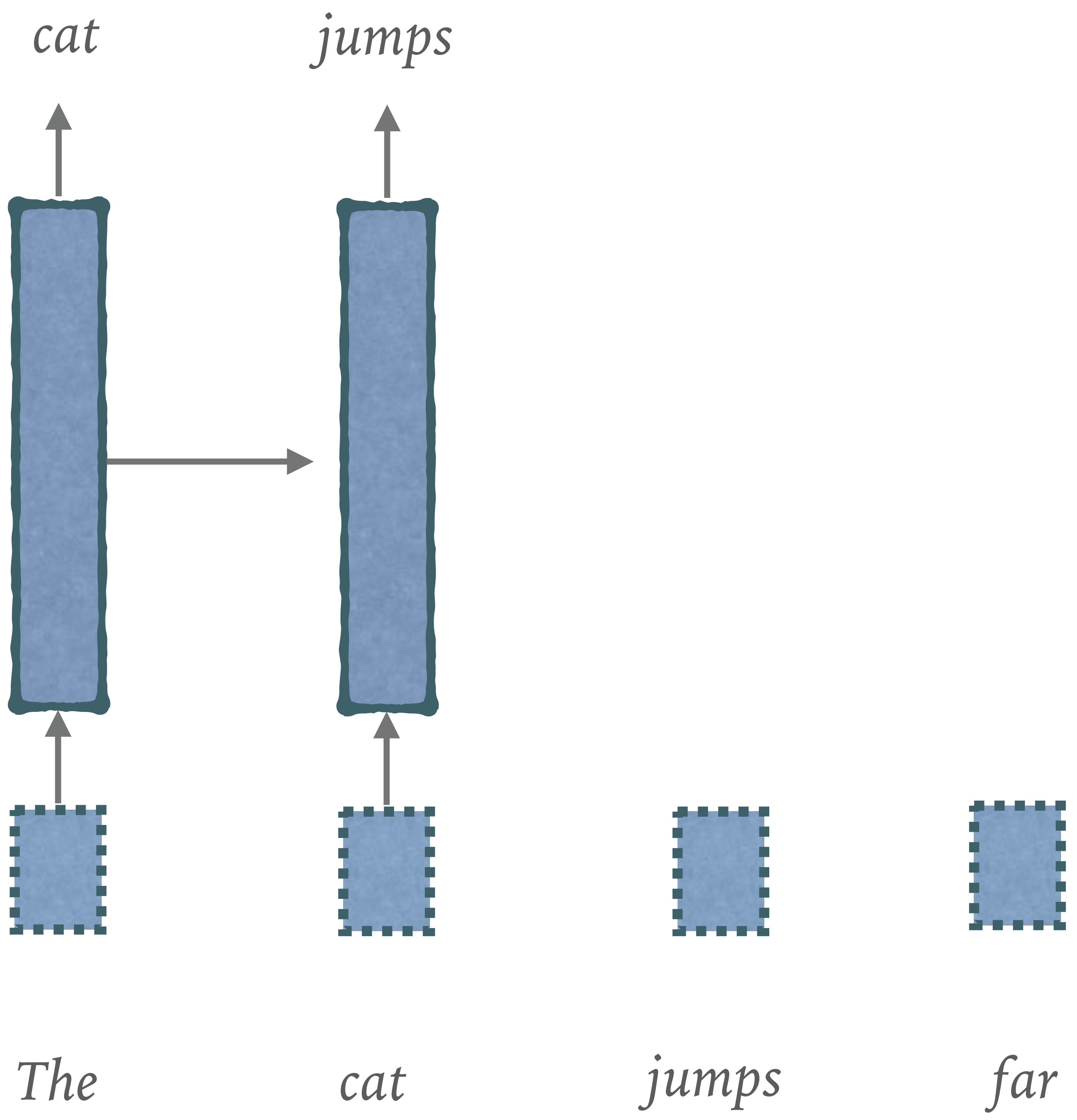
Recurrent Neural Network



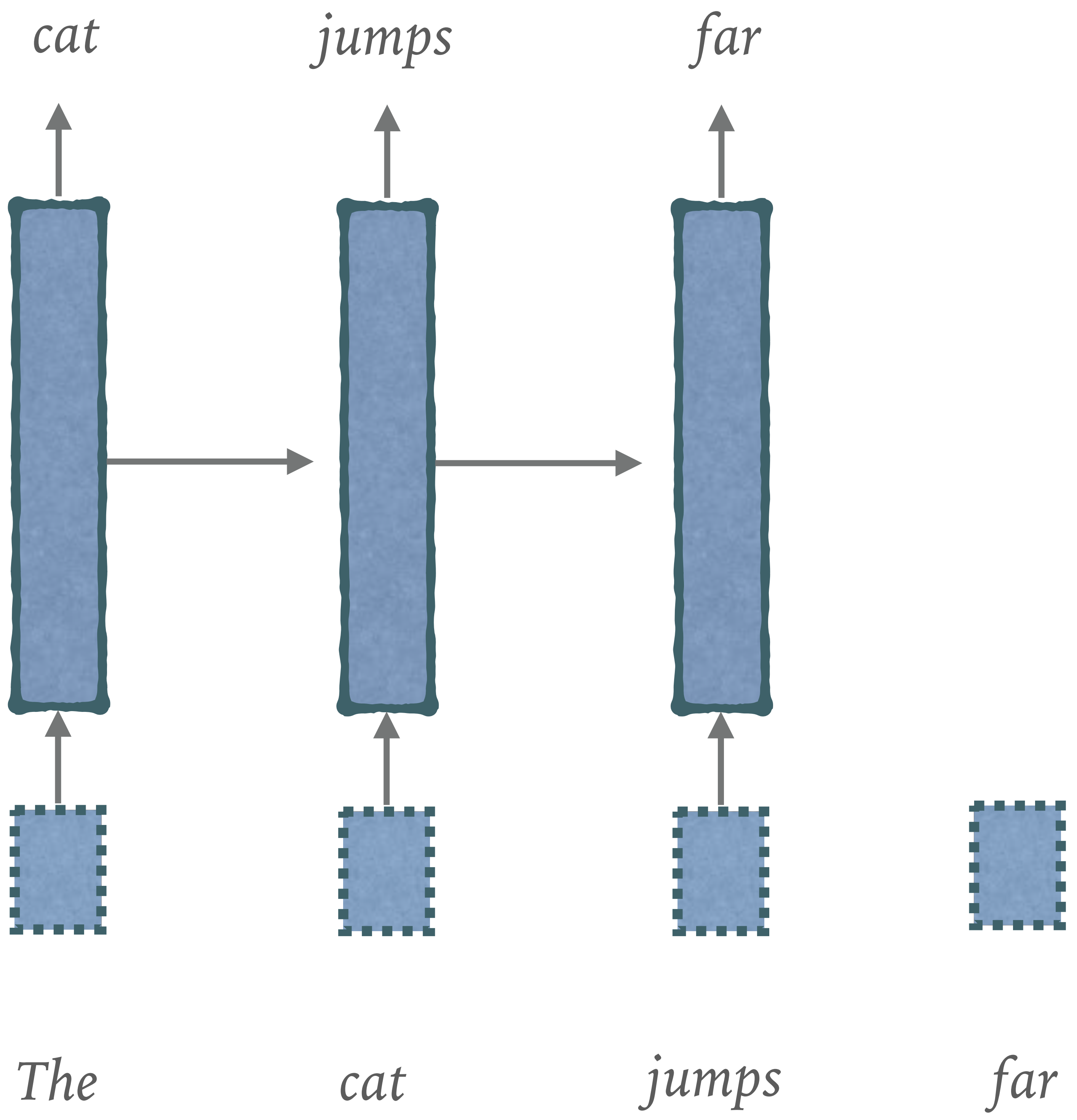
Recurrent Neural Network



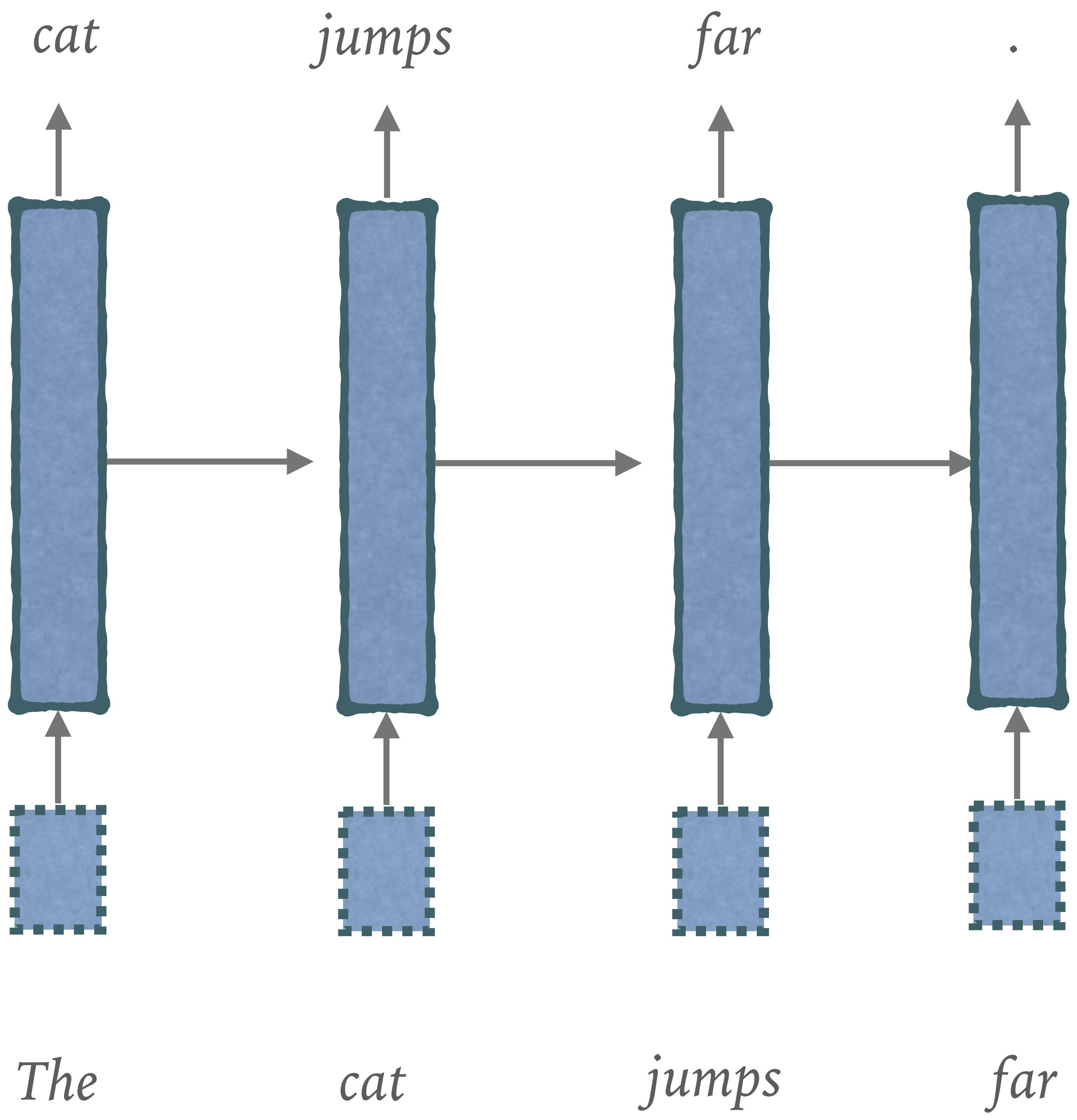
Recurrent Neural Network



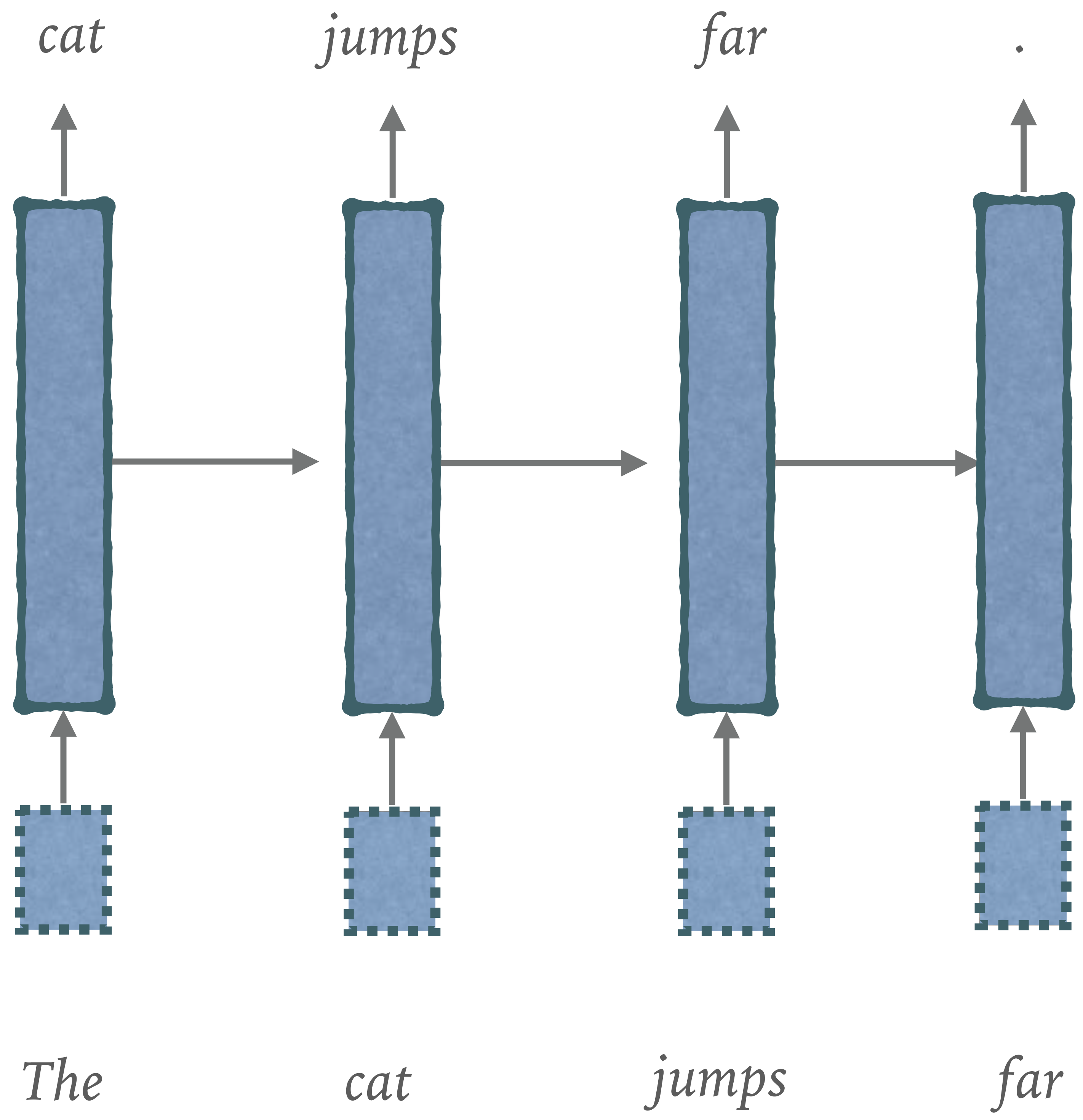
Recurrent Neural Network



Recurrent Neural Network

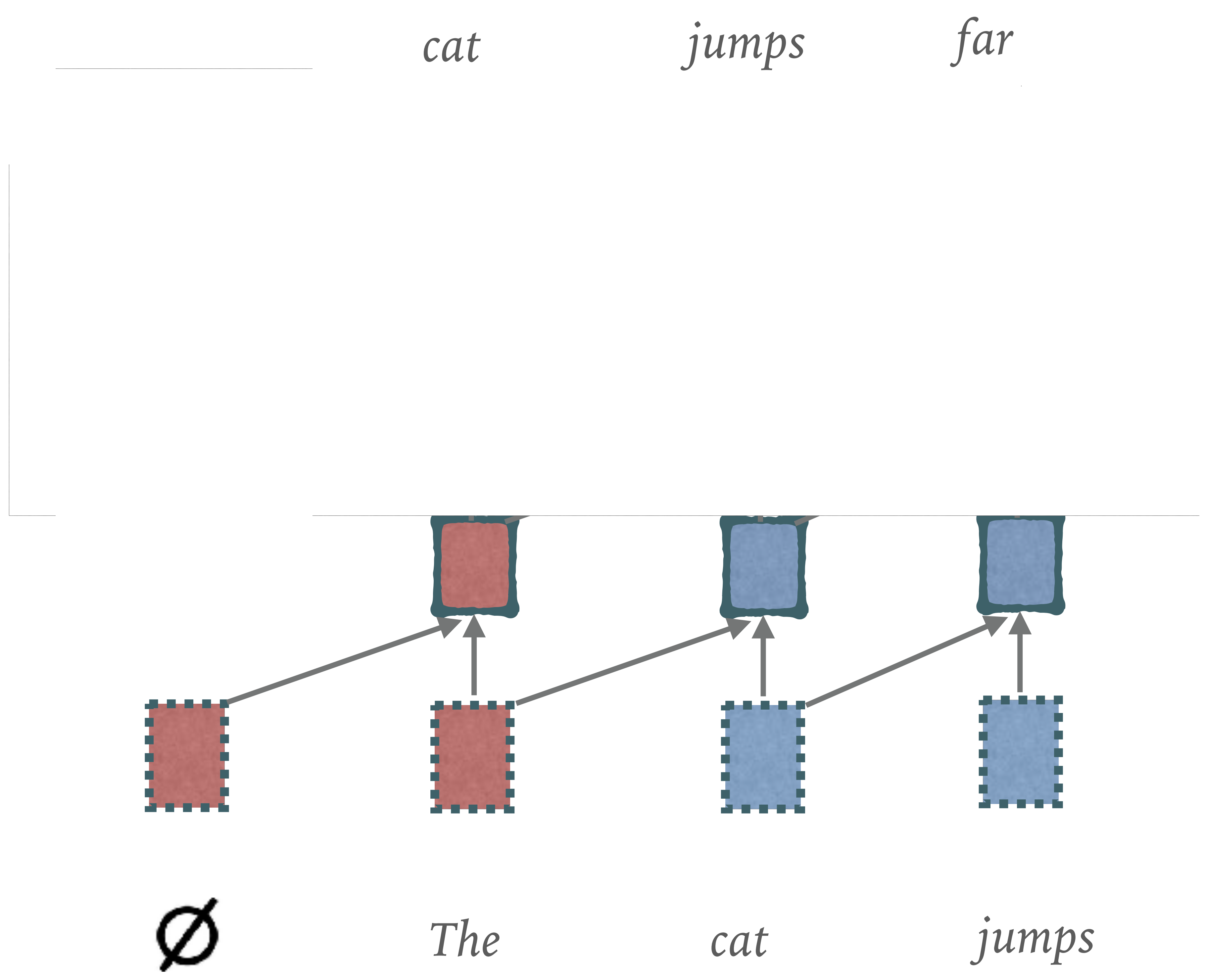


Recurrent Neural Network



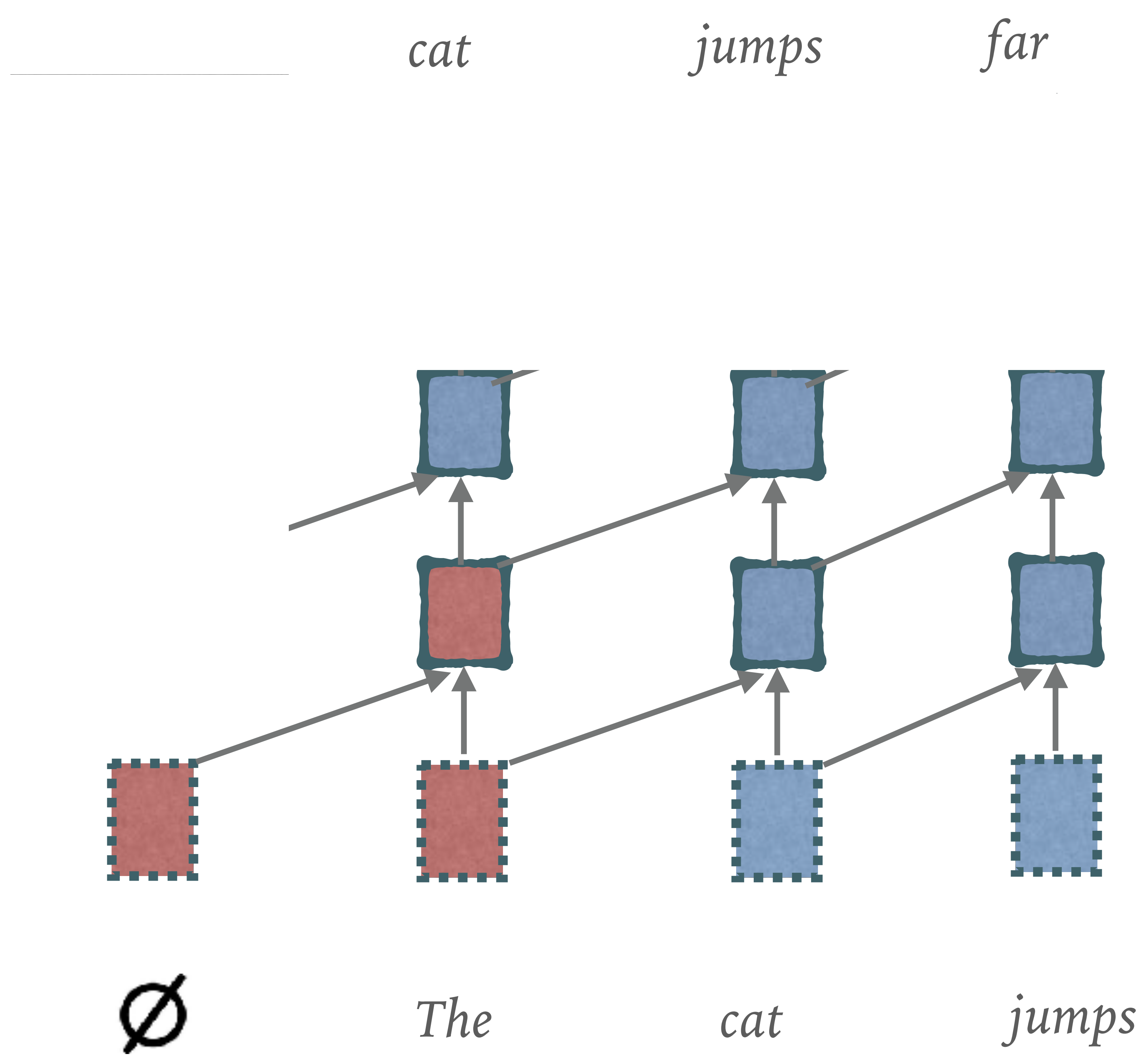
- O(T) sequential steps
- Recurrent connection causes vanishing gradient
- Are the recurrent connections necessary?

Convolutional Neural Network



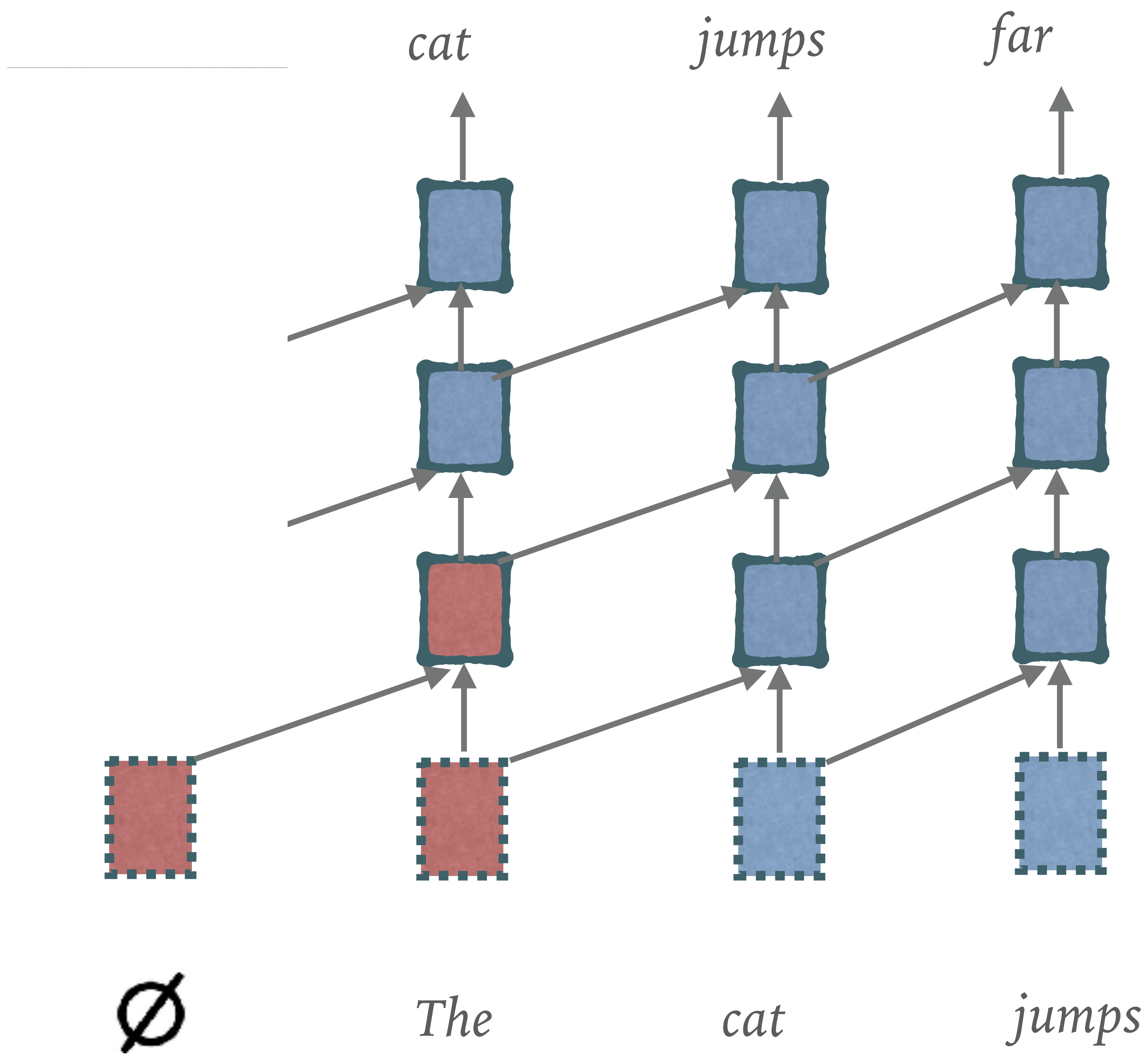
- Time Delay Neural Network (Waibel et al., 1989)
- O(1) sequential steps
- Incrementally build context of context windows
- Builds **hierarchical** structure

Convolutional Neural Network



- Time Delay Neural Network (Waibel et al., 1989)
- O(1) sequential steps
- Incrementally build context of context windows
- Builds **hierarchical** structure

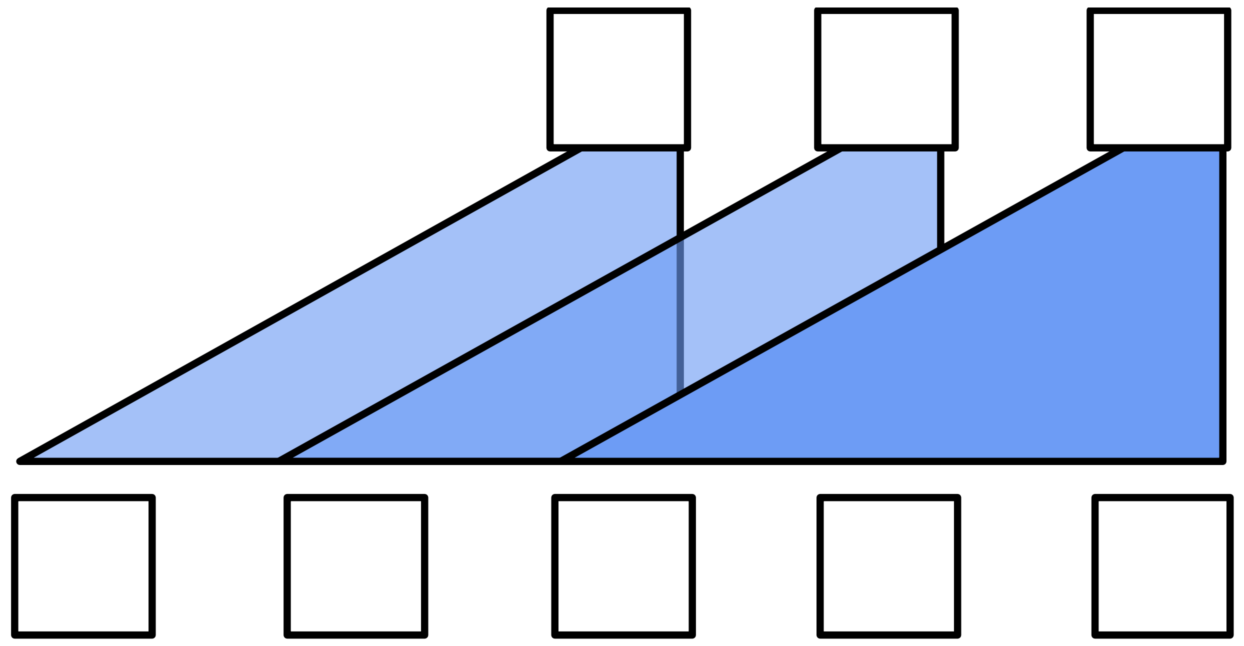
Convolutional Neural Network



- Time Delay Neural Network (Waibel et al., 1989)
- O(1) sequential steps
- Incrementally build context of context windows
- Builds **hierarchical** structure

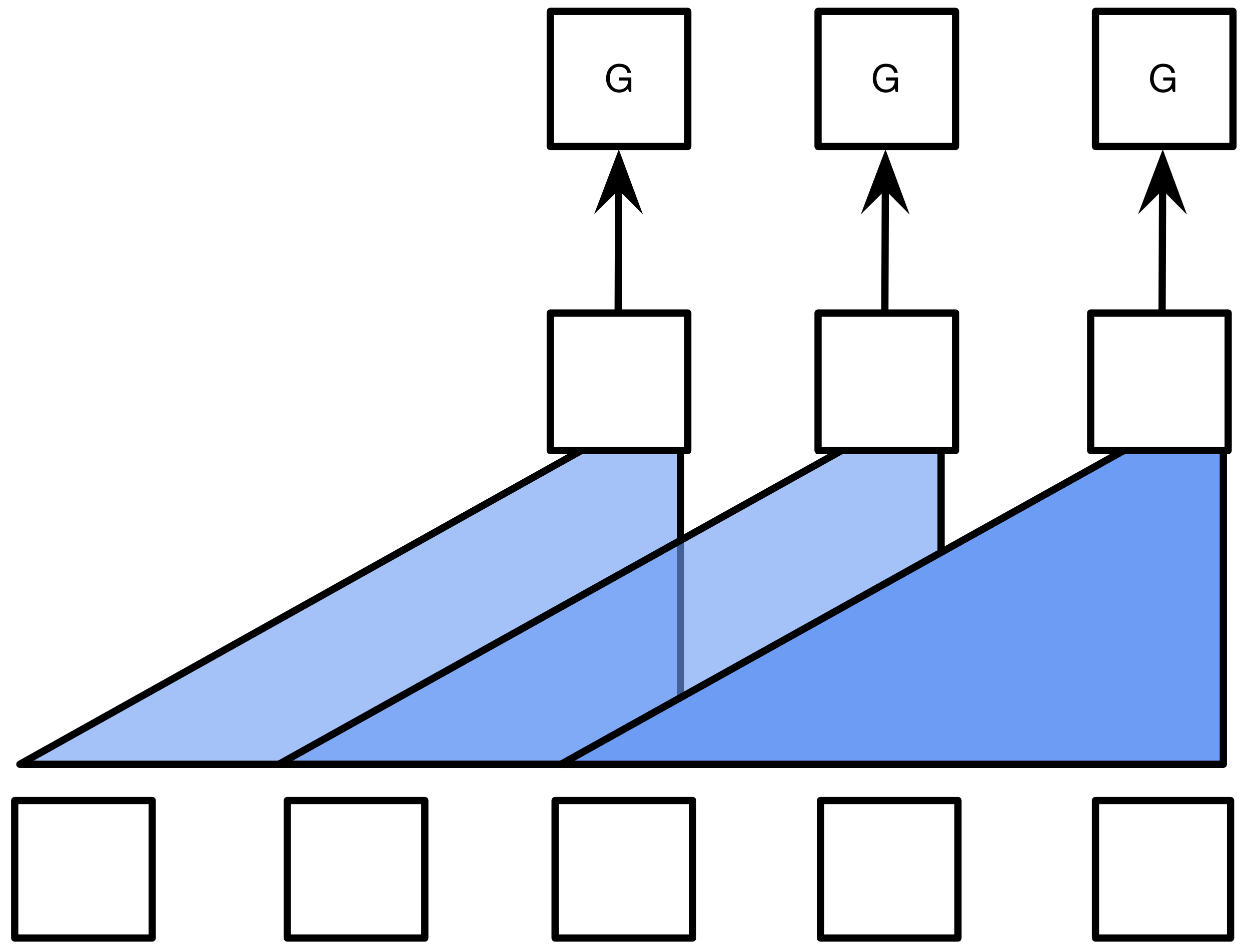
Gated Convolutional Neural Network

- Processes a sentence with a set of convolutions
- Each convolution learns higher level features
- Gates filter information to propagate up the hierarchy



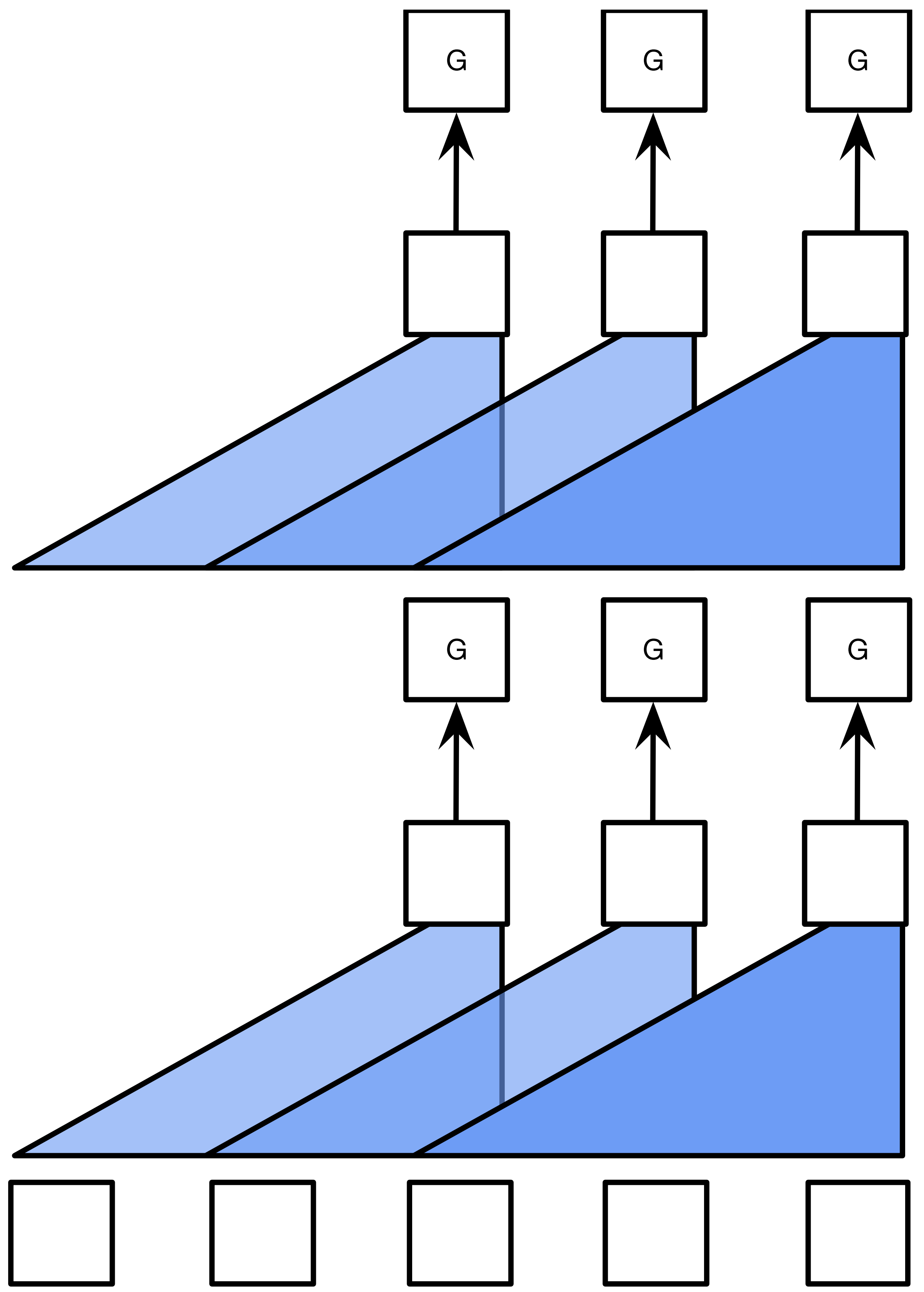
Gated Convolutional Neural Network

- Processes a sentence with a set of convolutions
- Each convolution learns higher level features
- Gates filter information to propagate up the hierarchy



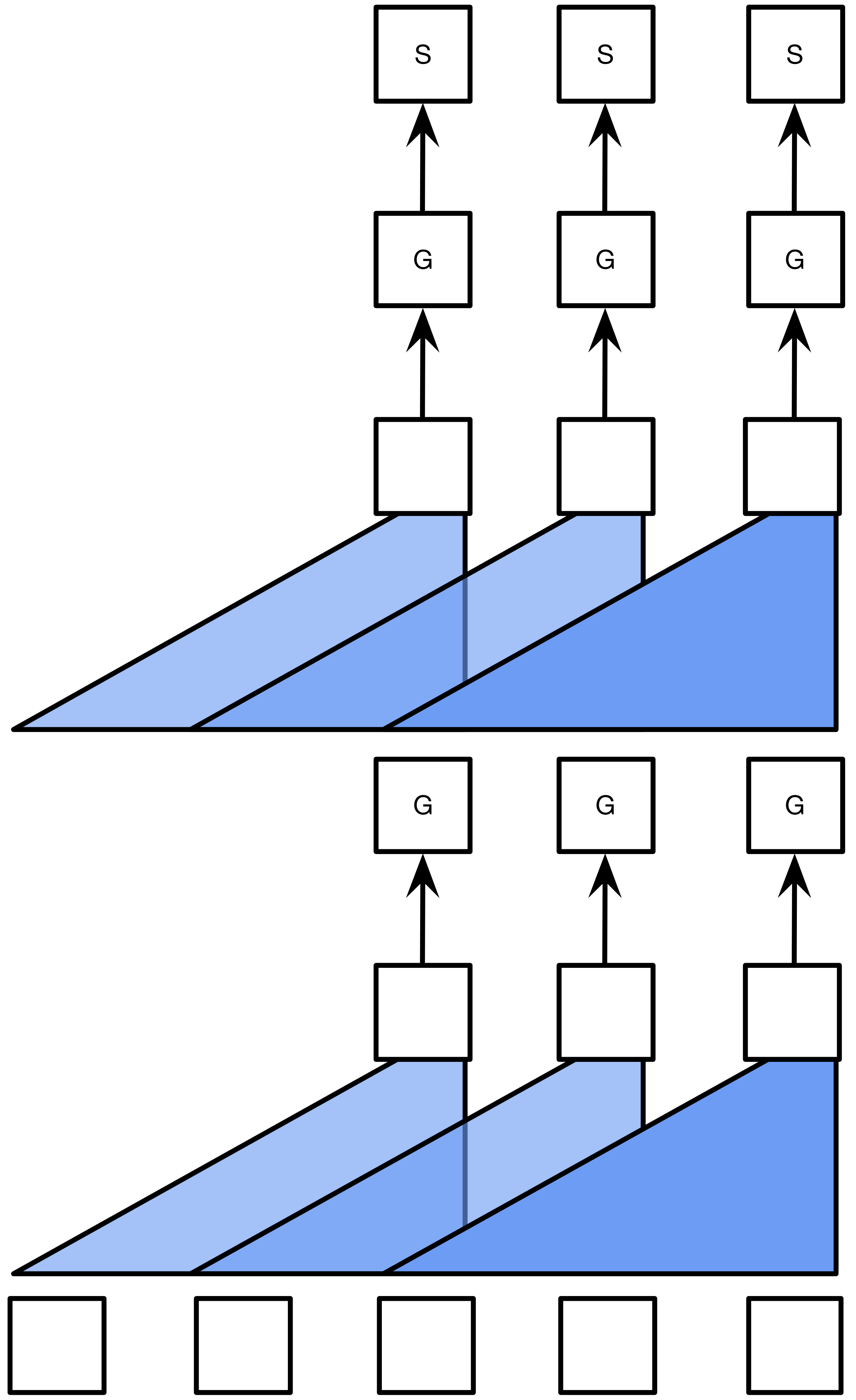
Gated Convolutional Neural Network

- Processes a sentence with a set of convolutions
- Each convolution learns higher level features
- Gates filter information to propagate up the hierarchy



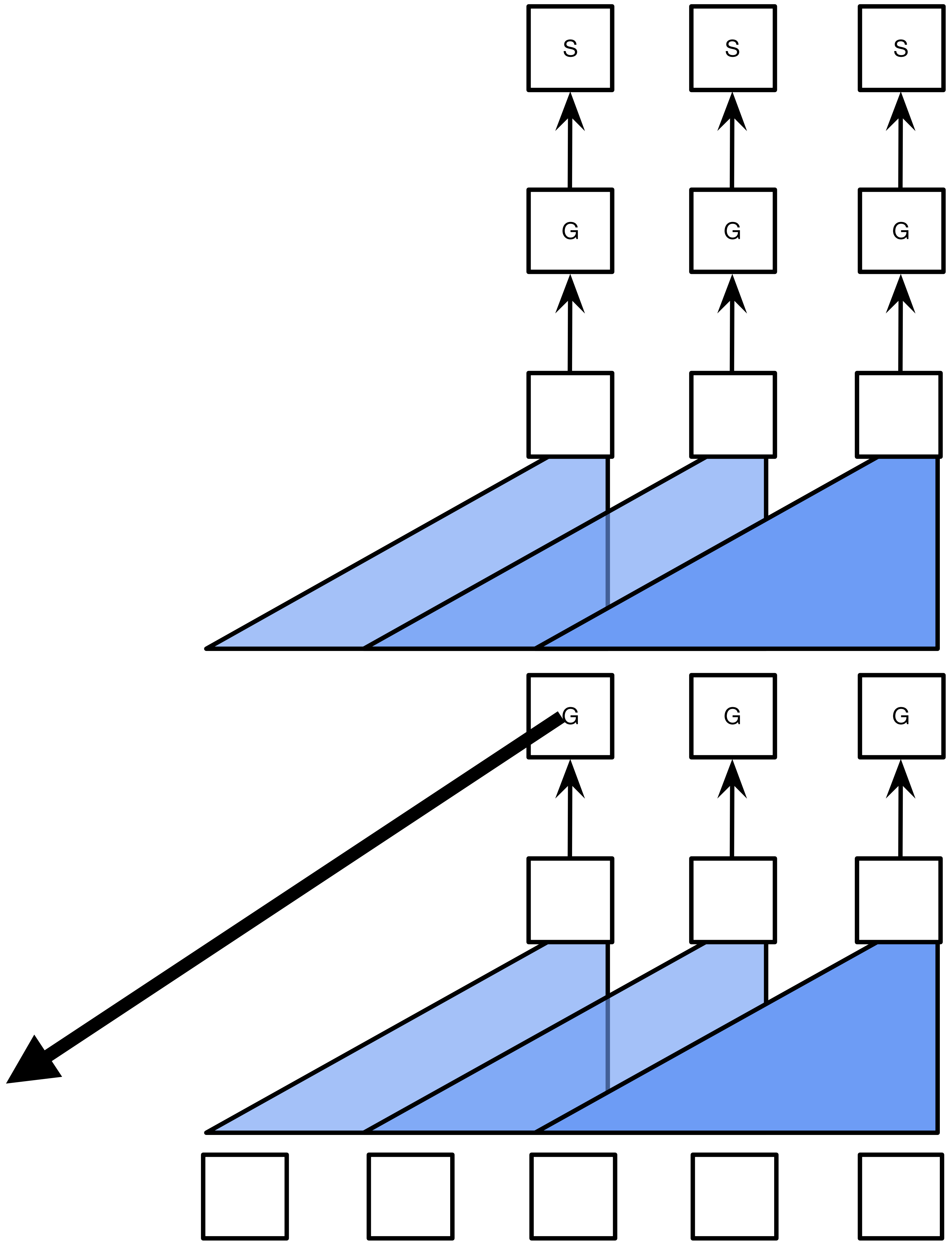
Gated Convolutional Neural Network

- Processes a sentence with a set of convolutions
- Each convolution learns higher level features
- Gates filter information to propagate up the hierarchy



Gated Convolutional Neural Network

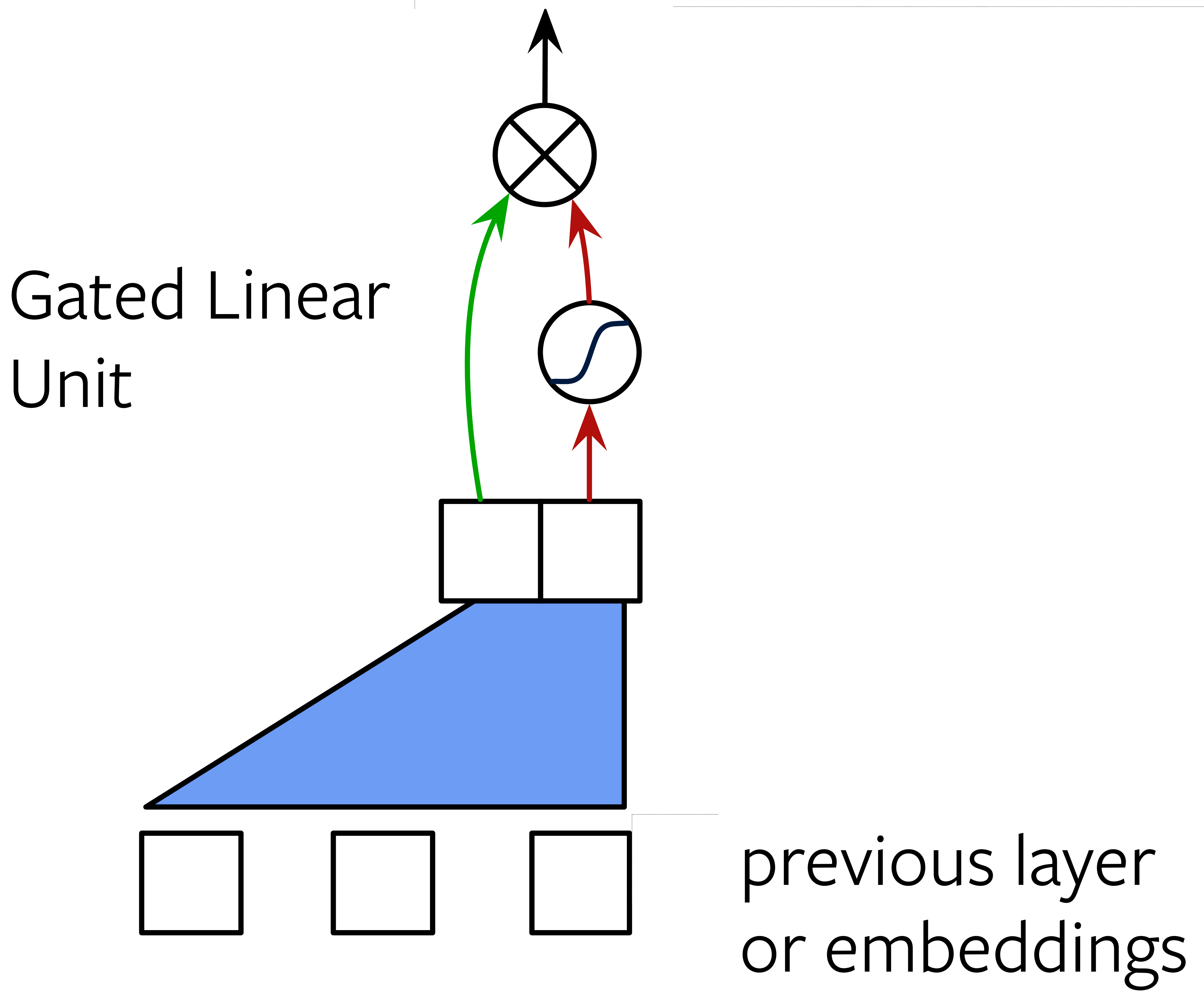
- Processes a sentence with a set of convolutions
- Each convolution learns higher level features
- Gates filter information to propagate up the hierarchy



$$y = x \otimes \sigma(x') \quad \text{(gated linear)}$$

Gated Linear Unit

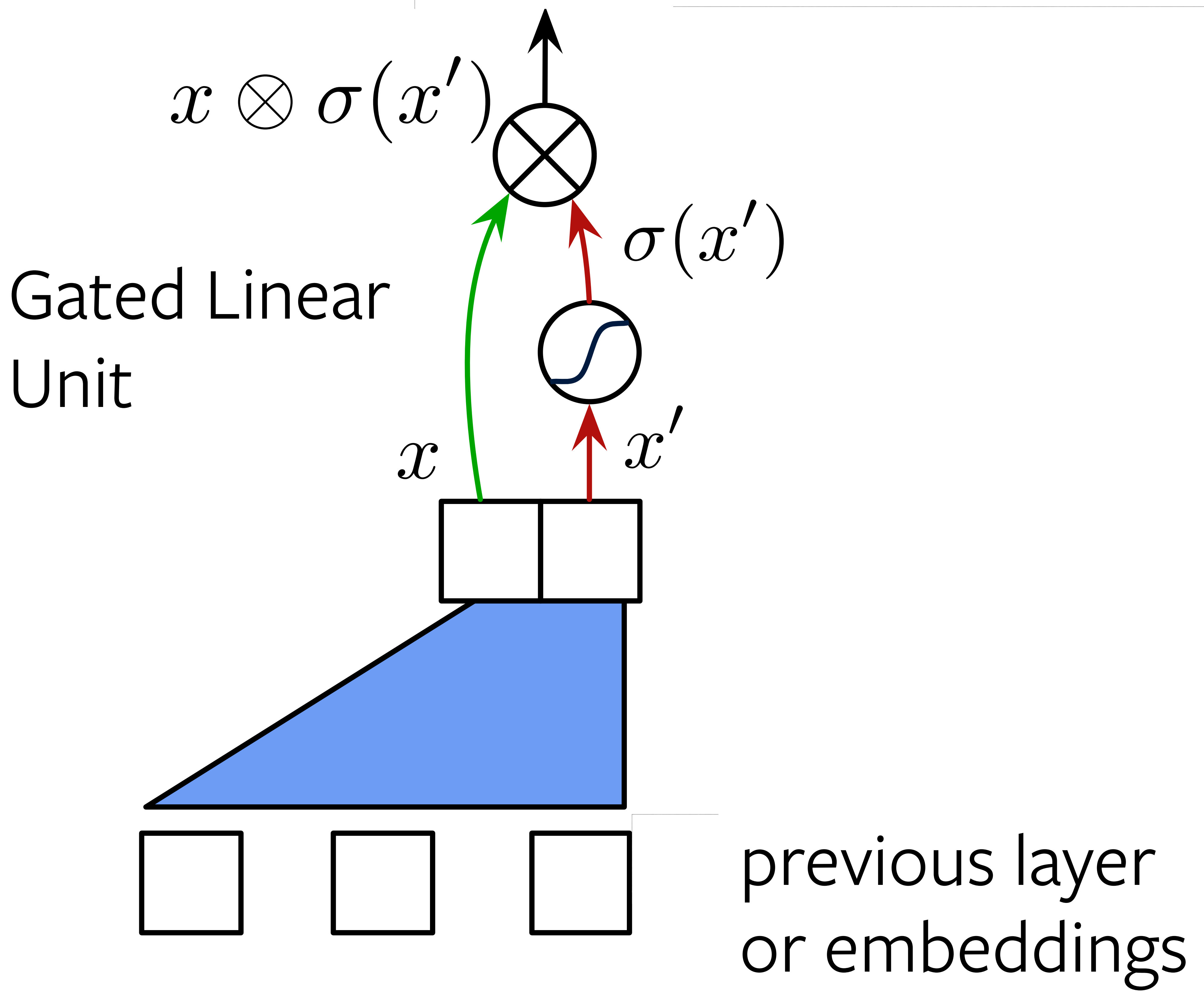
- The gated linear unit can be seen as a multiplicative skip connection
- We find this approach to gating improves performance



Similar to 'Swish'
(Ramachandran et al., 2017)

Gated Linear Unit

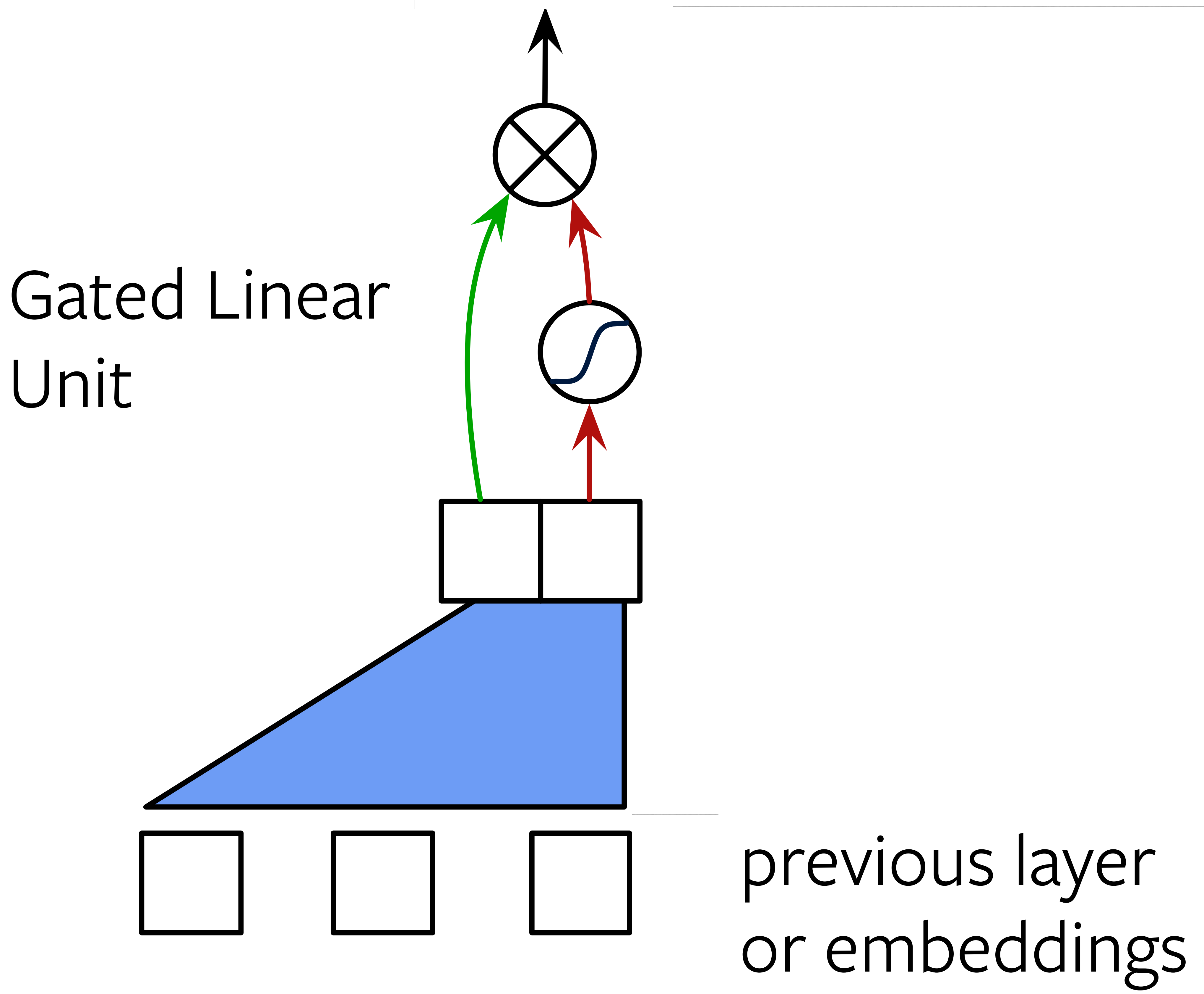
- The gated linear unit can be seen as a multiplicative skip connection
- We find this approach to gating improves performance



Similar to 'Swish'
(Ramachandran et al., 2017)

Gated Linear Unit

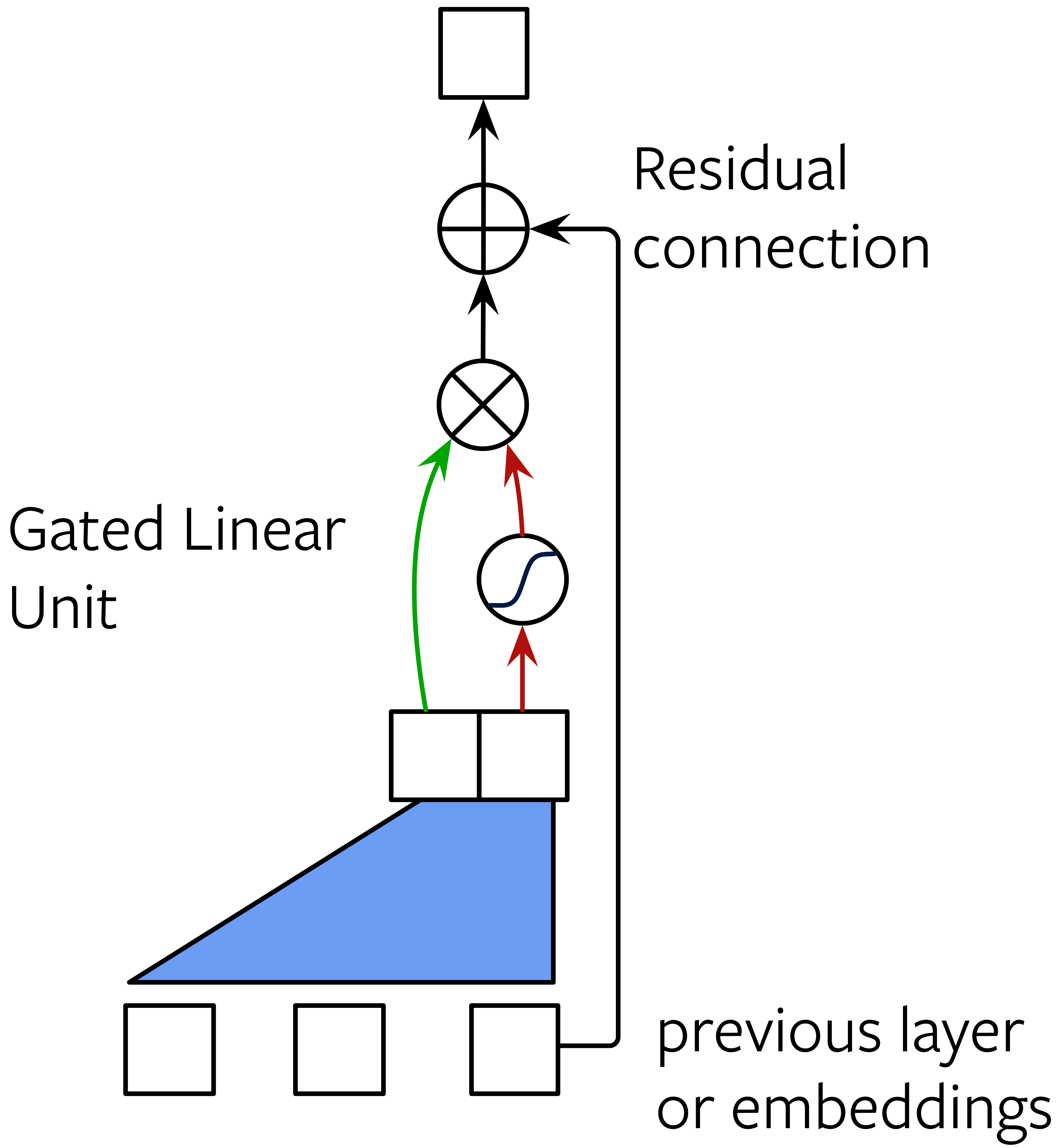
- The gated linear unit can be seen as a multiplicative skip connection
- We find this approach to gating improves performance



Similar to 'Swish' (Ramachandran et al., 2017)

Gated Linear Unit

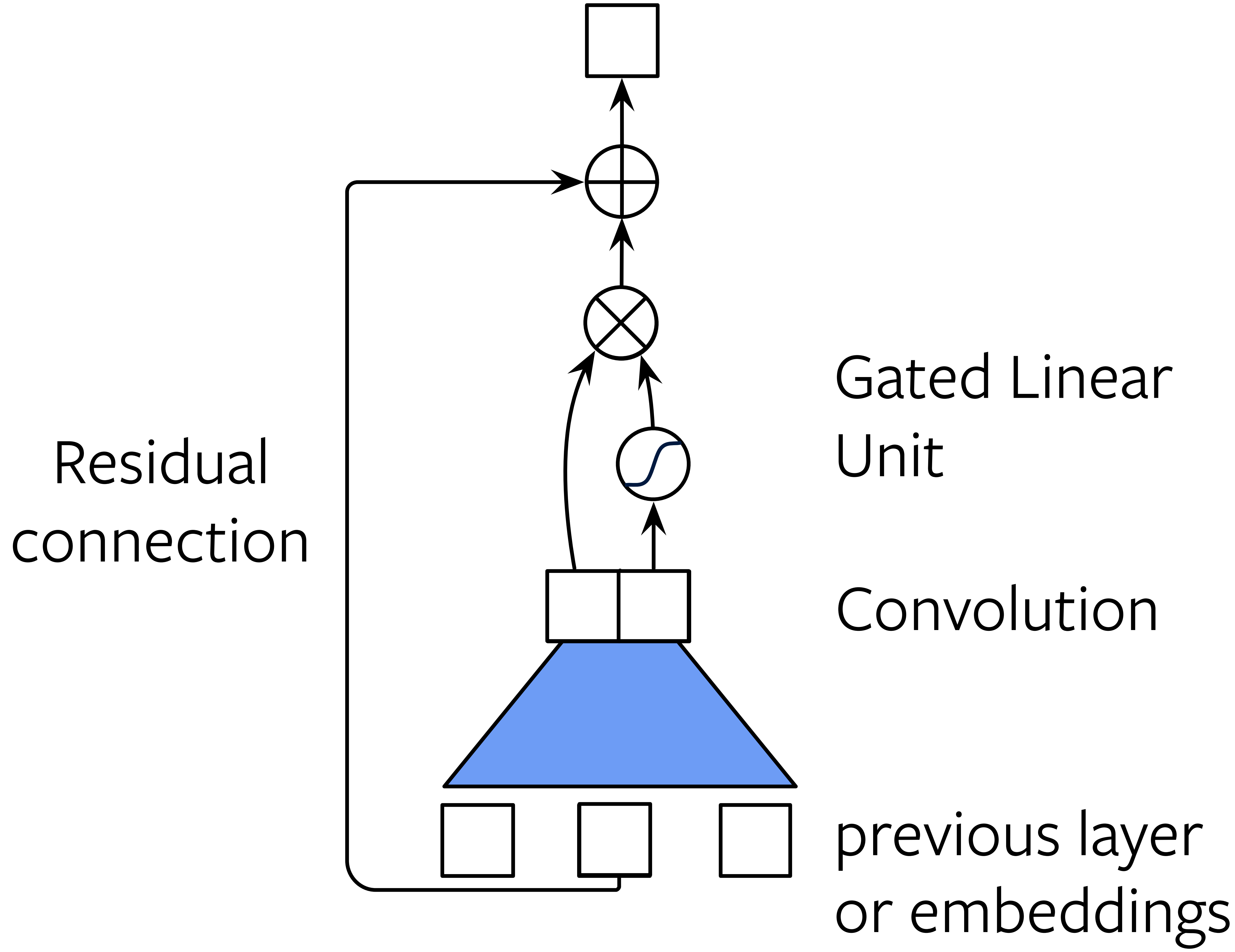
- The gated linear unit can be seen as a multiplicative skip connection
- We find this approach to gating improves performance



Similar to 'Swish'
(Ramachandran et al., 2017)

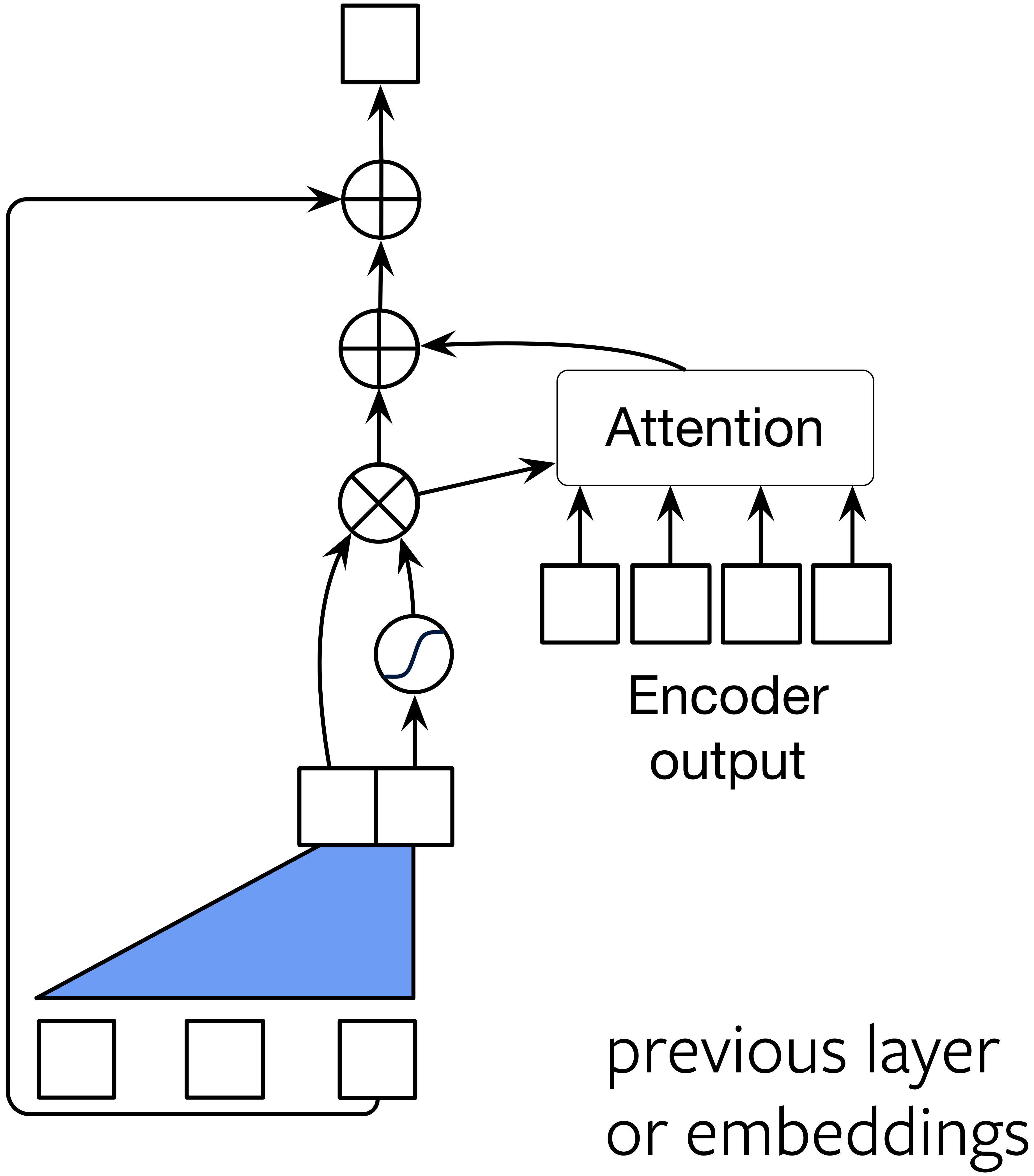
Convolutional S2S: Encoder

- Input: word + position embeddings: 1, 2, 3, ...
- Weight Normalization (Salimans & Kingma, 2016)
- No batch or layer norm: initialization (He at al. ′15) and scale by sqrt(1/2)
- Repeat N times

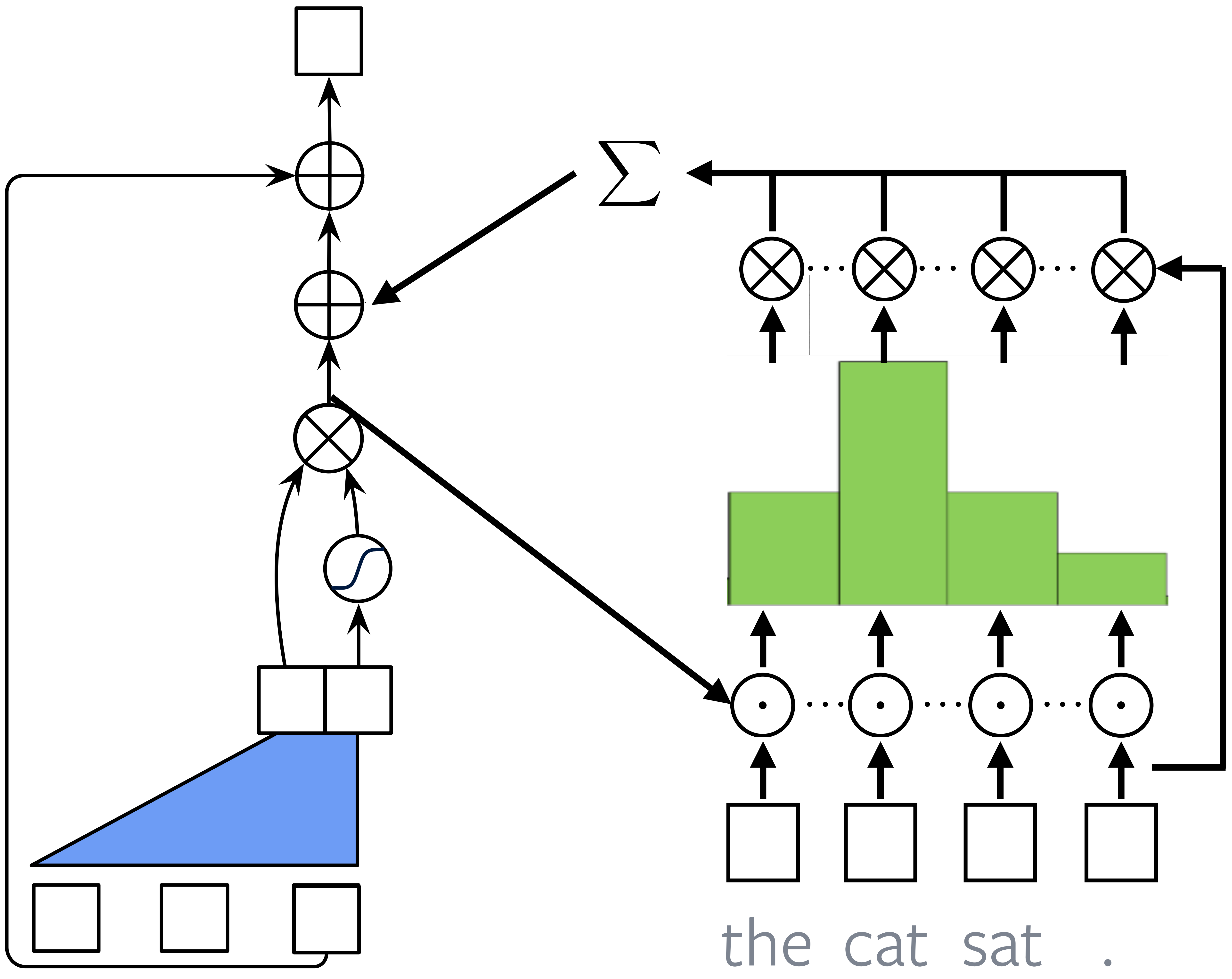


Convolutional S2S: Decoder

- Input: word embeddings
  + position embeddings: 1, 2, 3, ...
- ***Causal*** convolution over generated sequence so far
- **Dot-product attention** at every layer



Convolutional S2S: Attention



weighted sum

attention weights

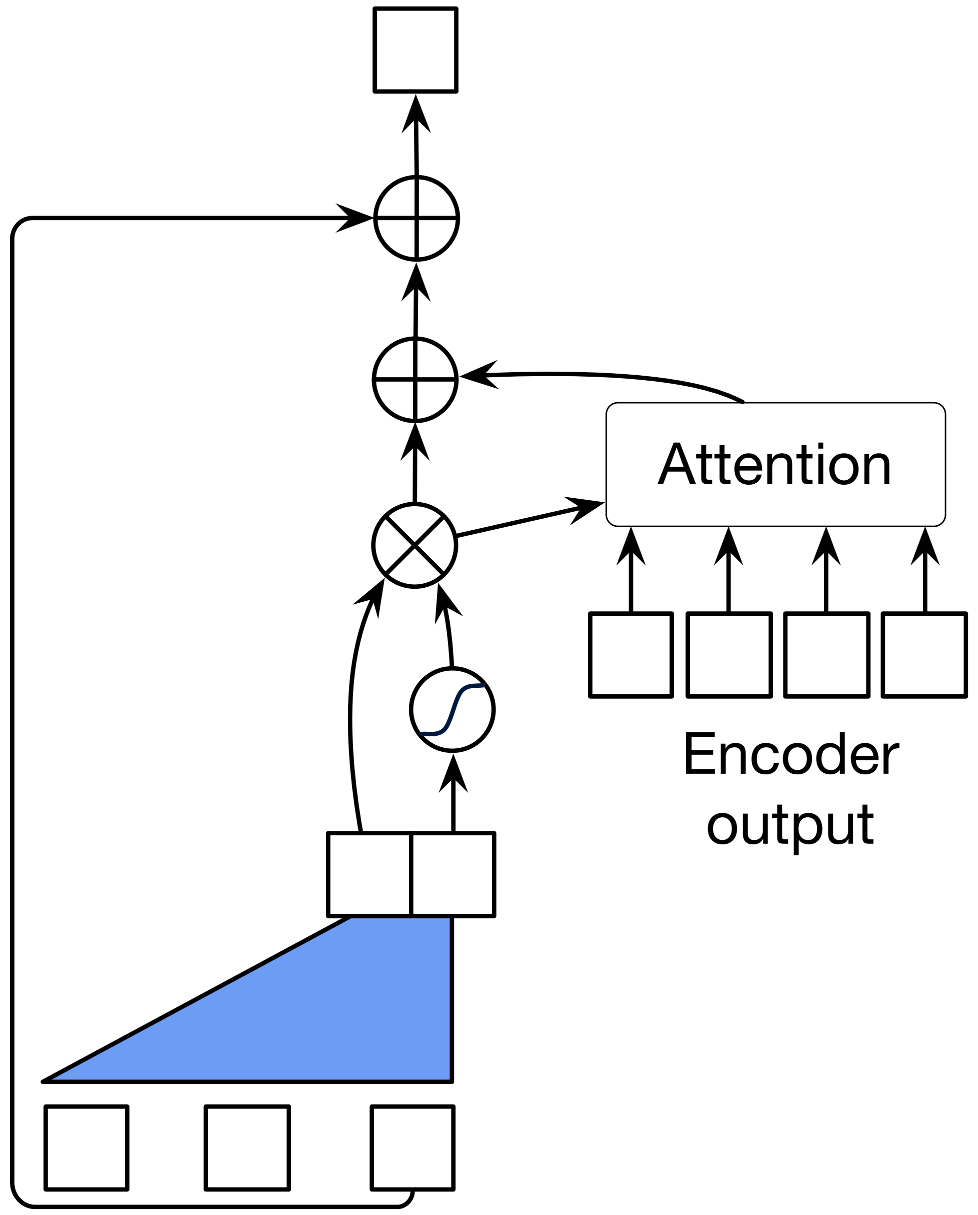
encoder output

source sentence

previous decoder
layer outputs

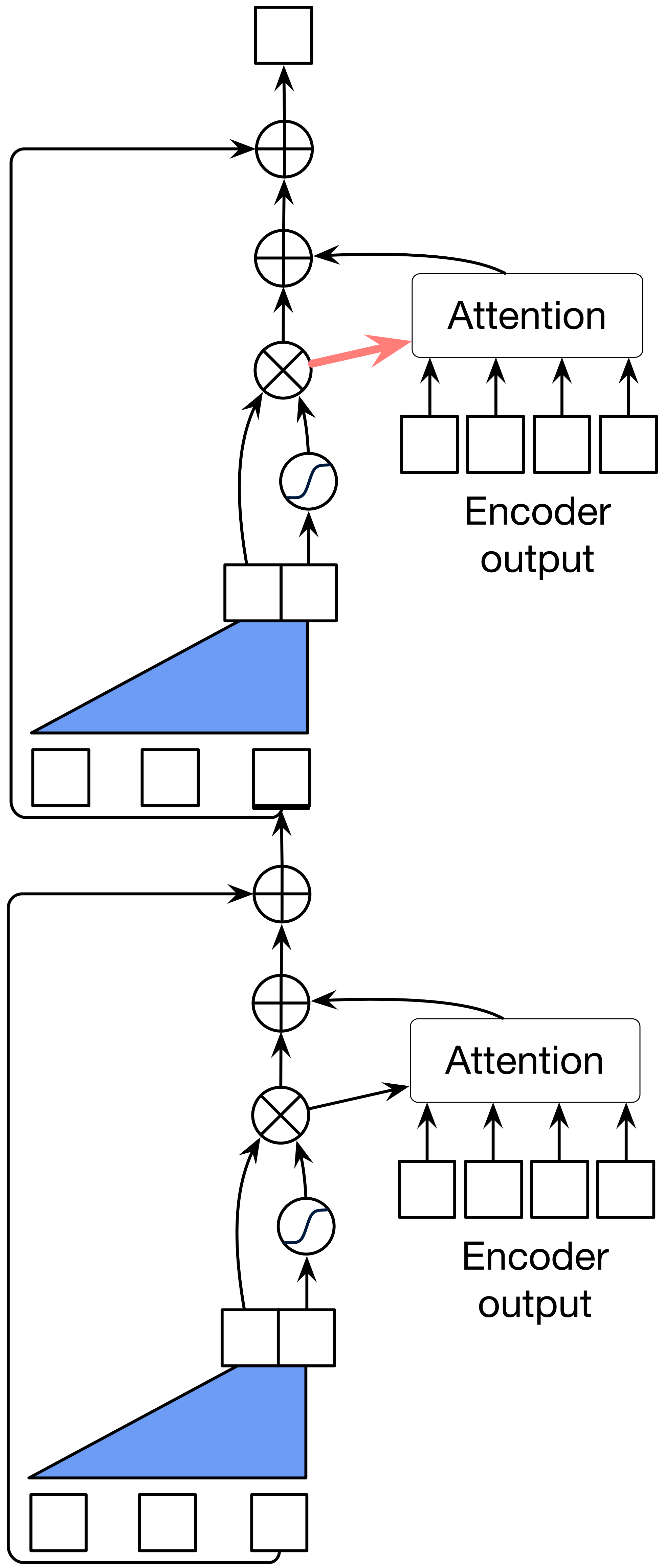
Convolutional S2S: Multi-hop Attention

- Attention in every decoder layer
- Queries contain information about previous source contexts



Convolutional S2S: Multi-hop Attention

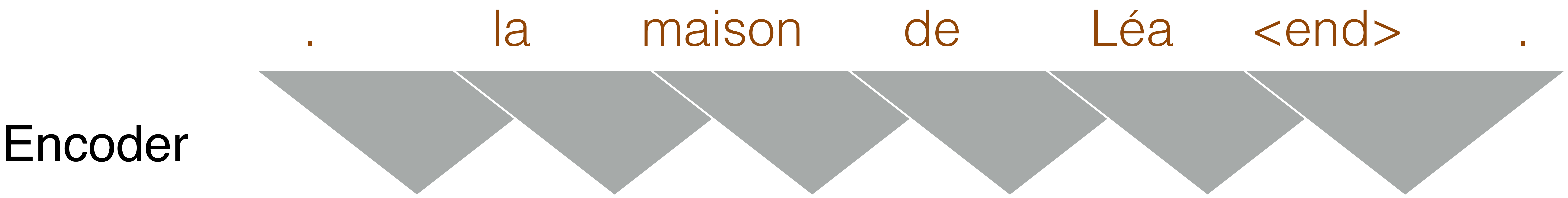
- Attention in every decoder layer
- Queries contain information about previous source contexts

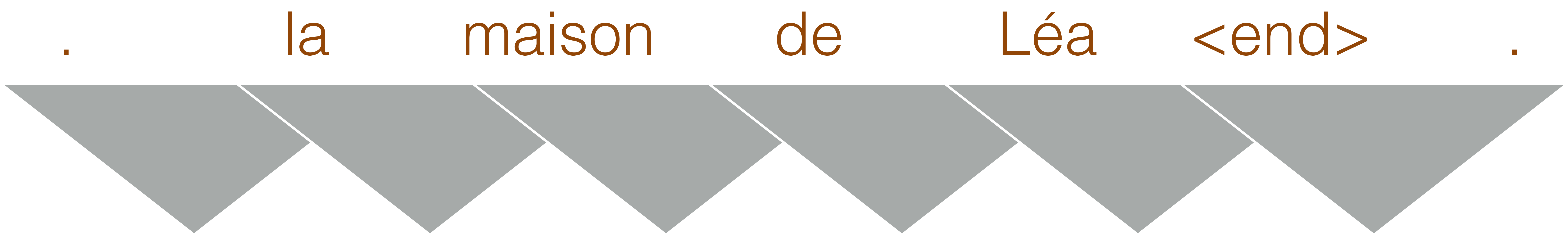


. la maison de Léa <end> .

. la maison de Léa <end> .

Encoder


.
la
maison
de
Léa
<end>
.
Encoder

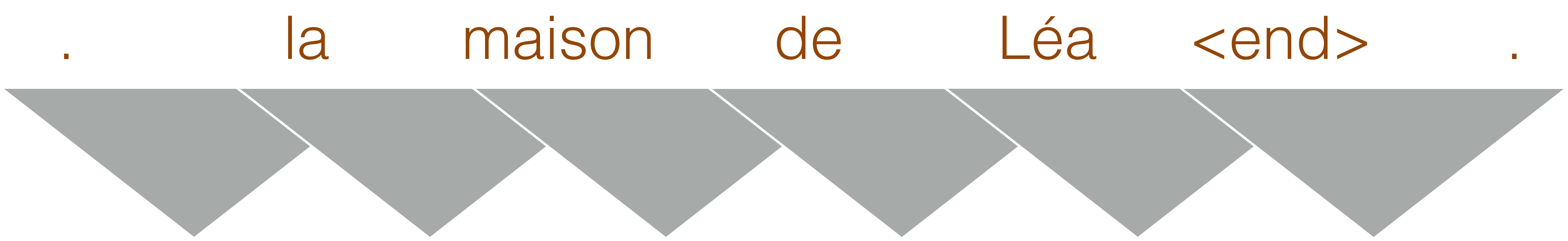


. la maison de Léa <end> .

Encoder

Decoder

. <start>

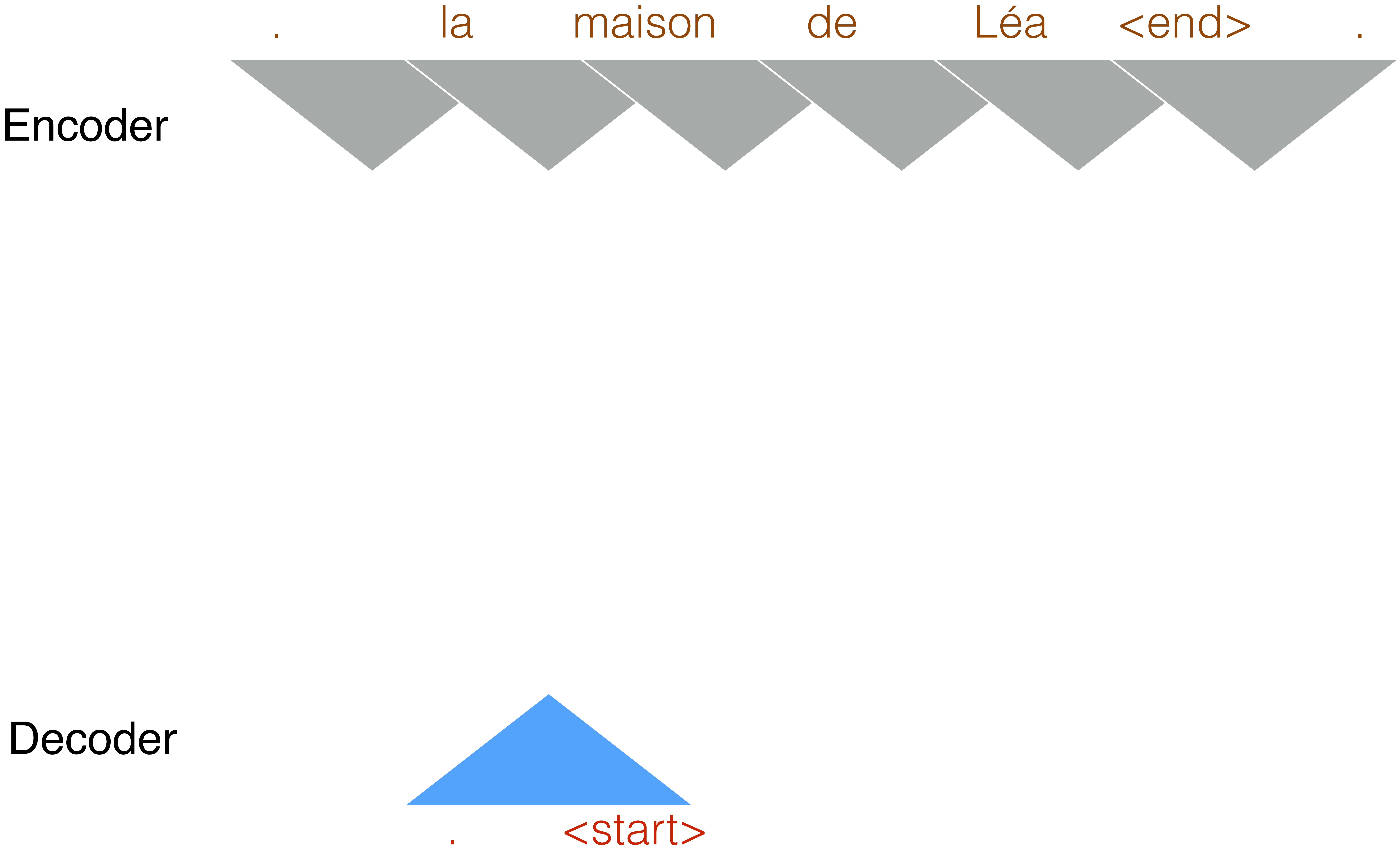


. la maison de Léa <end> .

Encoder

Decoder

. <start>

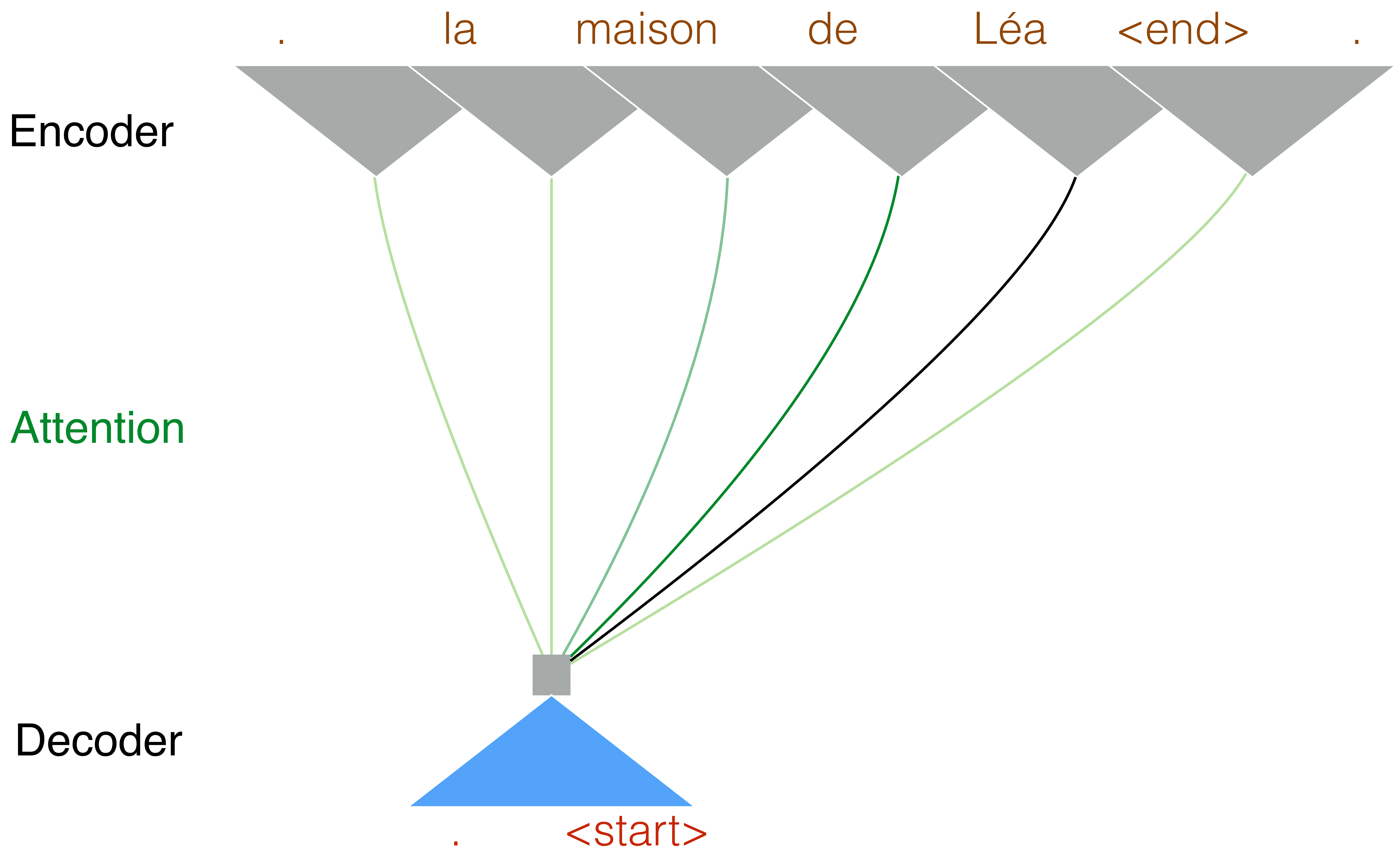


. la maison de Léa <end> .

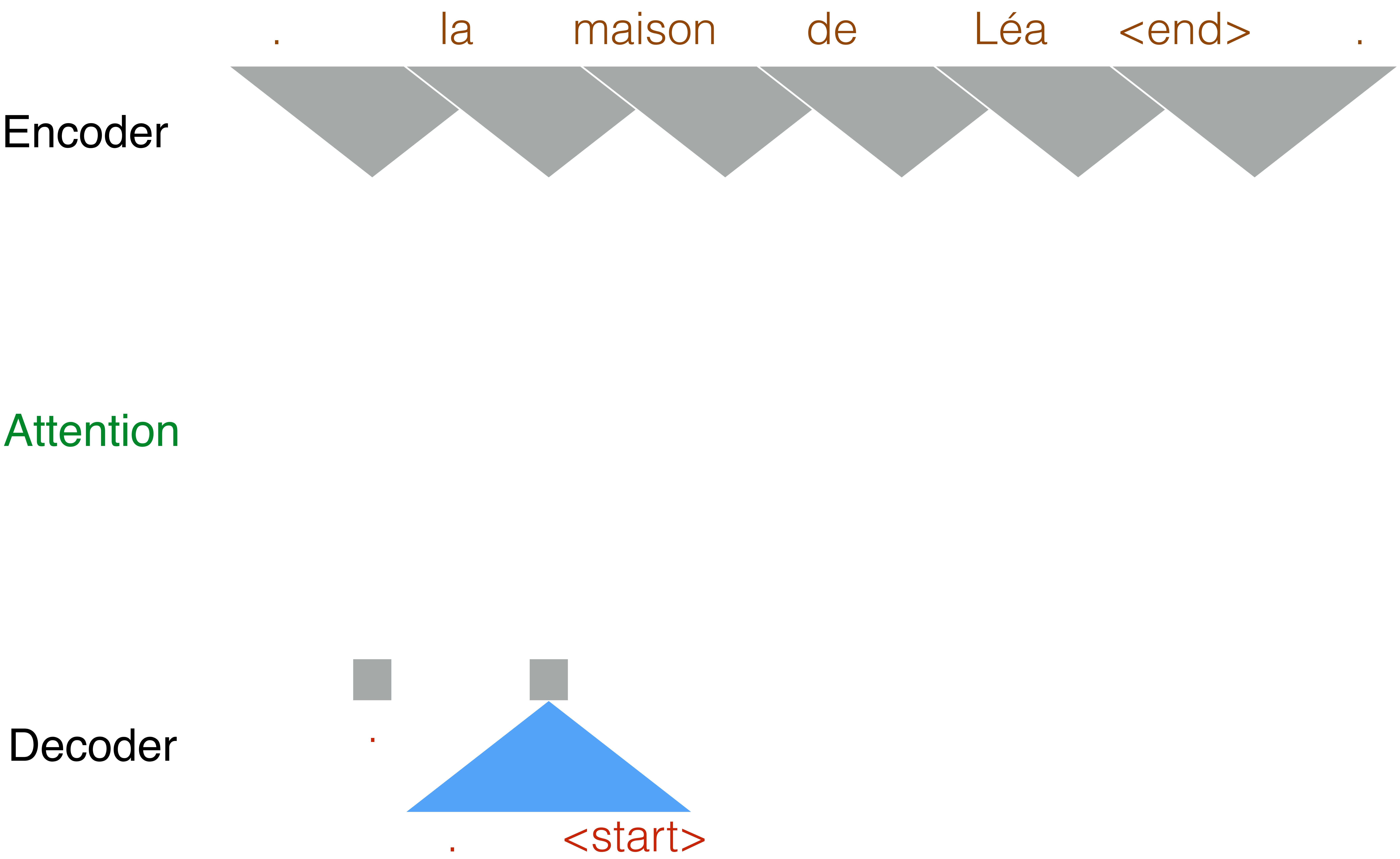
Encoder

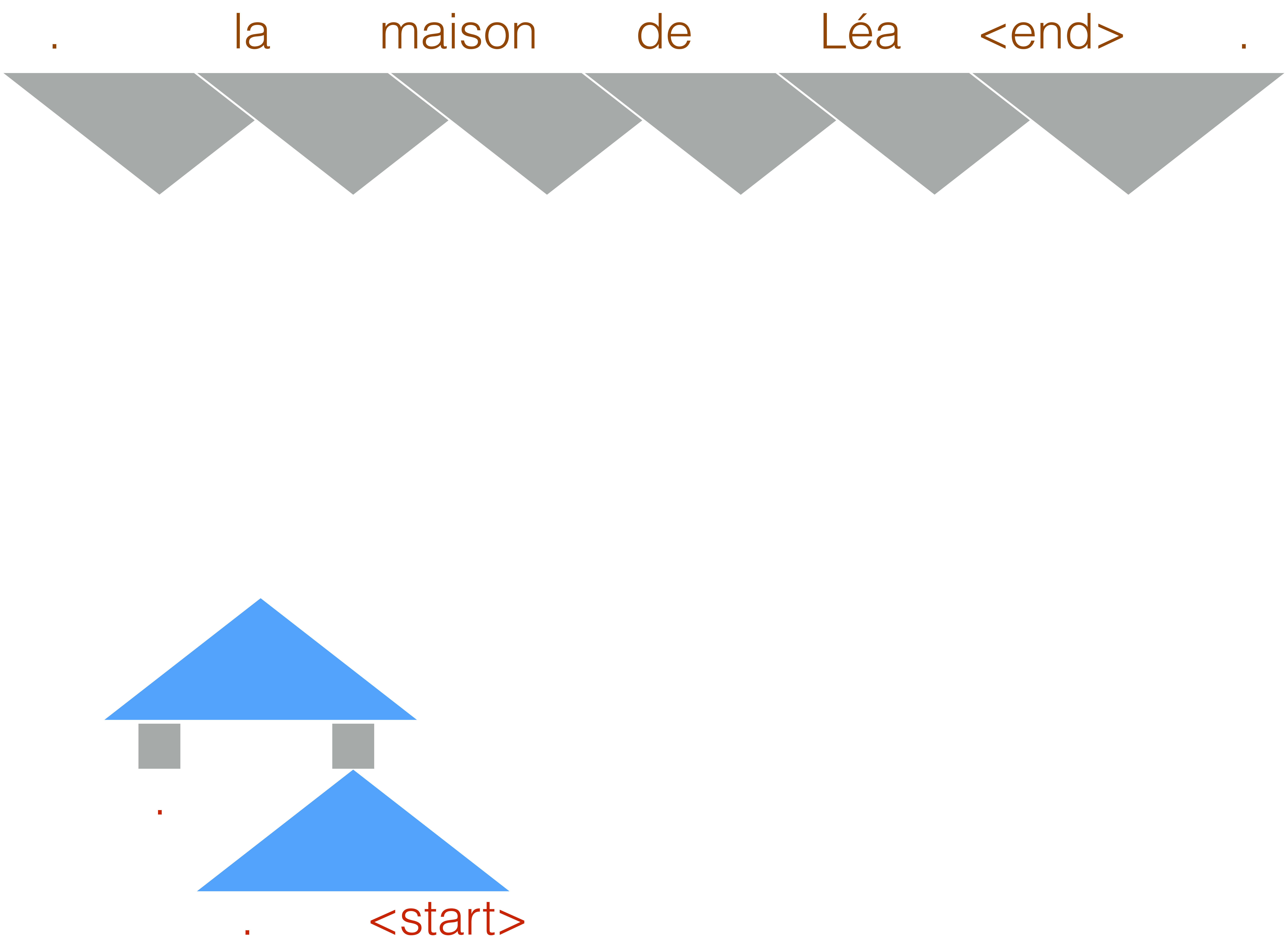
Decoder

. <start>



.
la
maison
de
Léa
<end>
.
Encoder
Attention
Decoder
.
<start>


.
la
maison
de
Léa
<end>
.
Encoder
Attention
Decoder
.
.
<start>



. la maison de Léa <end> .

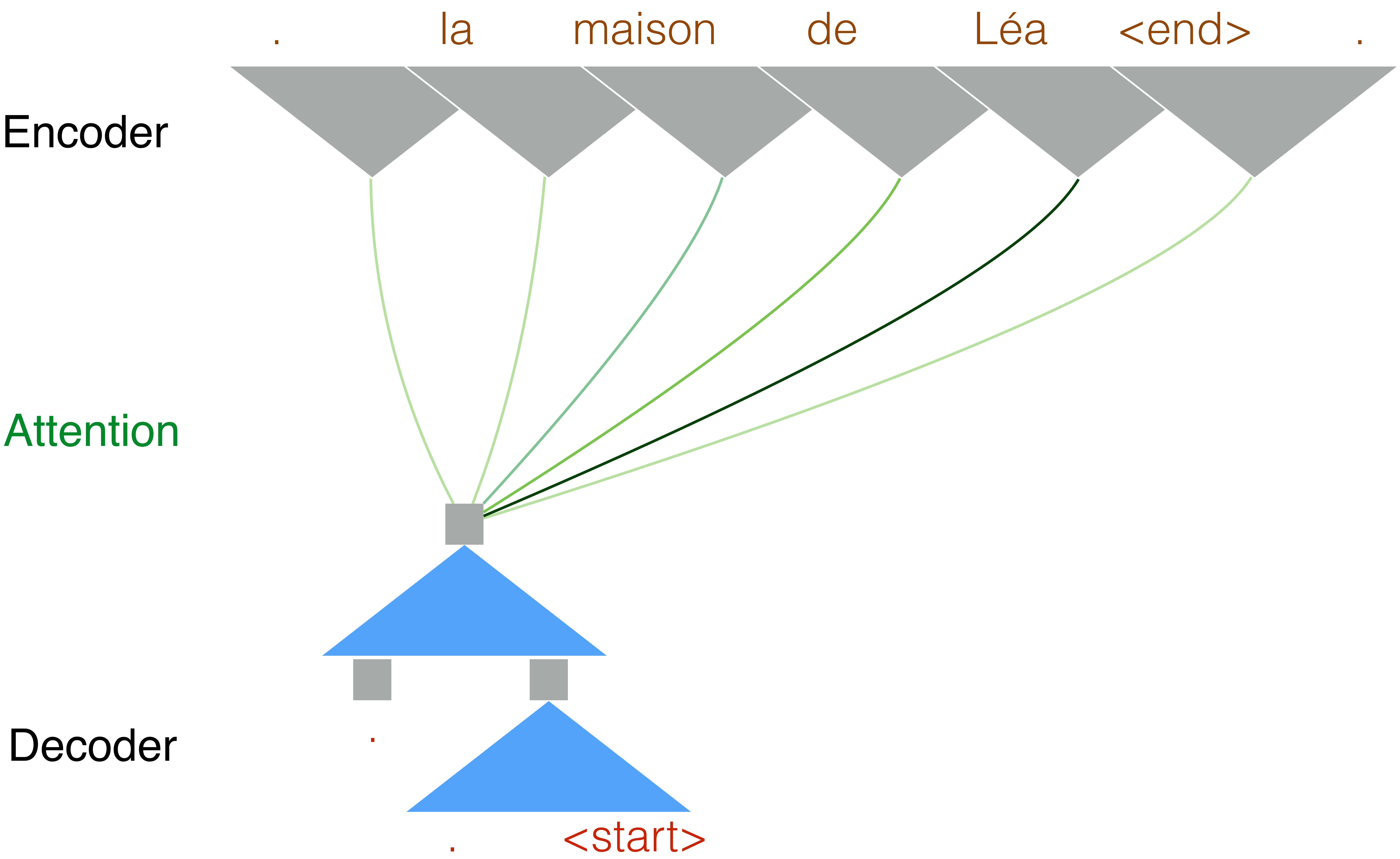
Encoder

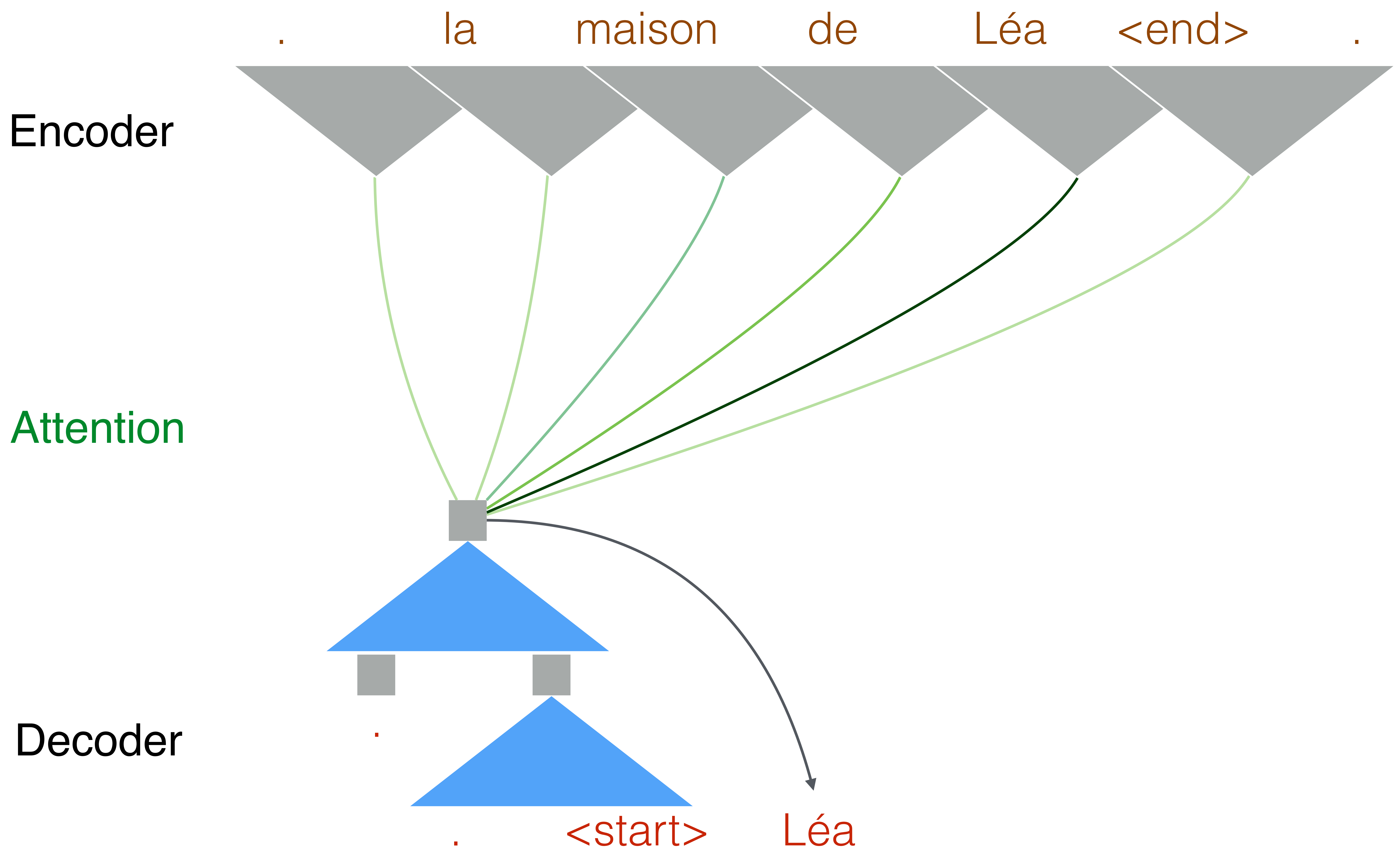
Attention

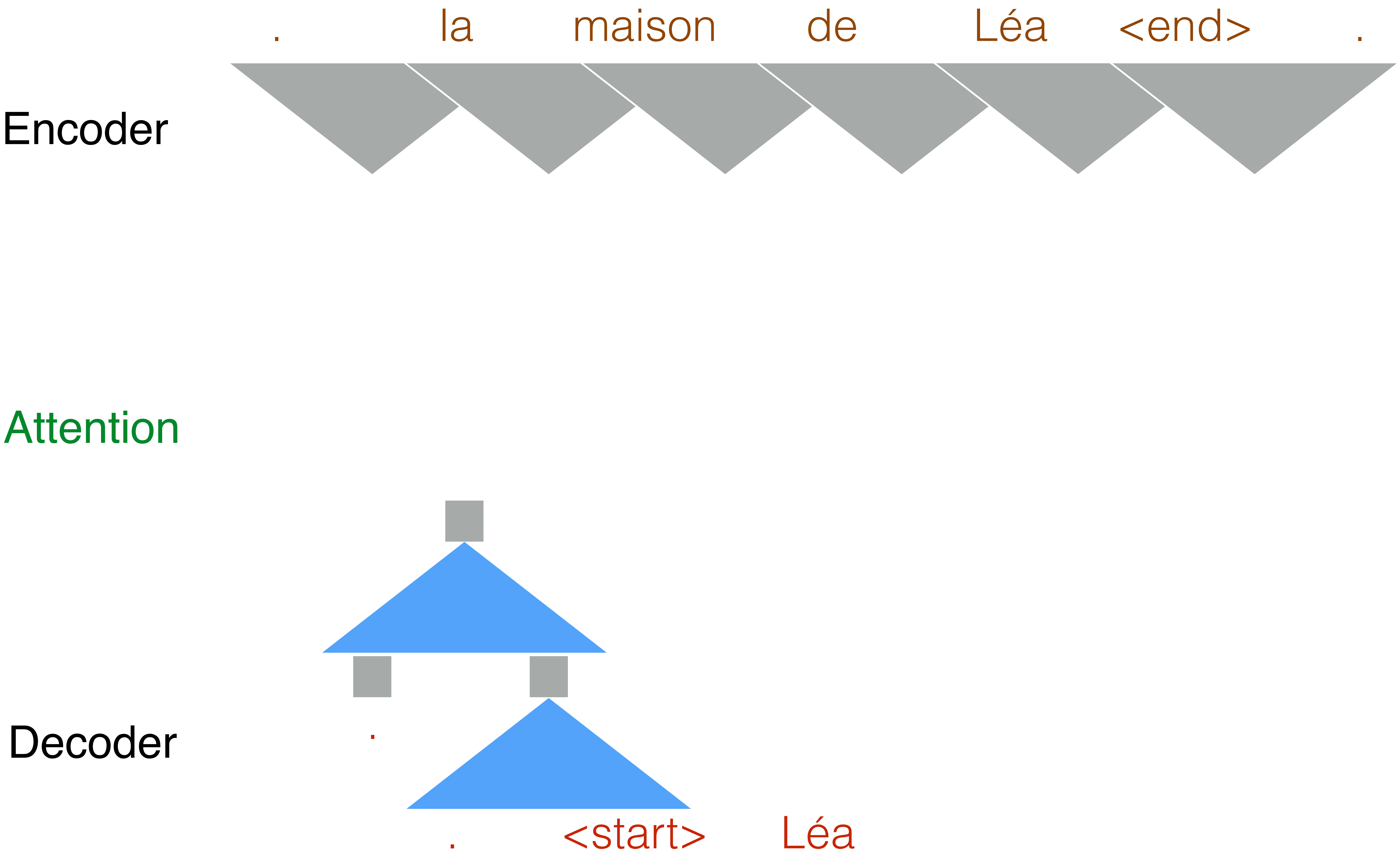
Decoder

.

. <start>


.
la
maison
de
Léa
<end>
.
Encoder
Attention
Decoder
.
.
<start>


. la maison de Léa <end> .
Encoder
Attention
Decoder
.
. <start> Léa



. la maison de Léa <end> .

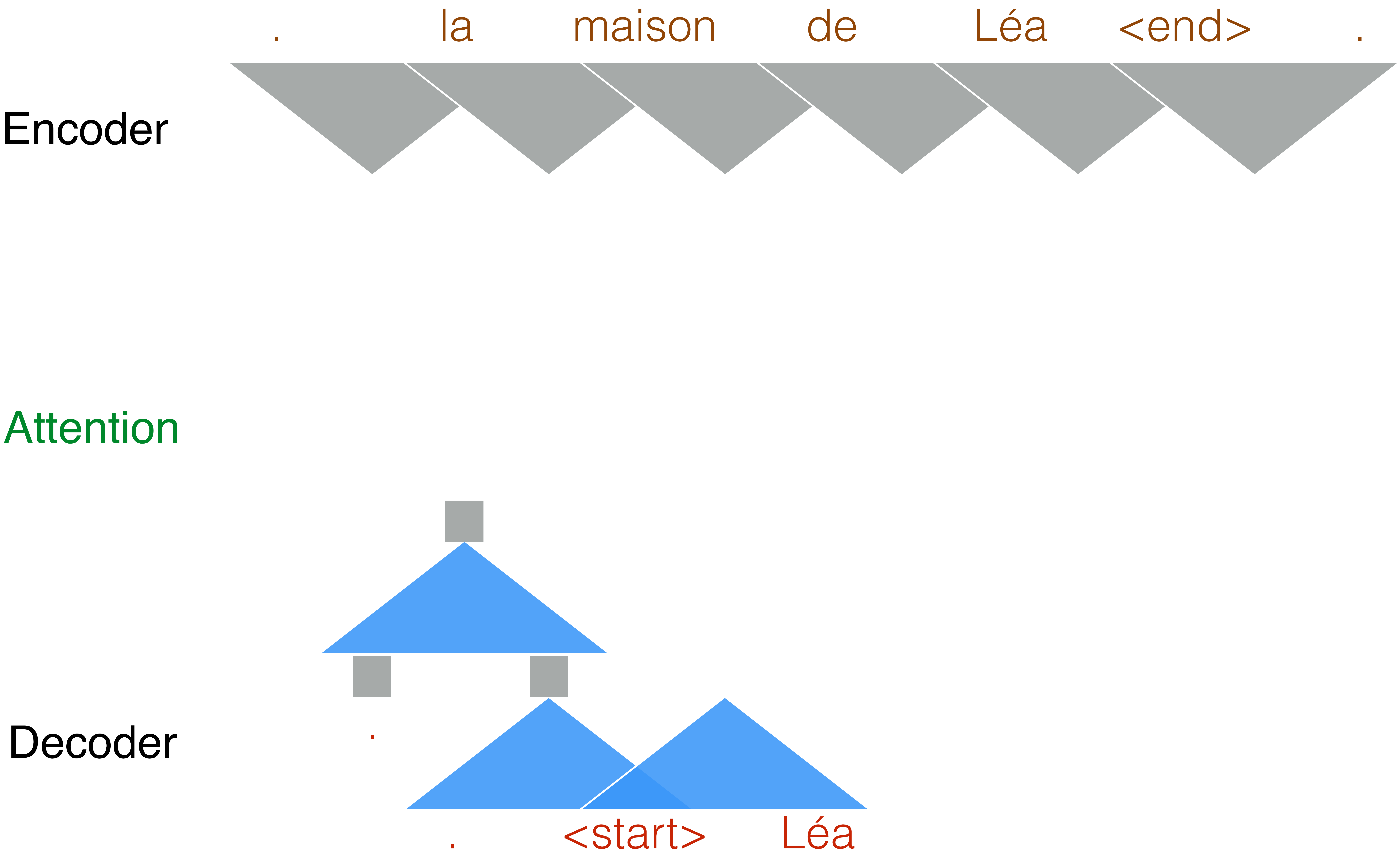
Encoder

Attention

Decoder

.

. <start> Léa



. la maison de Léa <end> .

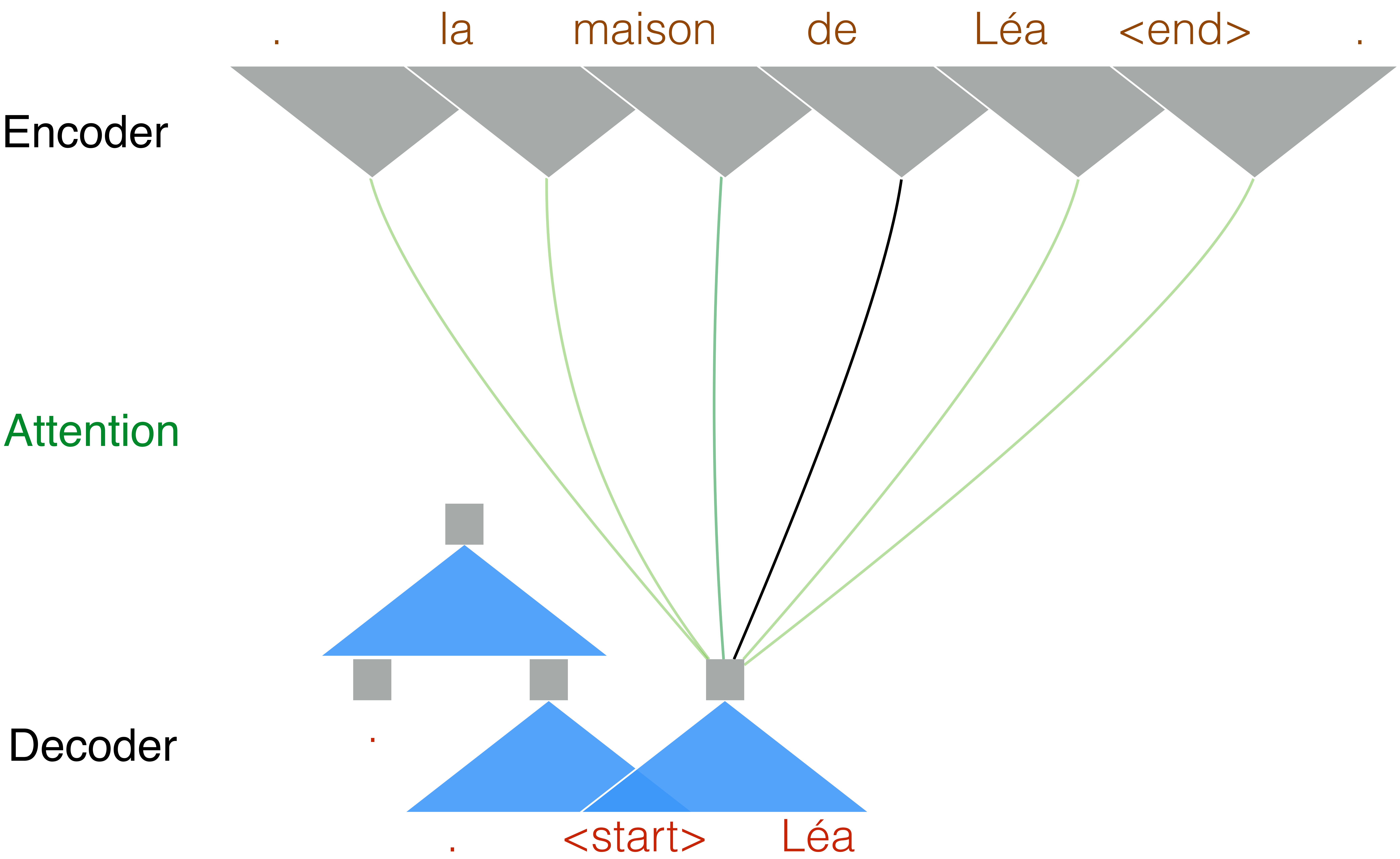
Encoder

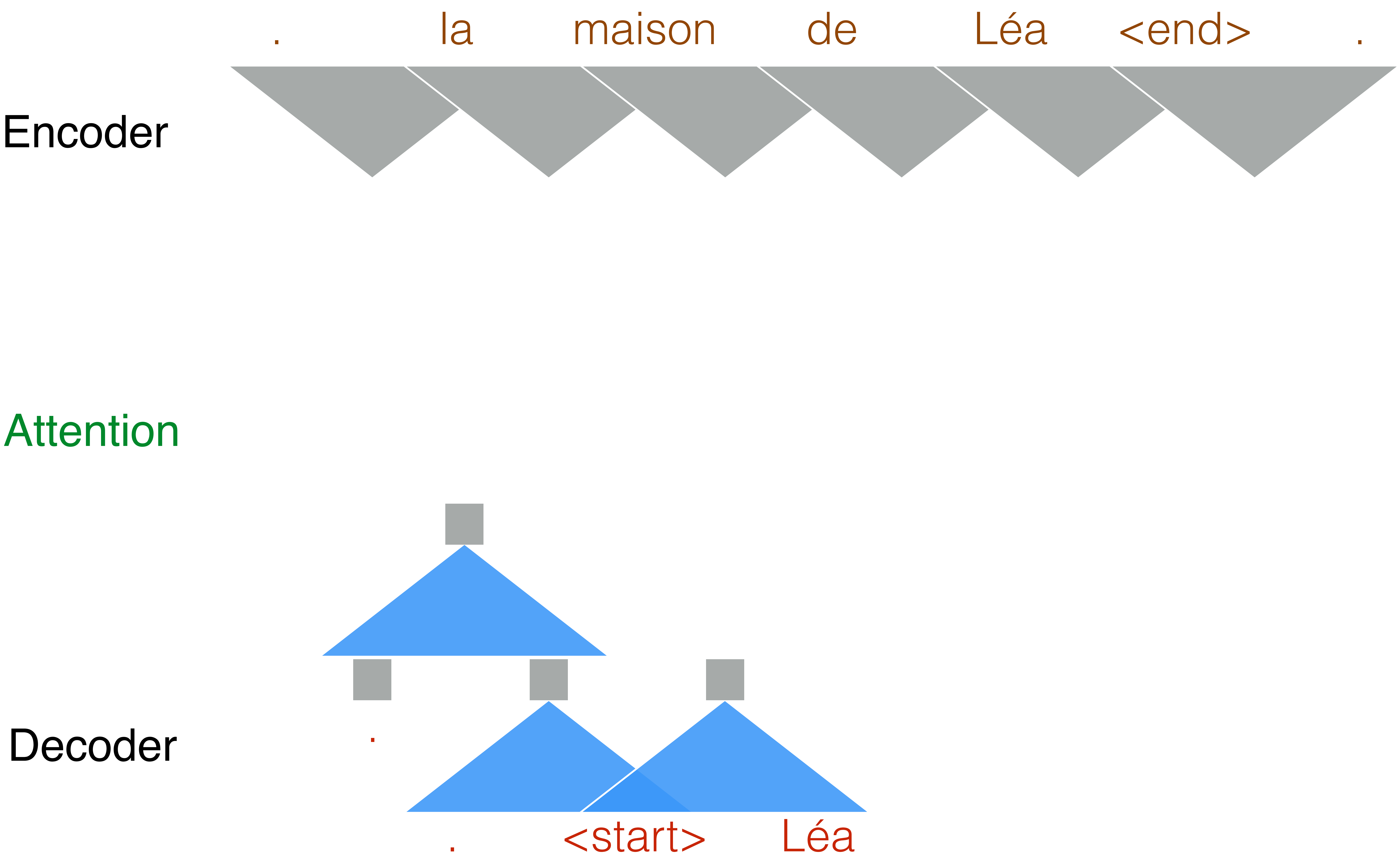
Attention

Decoder

.

. <start> Léa


.
la
maison
de
Léa
<end>
.
Encoder
Attention
Decoder
.
.
<start>
Léa



. la maison de Léa <end> .

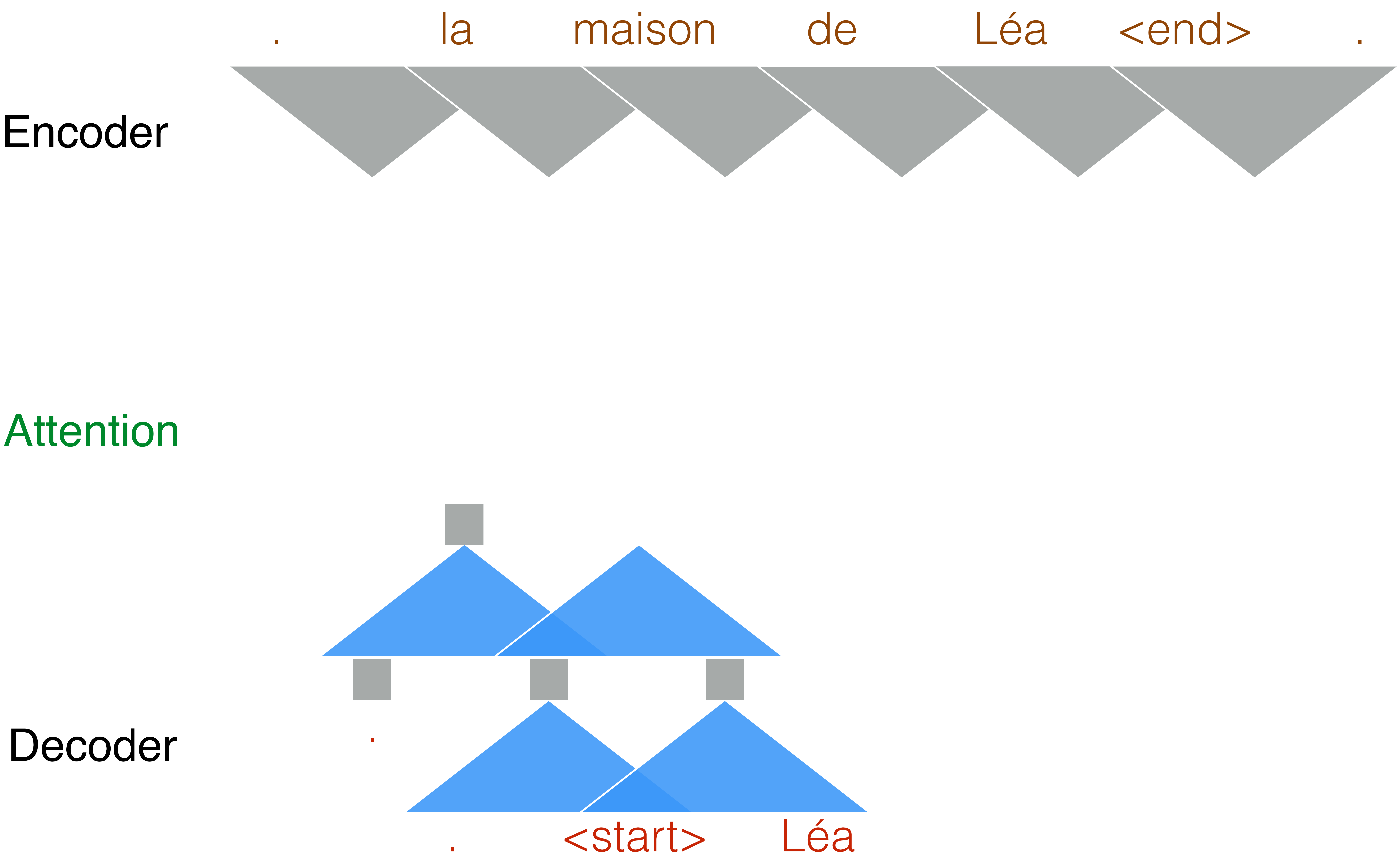
Encoder

Attention

Decoder

.

. <start> Léa



. la maison de Léa <end> .

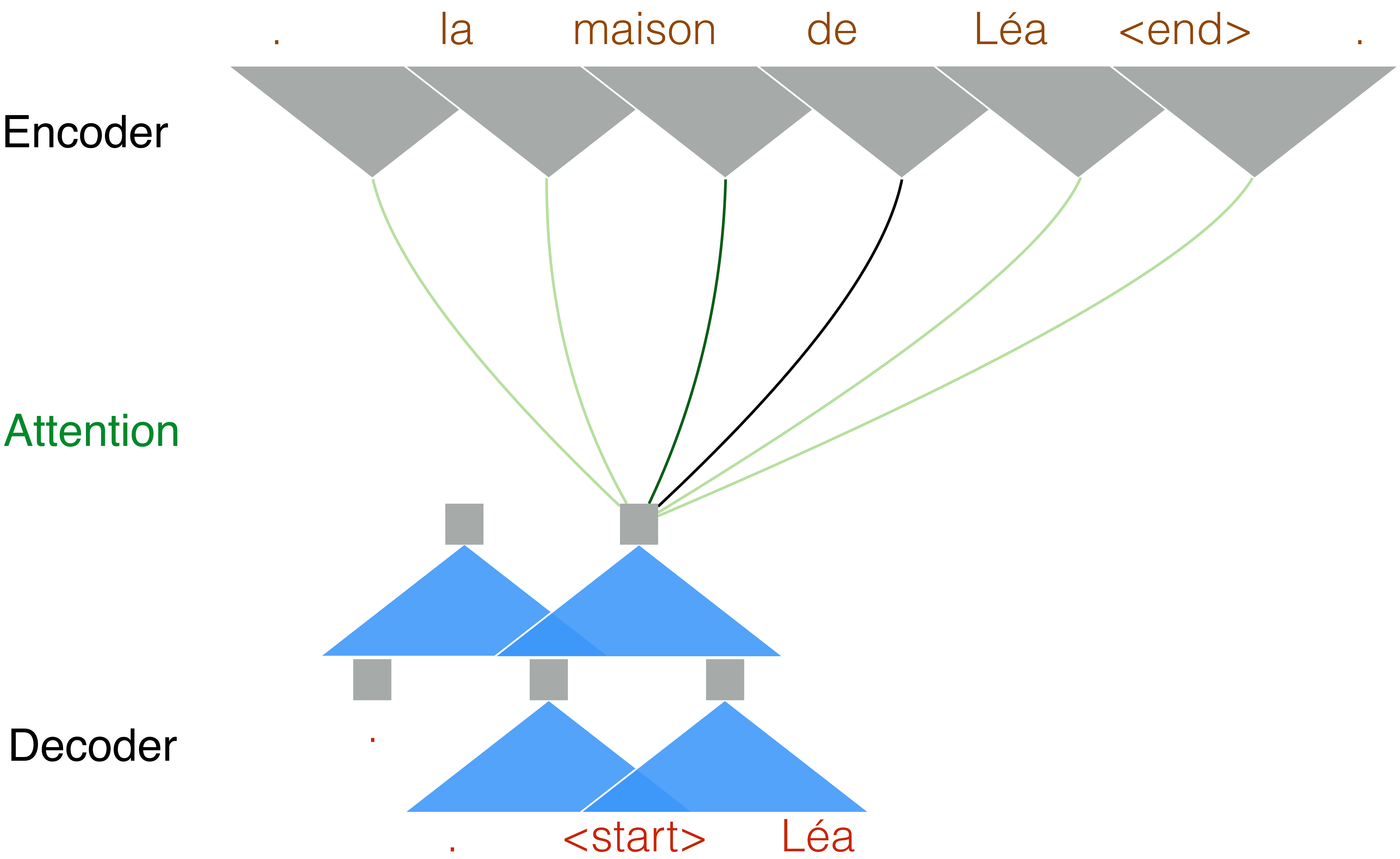
Encoder

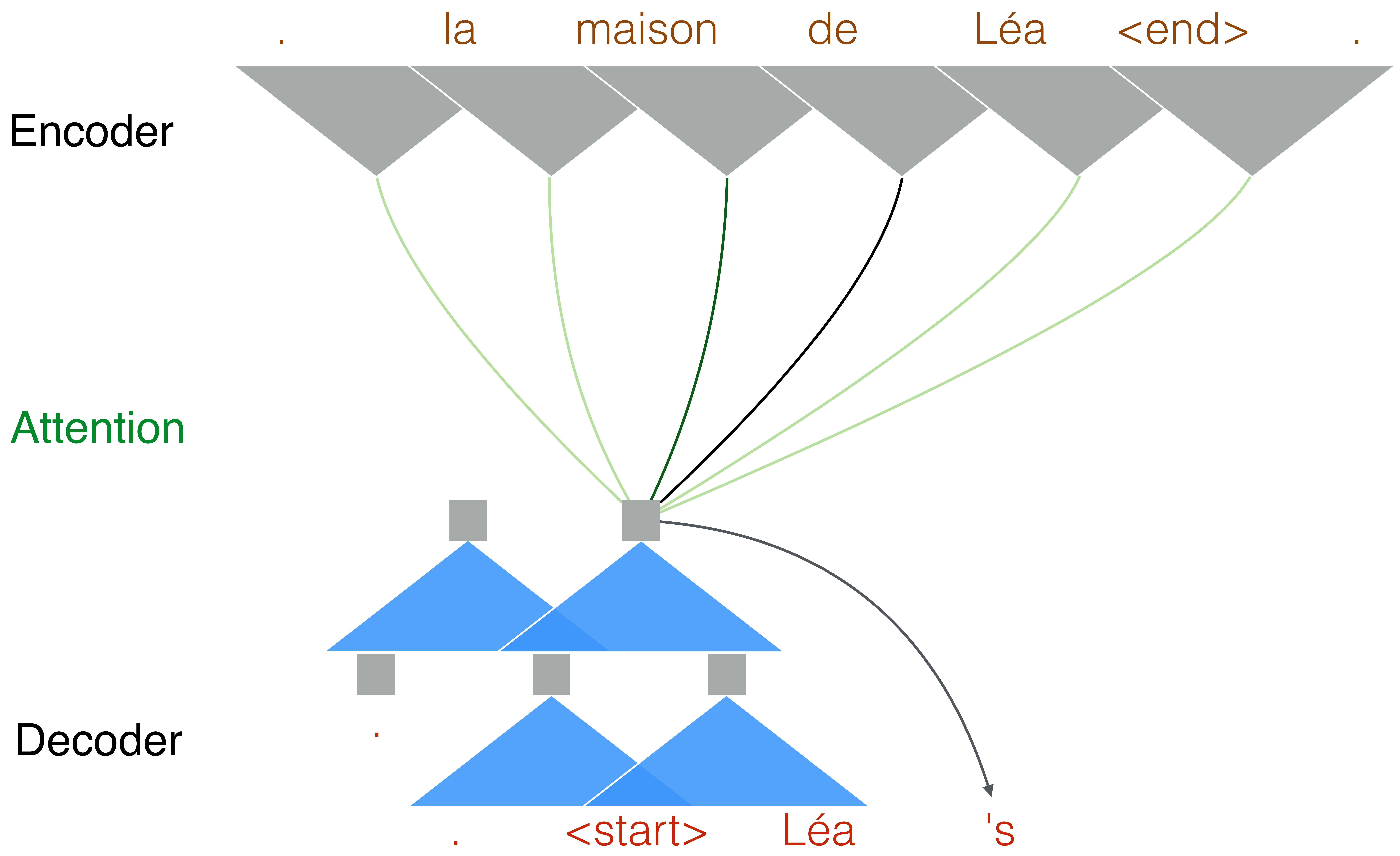
Attention

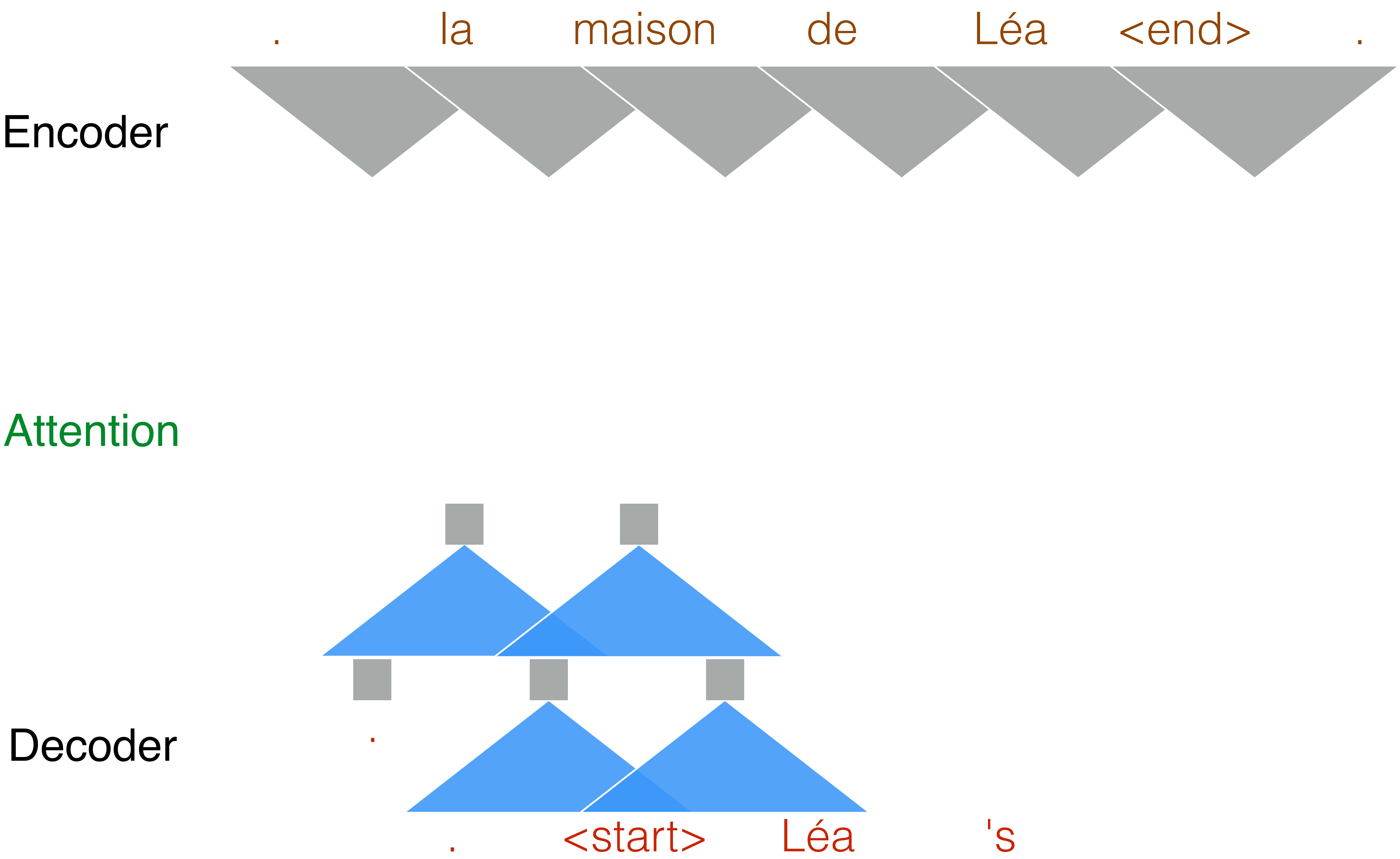
Decoder

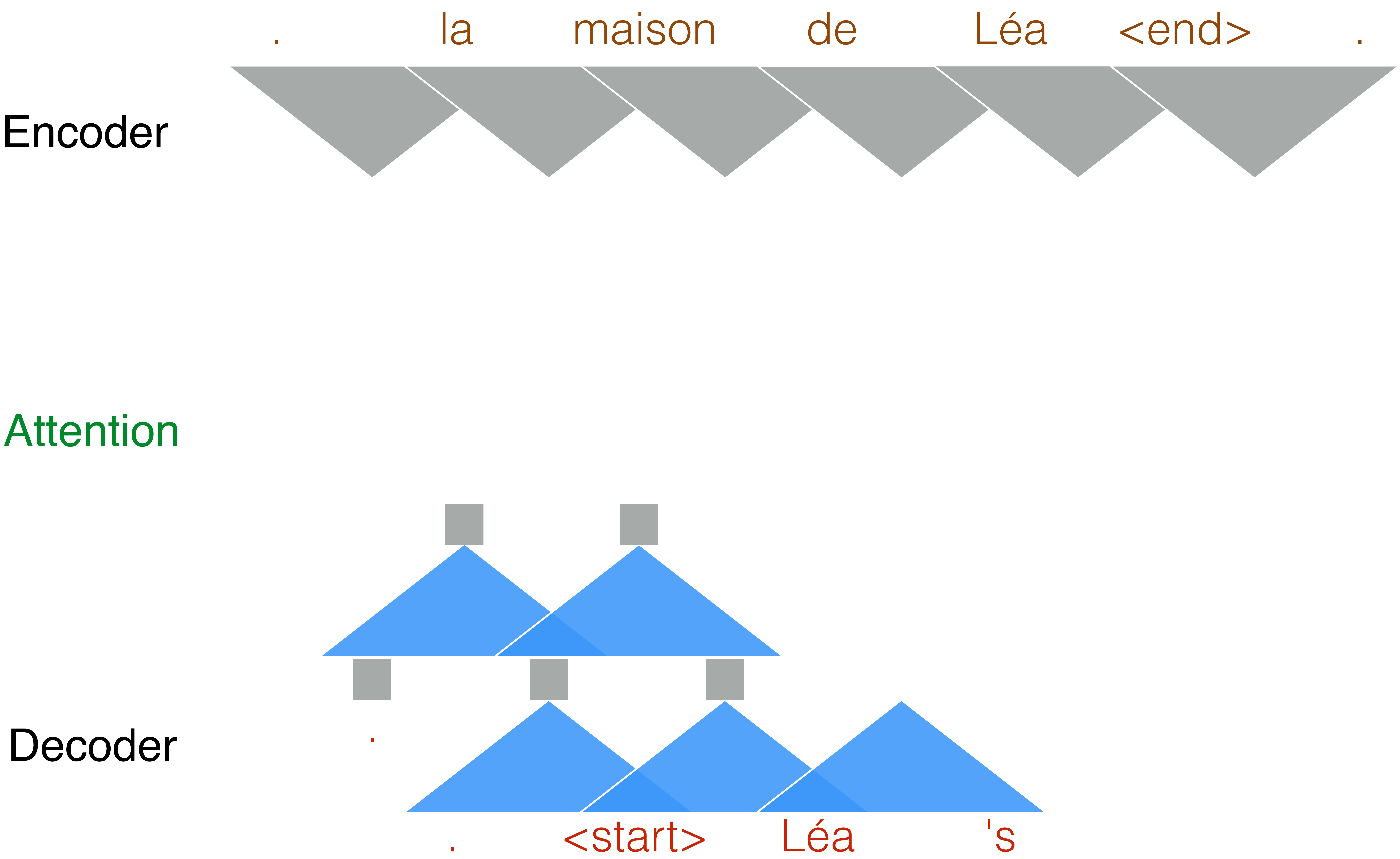
.

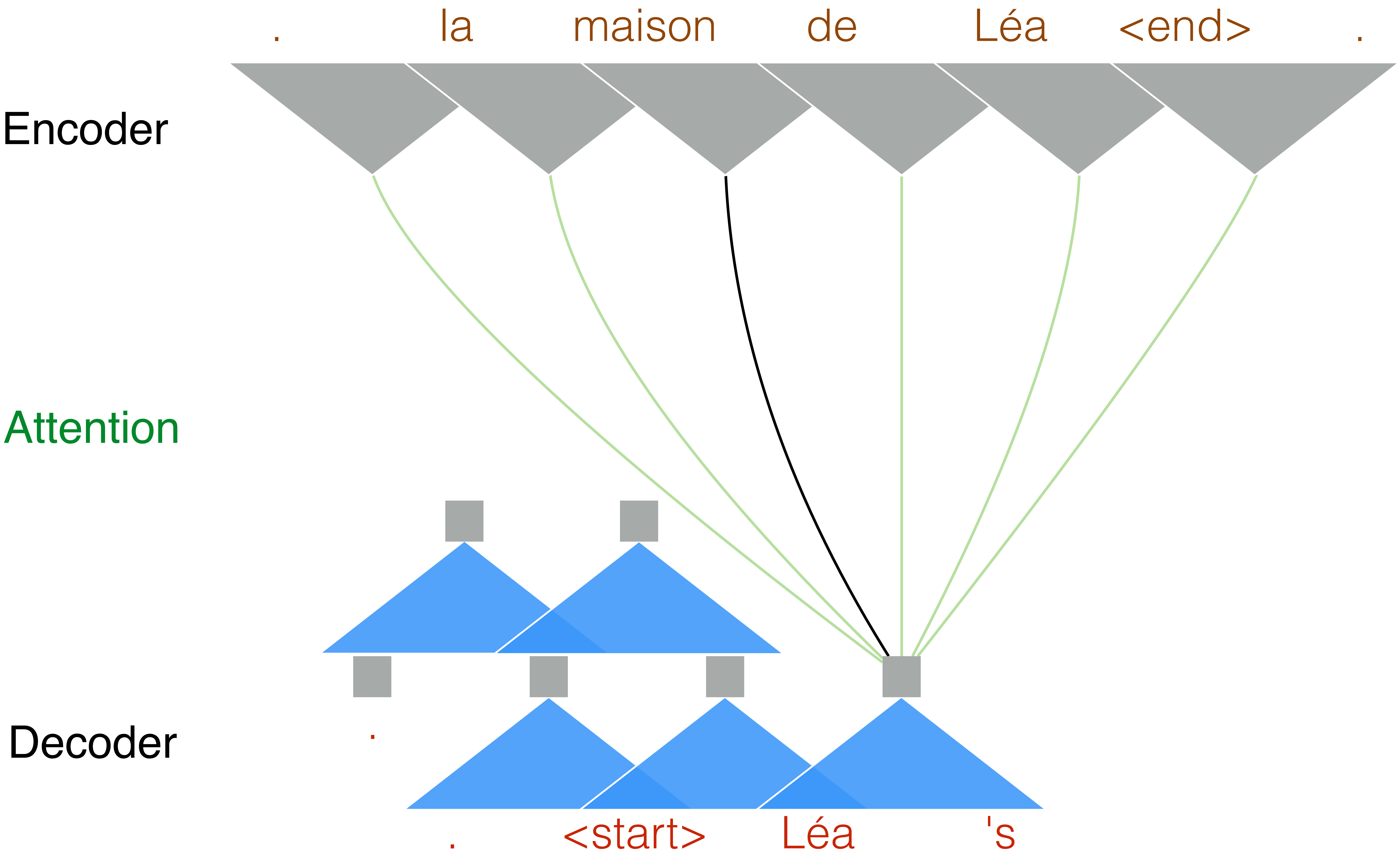
. <start> Léa


.
la
maison
de
Léa
<end>
.
Encoder
Attention
Decoder
.
.
<start>
Léa


.
la
maison
de
Léa
<end>
.
Encoder
Attention
Decoder
.
.
<start>
Léa
's


.
la
maison
de
Léa
<end>
.
Encoder
Attention
Decoder
.
.
<start>
Léa
's


. la maison de Léa <end> .
Encoder
Attention
Decoder
.
. <start> Léa 's



. la maison de Léa <end> .

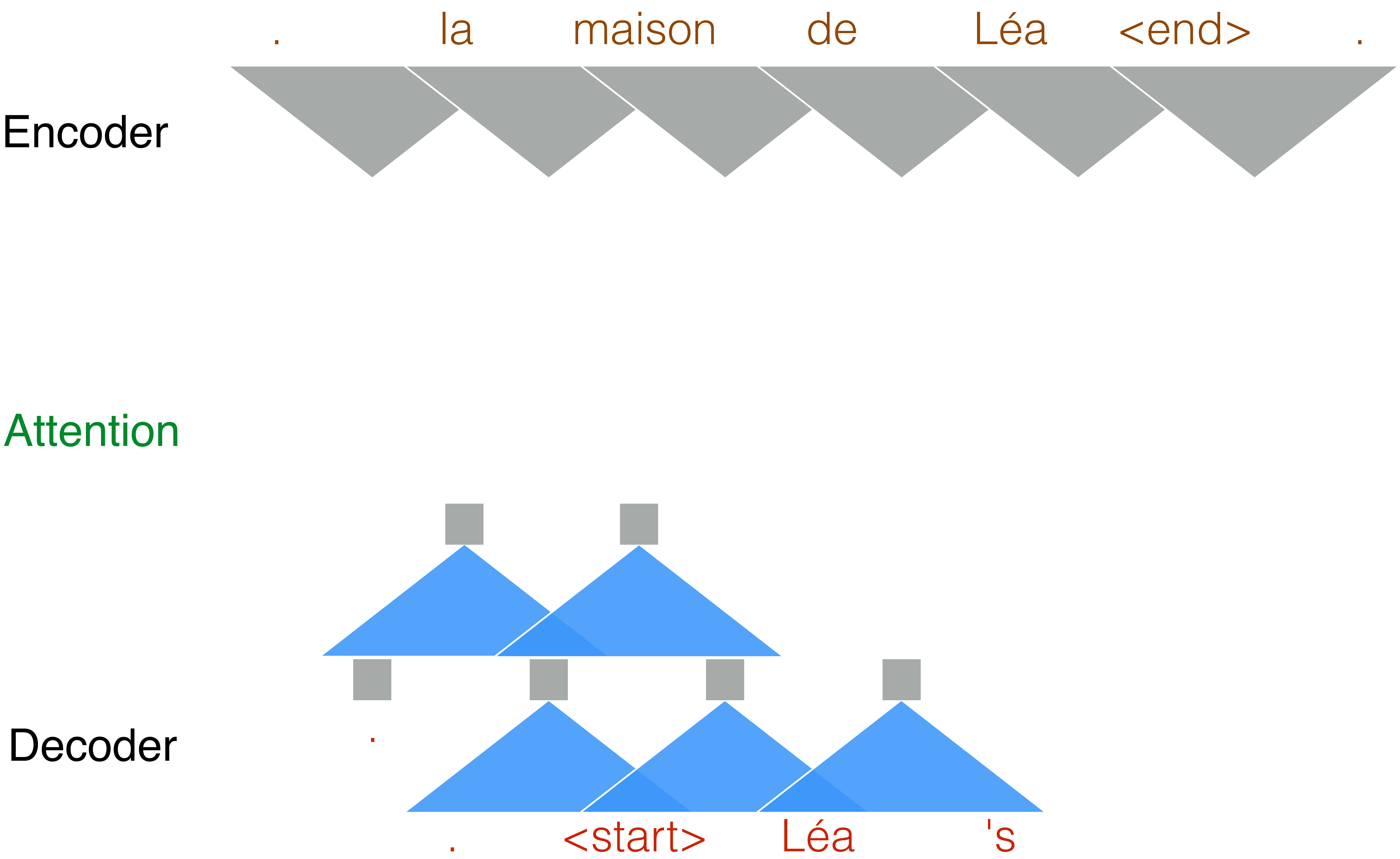
Encoder

Attention

Decoder

.

. <start> Léa 's



Encoder

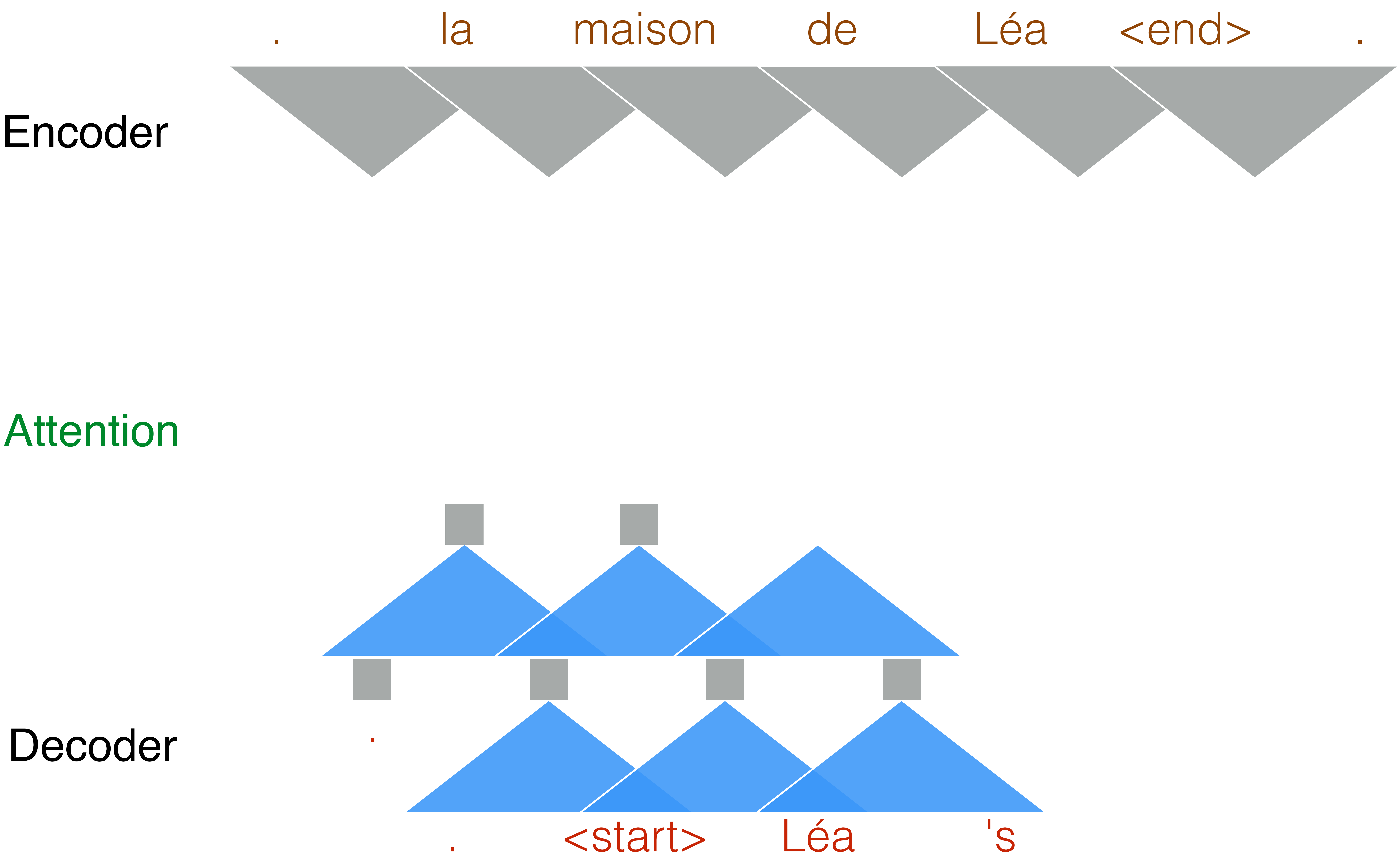
. la maison de Léa <end> .

Attention

Decoder

.

. <start> Léa 's



. la maison de Léa <end> .

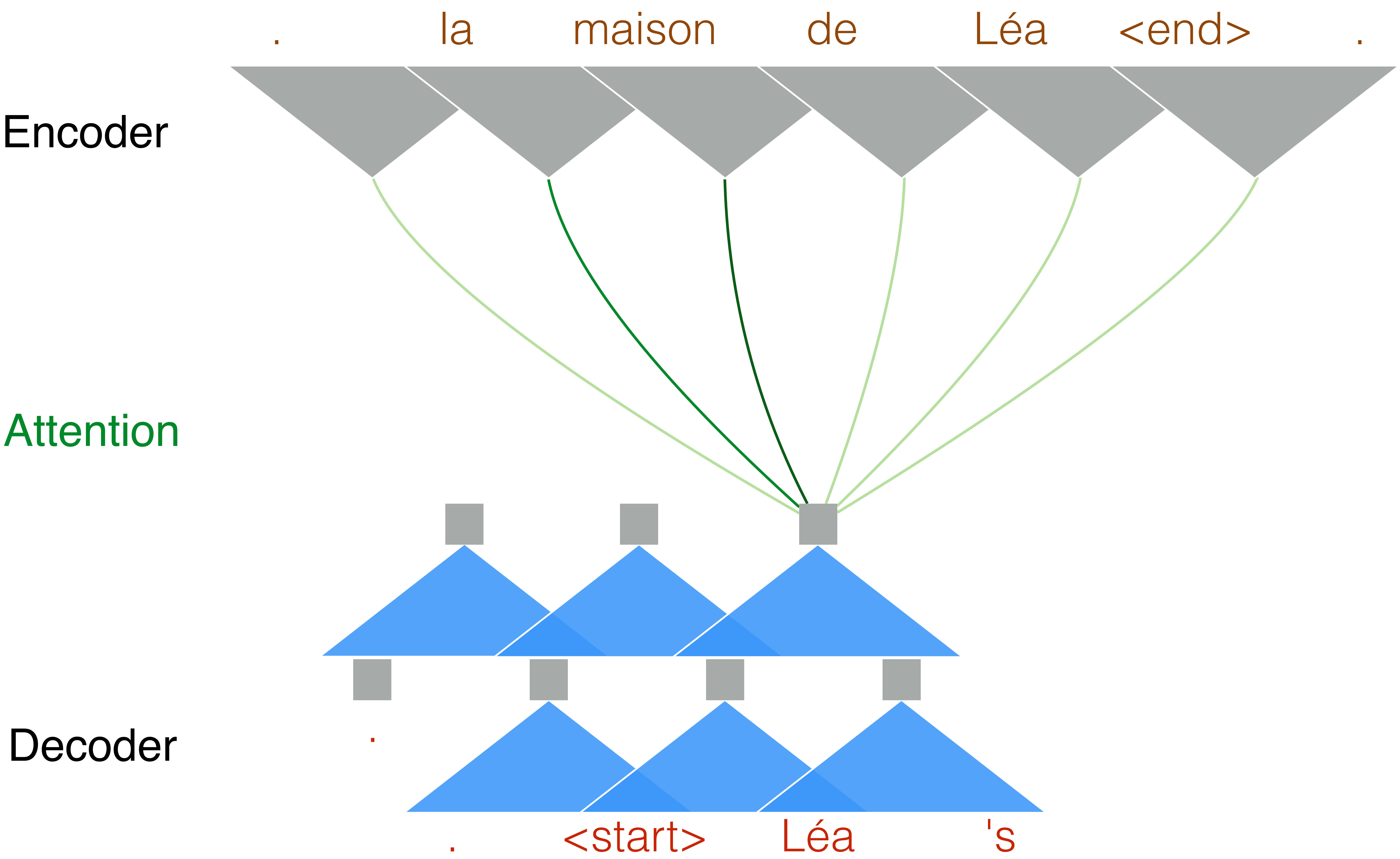
Encoder

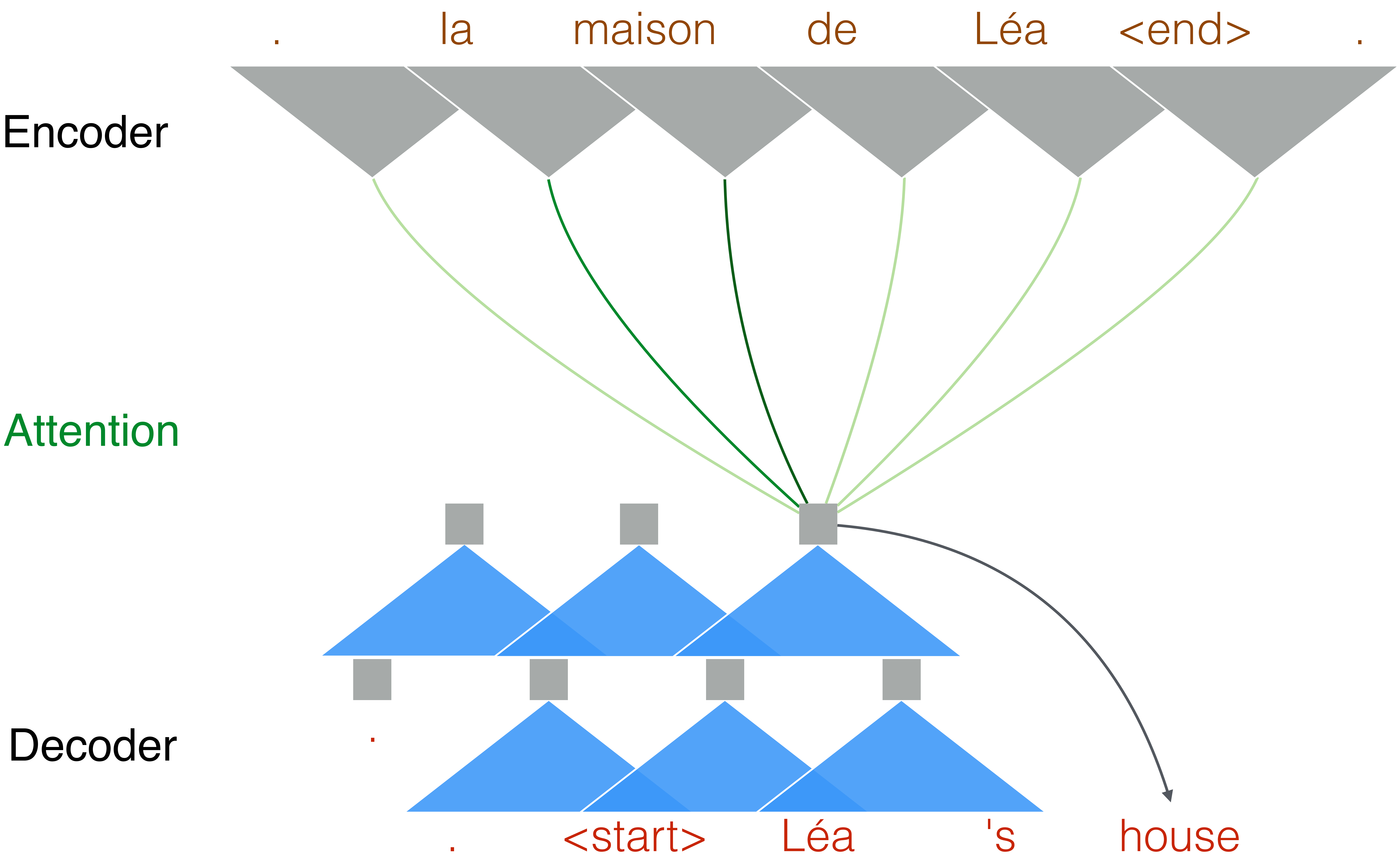
Attention

Decoder

.

. <start> Léa 's


.
la
maison
de
Léa
<end>
.
Encoder
Attention
Decoder
.
.
<start>
Léa
's


.
la
maison
de
Léa
<end>
.
Encoder
Attention
Decoder
.
.
<start>
Léa
's
house

WMT’14 English-German Translation

	Vocabulary	BLEU ↑
CNN ByteNet (Kalchbrenner et al., 2016)	Characters	23.75
RNN GNMT (Wu et al., 2016)	Word 80k	23.12
RNN GNMT (Wu et al., 2016)	Word pieces	24.61
ConvS2S	BPE 40k	25.16

ConvS2S: 15 layers in encoder/decoder (10x512 units, 3x768 units, 2x2048)
Maximum context size: 27 words

WMT’14 English-German Translation

	Vocabulary	BLEU ↑
CNN ByteNet (Kalchbrenner et al., 2016)	Characters	23.75
RNN GNMT (Wu et al., 2016)	Word 80k	23.12
RNN GNMT (Wu et al., 2016)	Word pieces	24.61
ConvS2S	BPE 40k	25.16
Transformer (Vaswani et al., 2017)	Word pieces	28.4

More work on non-RNN models!

ConvS2S: 15 layers in encoder/decoder (10x512 units, 3x768 units, 2x2048)
Maximum context size: 27 words

WMT’14 English-French Translation

	Vocabulary	BLEU ↑
RNN GNMT (Wu et al., 2016)	Word 80k	37.90
RNN GNMT (Wu et al., 2016)	Word pieces	38.95
RNN GNMT + RL (Wu et al., 2016)	Word pieces	39.92

WMT’14 English-French Translation

	Vocabulary	BLEU ↑
RNN GNMT (Wu et al., 2016)	Word 80k	37.90
RNN GNMT (Wu et al., 2016)	Word pieces	38.95
RNN GNMT + RL (Wu et al., 2016)	Word pieces	39.92
ConvS2S	BPE 40k	40.51
Transformer (Vaswani et al., 2017)	Word pieces	41.0

ConvS2S: 15 layers in encoder/decoder (5x512 units, 4x768 units, 3x2048, 2x4096)

Inference Speed on WMT’14 En-Fr

	Hardware	BLEU	Time (s)
RNN GNMT (Wu et al., 2016)	GPU (K80)	31.20	3028
RNN GNMT (Wu et al., 2016)	CPU (88 cores)	31.20	1322
RNN GNMT (Wu et al., 2016)	TPU	31.21	384

ntst1213 (6003 sentences)

Inference Speed on WMT’14 En-Fr

	Hardware	BLEU	Time (s)
RNN GNMT (Wu et al., 2016)	GPU (K80)	31.20	3028
RNN GNMT (Wu et al., 2016)	CPU (88 cores)	31.20	1322
RNN GNMT (Wu et al., 2016)	TPU	31.21	384
ConvS2S, beam=5	GPU (K40)	34.10	587
ConvS2S, beam=1	GPU (K40)	33.45	327

ntst1213 (6003 sentences)

Inference Speed on WMT’14 En-Fr

	Hardware	BLEU	Time (s)
RNN GNMT (Wu et al., 2016)	GPU (K80)	31.20	3028
RNN GNMT (Wu et al., 2016)	CPU (88 cores)	31.20	1322
RNN GNMT (Wu et al., 2016)	TPU	31.21	384
ConvS2S, beam=5	GPU (K40)	34.10	587
ConvS2S, beam=1	GPU (K40)	33.45	327
ConvS2S, beam=1	GPU (GTX-1080ti)	33.45	142
ConvS2S, beam=1	CPU (48 cores)	33.45	142

ntst1213 (6003 sentences)

Summary

- Alternative architecture for sequence to sequence learning
- Higher accuracy than models of similar size, despite fixed size context
- Faster generation (9x faster on lesser hardware)

Code & pre-trained models:
+ lua/torch: http://github.com/facebookresearch/fairseq
+ PyTorch: http://github.com/facebookresearch/fairseq-py

Exposure bias/Loss Mismatch: Training at the Sequence Level

Classical Structured Prediction Losses for Sequence to Sequence Learning
Sergey Edunov*, Myle Ott*, Michael Auli, David Grangier, Marc'Aurelio Ranzato
NAACL 2018
https://arxiv.org/abs/1711.04956

slide credit: Sergey Edunov, Marc'Aurelio Ranzato

Problems

- **Exposure bias:** training and testing are inconsistent
  At training, the model has never observed its own predictions as input.
- Training criterion != Evaluation criterion
- Evaluation criterion is not differentiable

Selection of Recent Literature

- Reinforcement Learning-inspired methods
  - MIXER (Ranzato et al., ICLR 2016)
  - Actor-Critic (Bahdanau et al., ICLR 2017)
- Using beam search at training time:
  - Beam search optimization (Wiseman et al. ACL 2016)
  - Distillation based (Kim et al., EMNLP 2016)

Questions

1) How do classical structure prediction losses compare to recent methods?
2) Classical losses were often applied to log-linear models - how well do they work for neural nets?

Bottou et al. “Global training of document processing systems with graph transformer networks” CVPR 1997

Collins “Discriminative training methods for HMMs” EMNLP 2002

Taskar et al. “Max-margin Markov networks” NIPS 2003

Tsochantaridis et al. “Large margin methods for structured and interdependent output variables” JMLR 2005

Och “Minimum error rate training in statistical machine translation” ACL 2003

Smith and Eisner “Minimum risk annealing for training log-linear models” ACL 2006

Gimpel and Smith “Softmax-margin CRFs: training log-linear models with cost functions” ACL 2010

Notation

$\mathbf{x} = (x_1, \ldots, x_m)$ input sentence

Notation

$\mathbf{x}$	input sentence
$\mathbf{t}$	target sentence

Notation

$\mathbf{x}$	input sentence
$\mathbf{t}$	target sentence
$\mathbf{u}$	hypothesis generated by the model

Notation

$\mathbf{x}$	input sentence
$\mathbf{t}$	target sentence
$\mathbf{u}$	hypothesis generated by the model
$\mathbf{u}^* = \arg\min_{\mathbf{u} \in \mathcal{U}(\mathbf{x})} \operatorname{cost}(\mathbf{u}, \mathbf{t})$	oracle hypothesis

Notation

$\mathbf{x}$	input sentence
$\mathbf{t}$	target sentence
$\mathbf{u}$	hypothesis generated by the model
$\mathbf{u}^*$	oracle hypothesis
$\hat{\mathbf{u}}$	most likely hypothesis

Baseline: Token Level NLL

$$\mathcal{L}_{\text{TokNLL}} = -\sum_{i=1}^{n} \log p(t_i|t_1, \ldots, t_{i-1}, \mathbf{x})$$

for one particular training example.

'Locally' normalized over vocabulary.

Sequence Level NLL

$$\mathcal{L}_{\text{SeqNLL}} = -\log p(\mathbf{u}^*|\mathbf{x}) + \log \sum_{\mathbf{u}\in\mathcal{U}(\mathbf{x})} p(\mathbf{u}|\mathbf{x})$$

The sequence log-probability is simply the sum of the token-level log-probabilities.

'Globally' normalized over set of hypothesis $\mathcal{U}(\mathbf{x})$

Sequence Level NLL

Source:
Wir müssen unsere Einwanderungspolitik in Ordnung bringen.

Target:
We have to fix our immigration policy.

Beam:

$\mathcal{U}(\mathbf{x})$

BLEU	Model score	
75.0	-0.23	We need to fix our immigration policy.
100.0	-0.30	We have to fix our immigration policy.
36.9	-0.36	We need to fix our policy policy.
66.1	-0.42	We have to fix our policy policy.
66.1	-0.44	We've got to fix our immigration policy.

Sequence Level NLL

Source:
Wir müssen unsere Einwanderungspolitik in Ordnung bringen.

Target:
We have to fix our immigration policy.

Beam:

$$\mathcal{U}(\mathbf{x}) \left\{ \right.$$

BLEU	Model score		
75.0	-0.23	↓	We need to fix our immigration policy.
100.0	-0.30	↑	We have to fix our immigration policy.
36.9	-0.36	↓	We need to fix our policy policy.
66.1	-0.42	↓	We have to fix our policy policy.
66.1	-0.44	↓	We've got to fix our immigration policy.

Observations

- Important to use **oracle hypothesis** as surrogate target.
  Otherwise, the model learns to assign very **bad scores** to its hypotheses but is not trained to reach the target.

- Evaluation metric only used for **oracle selection** of target.

- Several ways to generate $\mathcal{U}(\mathbf{x})$

Expected Risk

$$\mathcal{L}_{\text{Risk}} = \sum_{\mathbf{u} \in \mathcal{U}(\mathbf{x})} \text{cost}(\mathbf{t}, \mathbf{u}) \frac{p(\mathbf{u}|\mathbf{x})}{\sum_{\mathbf{u}' \in \mathcal{U}(\mathbf{x})} p(\mathbf{u}'|\mathbf{x})}$$

- The cost is the evaluation metric; e.g.: 100-BLEU.

- REINFORCE is a special case of this (a single sample Monte Carlo estimate of the expectation over the *whole* hypothesis space).

Expected Risk

Source:
Wir müssen unsere Einwanderungspolitik in Ordnung bringen.

Target
We have to fix our immigration policy.

Beam:

$\mathcal{U}(\mathbf{x})$

BLEU	Model score		
75.0	-0.23	↑	We need to fix our immigration policy.
100.0	-0.30	↑	We have to fix our immigration policy.
36.9	-0.36	↓	We need to fix our policy policy.
66.1	-0.42	↓	We have to fix our policy policy.
66.1	-0.44	↓	We've got to fix our immigration policy.

(expected BLEU=69)

Check out the paper for more examples of sequence level training losses!

Practical Tips

- Start from a model pre-trained at the token level. Training with search is excruciatingly slow…
- Even better if pre-trained model had label smoothing.
- Accuracy vs speed trade-off: offline/online generation of hypotheses.
- Mix token level NLL loss with sequence level loss to improve robustness.

Results on IWSLT'14 De-En

	TEST
TokNLL (Wiseman et al. 2016)	24.0
BSO (Wiseman et al. 2016)	26.4
Actor-Critic (Bahdanau et al. 2016)	28.5
Phrase-based NMT (Huang et al. 2017)	29.2

Results on IWSLT'14 De-En

	TEST
TokNLL (Wiseman et al. 2016)	24.0
BSO (Wiseman et al. 2016)	26.4
Actor-Critic (Bahdanau et al. 2016)	28.5
Phrase-based NMT (Huang et al. 2017)	29.2
our TokNLL	31.8
SeqNLL	32.7
Risk	**32.8**
Max-Margin	32.6

Fair Comparison to BSO

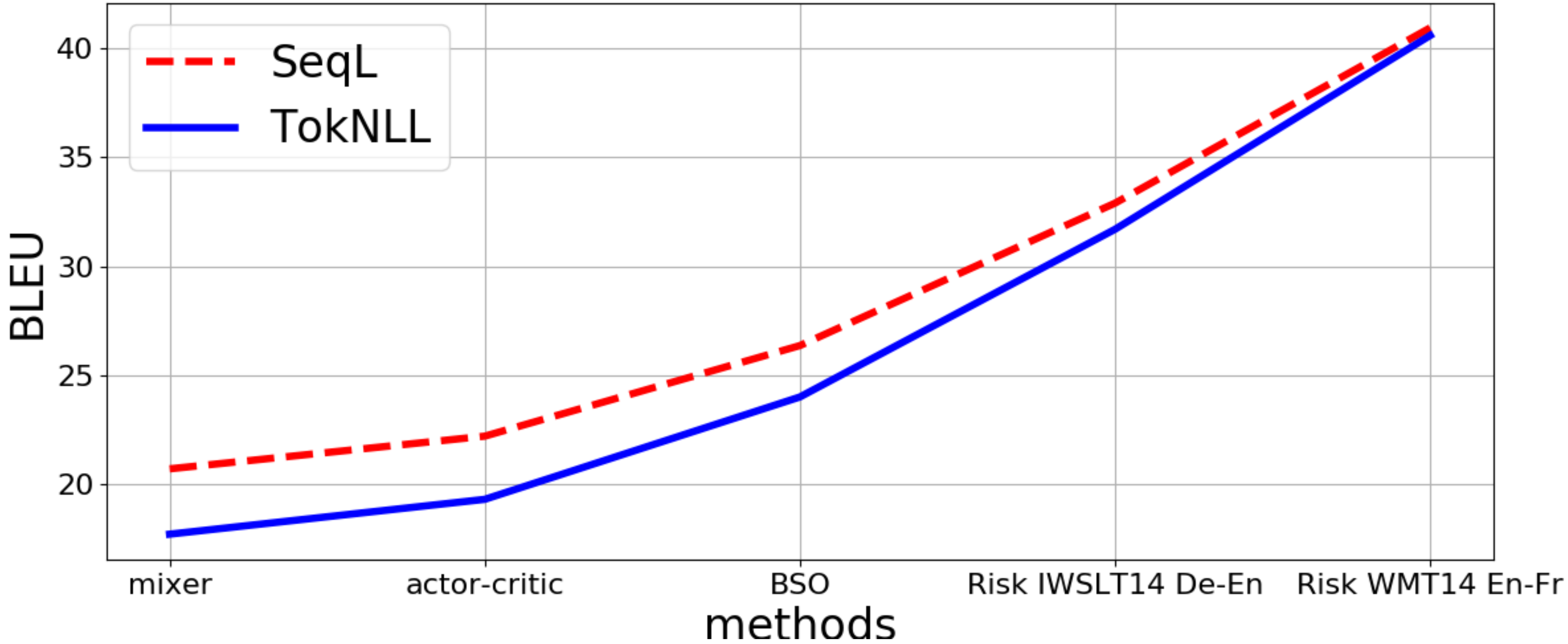
	TEST
TokNLL (Wiseman et al. 2016)	24.0
BSO (Wiseman et al. 2016)	26.4
Our re-implementation of their TokNLL	23.9
Risk on top of the above TokNLL	26.7

Fair Comparison to BSO

	TEST
TokNLL (Wiseman et al. 2016)	24.0
BSO (Wiseman et al. 2016)	26.4
Our re-implementation of their TokNLL	23.9
Risk on top of the above TokNLL	26.7

Methods fare comparably once the baseline is the same…

Diminishing Returns



On WMT'14 En-Fr, TokNLL gets 40.6 while Risk gets 41.0

The stronger the baseline, the less to be gained.

Summary

- Sequence level training does improve, but with **diminishing returns**. It’s computationally very expensive.
- Particular **method** to train at the sequence level **does not matter.**
- Important to use **pseudo reference** as opposed to real reference.

Analyzing Uncertainty: model fitting and effects on search

Why do larger beams perform worse?

Why is the model under-estimating rare words?

Analyzing uncertainty in neural machine translation
Myle Ott, Michael Auli, David Grangier, Marc'Aurelio Ranzato
in submission

slide credit: Marc'Aurelio Ranzato

Goal

BETTER UNDERSTANDING

rare word under-estimation

- artifact of beam search (argmax)?
- due to exposure bias?
- due to poor estimation?

performance degradation with wide beams

- due to heuristic nature of beam search?
- is the model poorly trained?

model fitting

- are NMT models calibrated?
- what do NMT models over/under-estimate?

Outline

- **Data uncertainty**
- Search
- Analyzing the model distribution

Data Uncertainty

Intrinsic

- there are many semantically equivalent translations of the same sentence. E.g.: style, skipping prepositions, choice of words, structural choices (active/passive), etc.

EXAMPLE
Source: The night before would be practically sleepless .

Target #1: La nuit qui précède pourrait s’avérer quasiment blanche .
Target #2: Il ne dormait pratiquement pas la nuit précédente .
Target #3: La nuit précédente allait être pratiquement sans sommeil .
Target #4: La nuit précédente , on n’a presque pas dormi .
Target #5: La veille , presque personne ne connaitra le sommeil .

Data Uncertainty

Intrinsic

- there are many semantically equivalent translations of the same sentence. E.g.: style, skipping prepositions, choice of words, structural choices (active/passive), etc.
- under-specification. E.g.: gender, tense, number, etc.

EXAMPLE
Source: nice .

Target #1: chouette .
Target #2: belle .
Target #3: beau .

Data Uncertainty

Intrinsic

- there are many semantically equivalent translations of the same sentence. E.g.: style, skipping prepositions, choice of words, structural choices (active/passive), etc.
- under-specification. E.g.: gender, tense, number, etc.

Extrinsic

- noise in the data. E.g.: partial translation, copies of the source, etc.

Data Uncertainty

Intrinsic

- there are many semantically equivalent translations of the same sentence. E.g.: style, skipping prepositions, choice of words, structural choices (active/passive), etc.
- under-specification. E.g.: gender, tense, number, etc.

Extrinsic

- noise in the data. E.g.: partial translation, copies of the source, etc.

Example: on WMT between 1 and 2% of the training target sentences are copies of the source.

Data Uncertainty

Intrinsic

- there are many semantically equivalent translations of the same sentence. E.g.: style, skipping prepositions, choice of words, structural choices (active/passive), etc.
- under-specification. E.g.: gender, tense, number, etc.

HOW DOES THIS AFFECT NMT?

Extrinsic

- noise in the data. E.g.: partial translation, copies of the source, etc.

Example: on WMT between 1 and 2% of the training target sentences are copies of the source.

Outline

- Data uncertainty
- **Search**
- Analyzing the model distribution

Search

Find the most likely sequence according to the model:

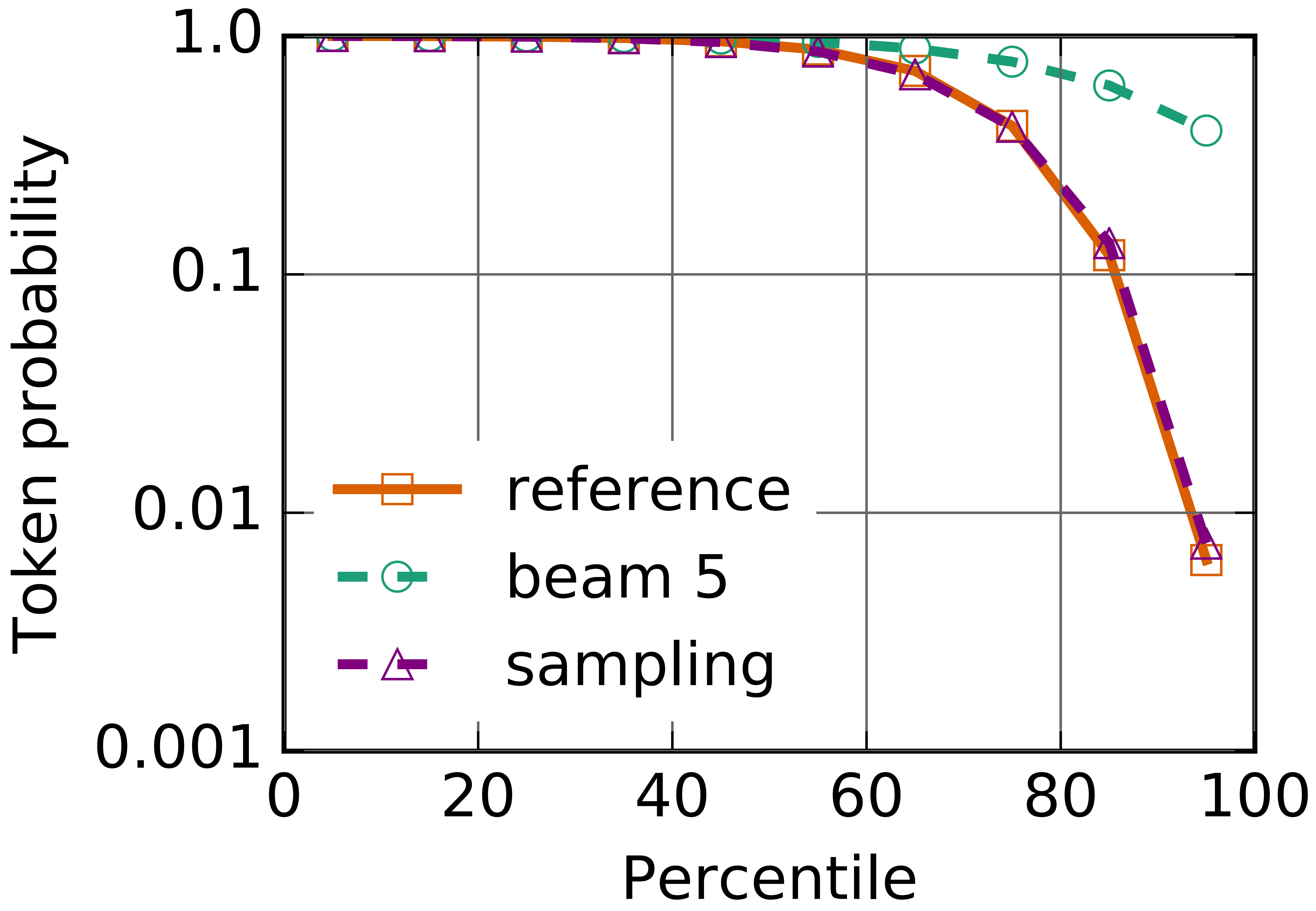
$$\arg\max_{y} p(y|x;\theta)$$

Preliminary questions:

- is beam search effective?
- is beam search efficient?
- are there better search strategies?

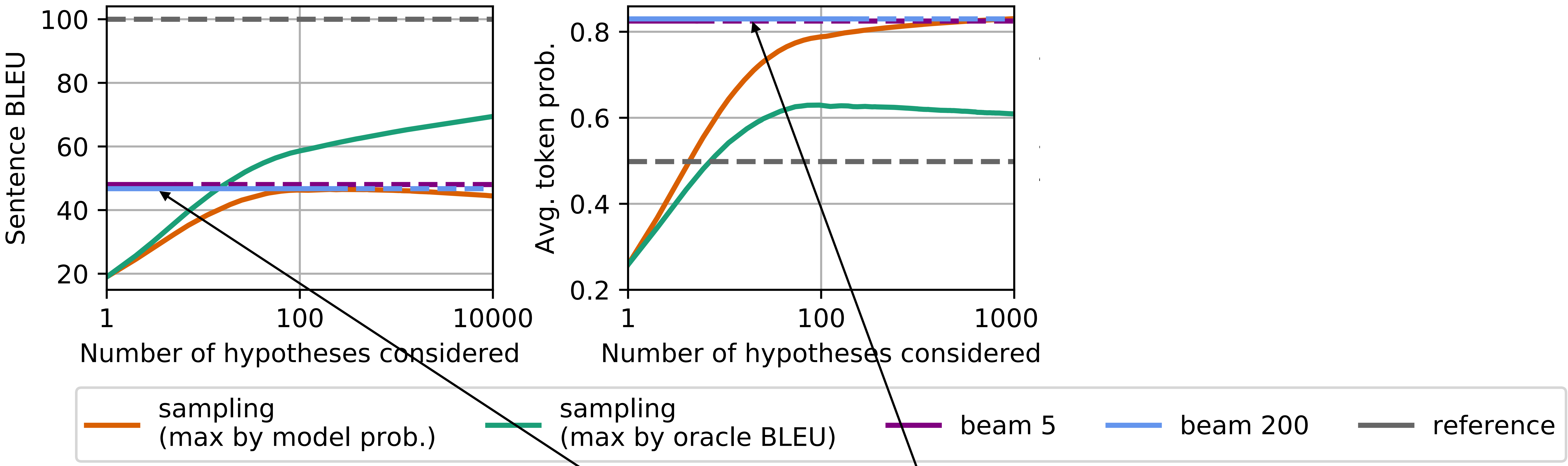
Search

- Convolutional NMT with attention
- 15 layers
- 716D embeddings
- ~250M parameters



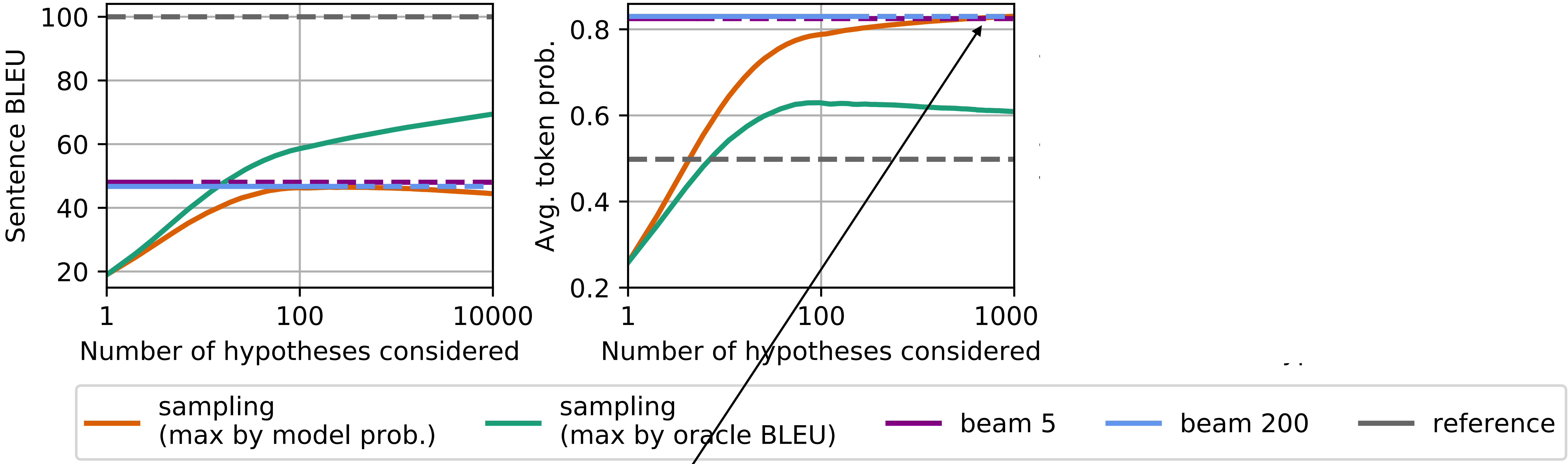
Beam search is very effective; only 20% of the tokens with probability < 0.7 (despite exposure bias)!

Search



- Increasing the beam width does not increase BLEU, while probability increases.

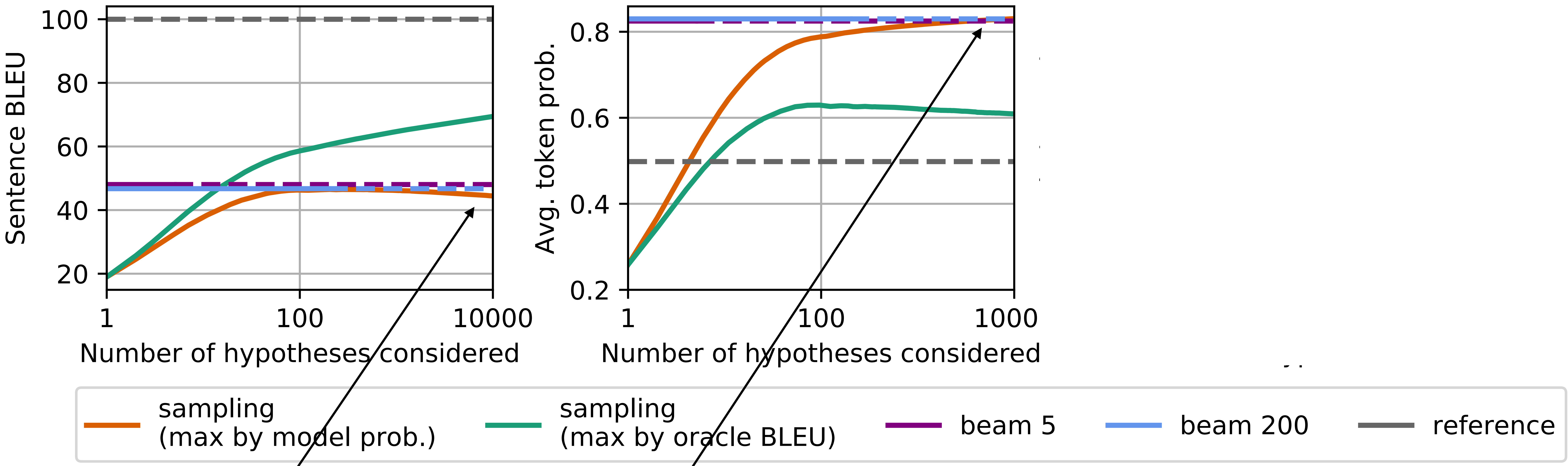
Search



- Increasing the beam width does not increase BLEU, while probability increases.

- Sampling can find hypotheses with similar logprob but:

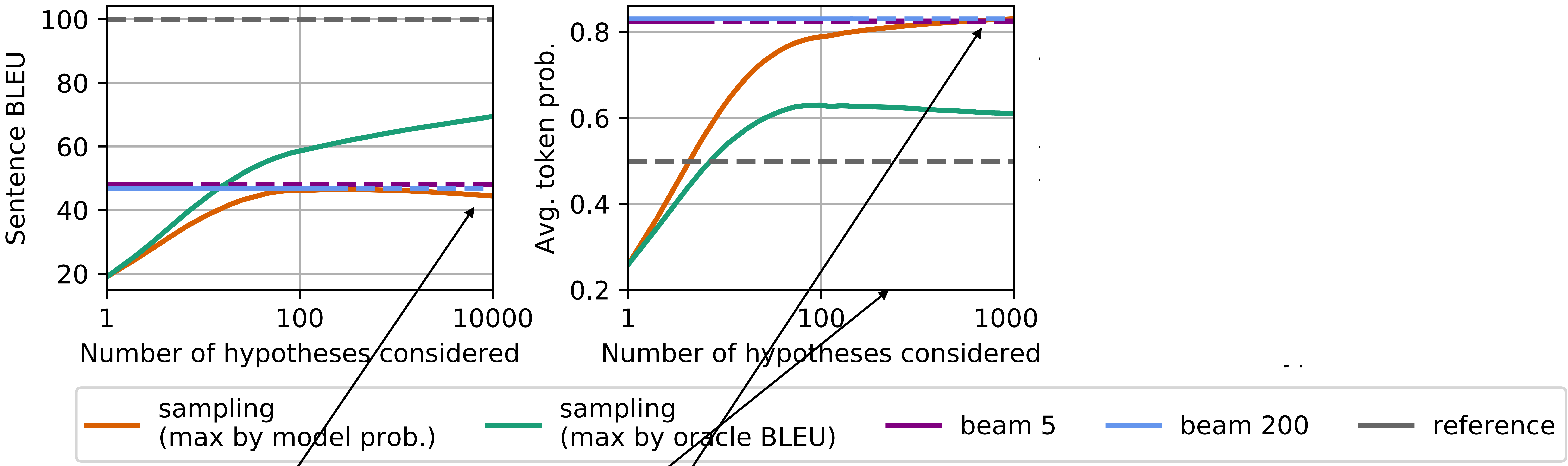
Search



- Increasing the beam width does not increase BLEU, while probability increases.

- Sampling can find hypotheses with similar logprob but:
  - lower BLEU

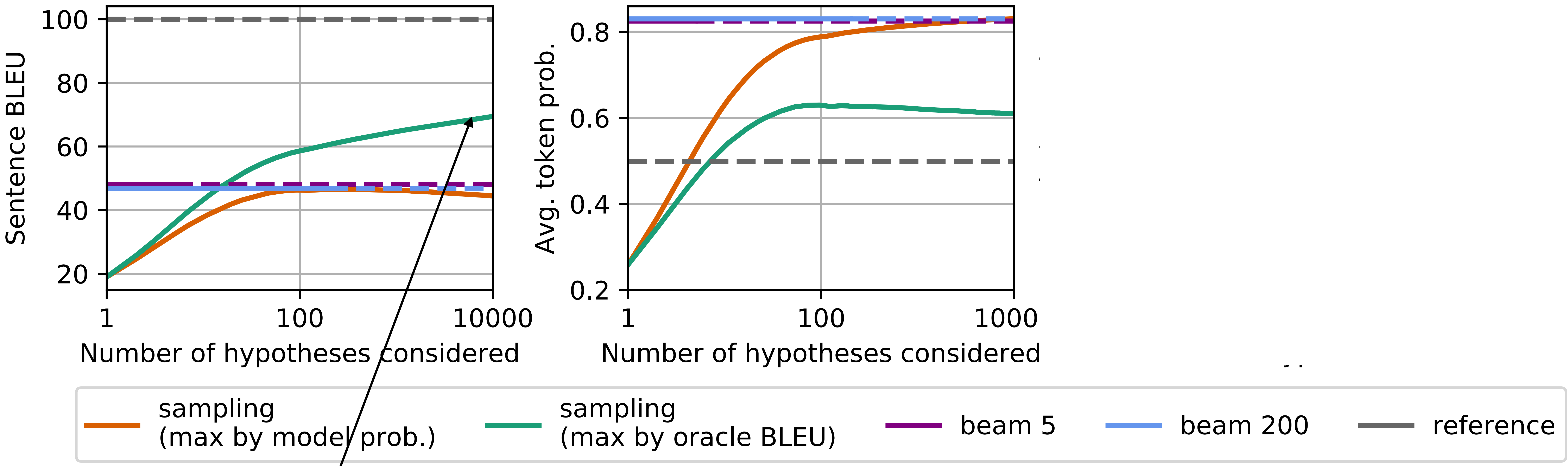
Search



- Increasing the beam width does not increase BLEU, while probability increases.

- Sampling can find hypotheses with similar logprob but:
    - lower BLEU
    - it’s 20x less inefficient

Search

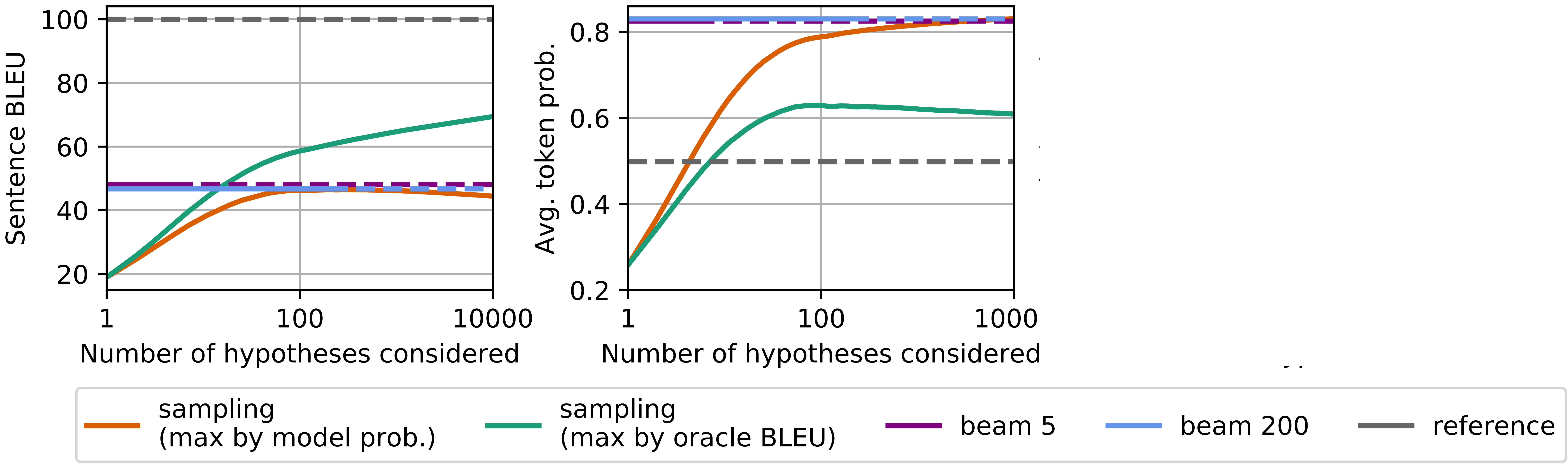


- Increasing the beam width does not increase BLEU, while probability increases.

- Sampling can find hypotheses with similar logprob but…

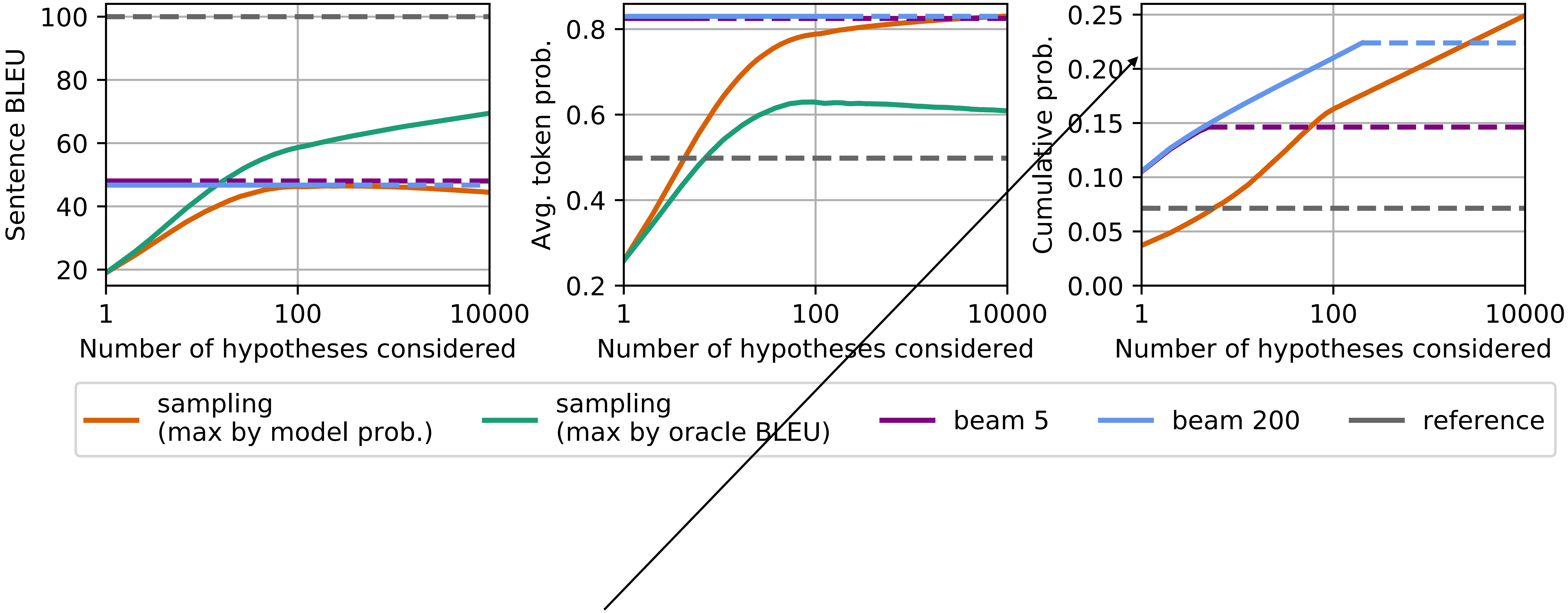
- Among the generated hypotheses, there exist at least one that is pretty close to the reference.

Search



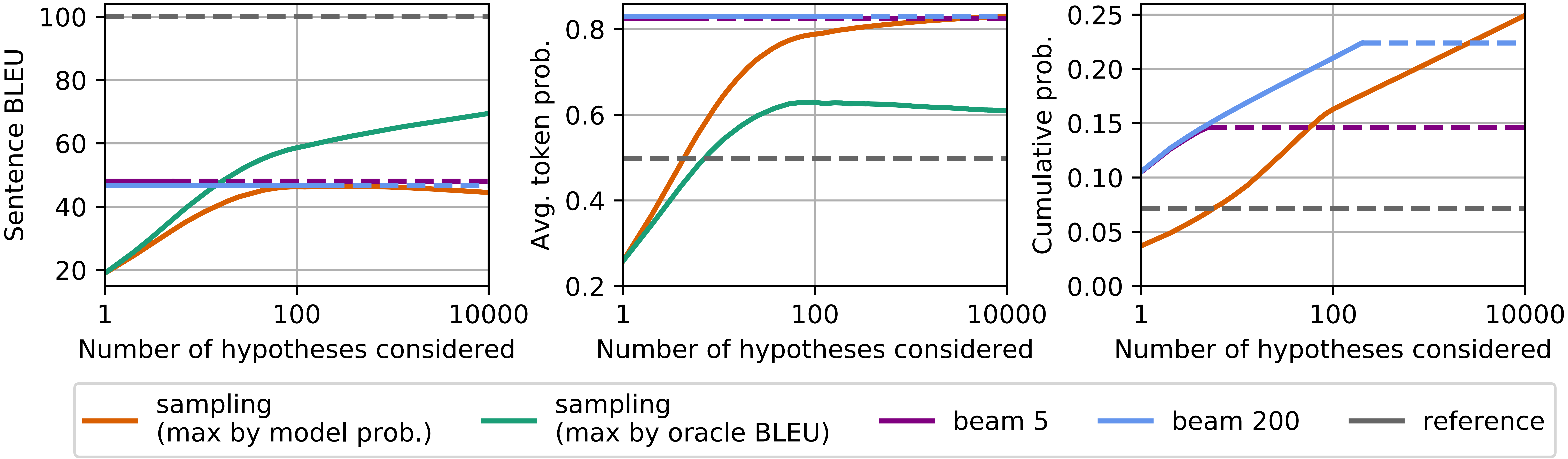
Beam search is very effective and efficient.
However, large beams yield worse BLEU!

Search



- Beam 200/sampling 10K cover only about 22% of the total probability mass
  Where is the rest?

Search



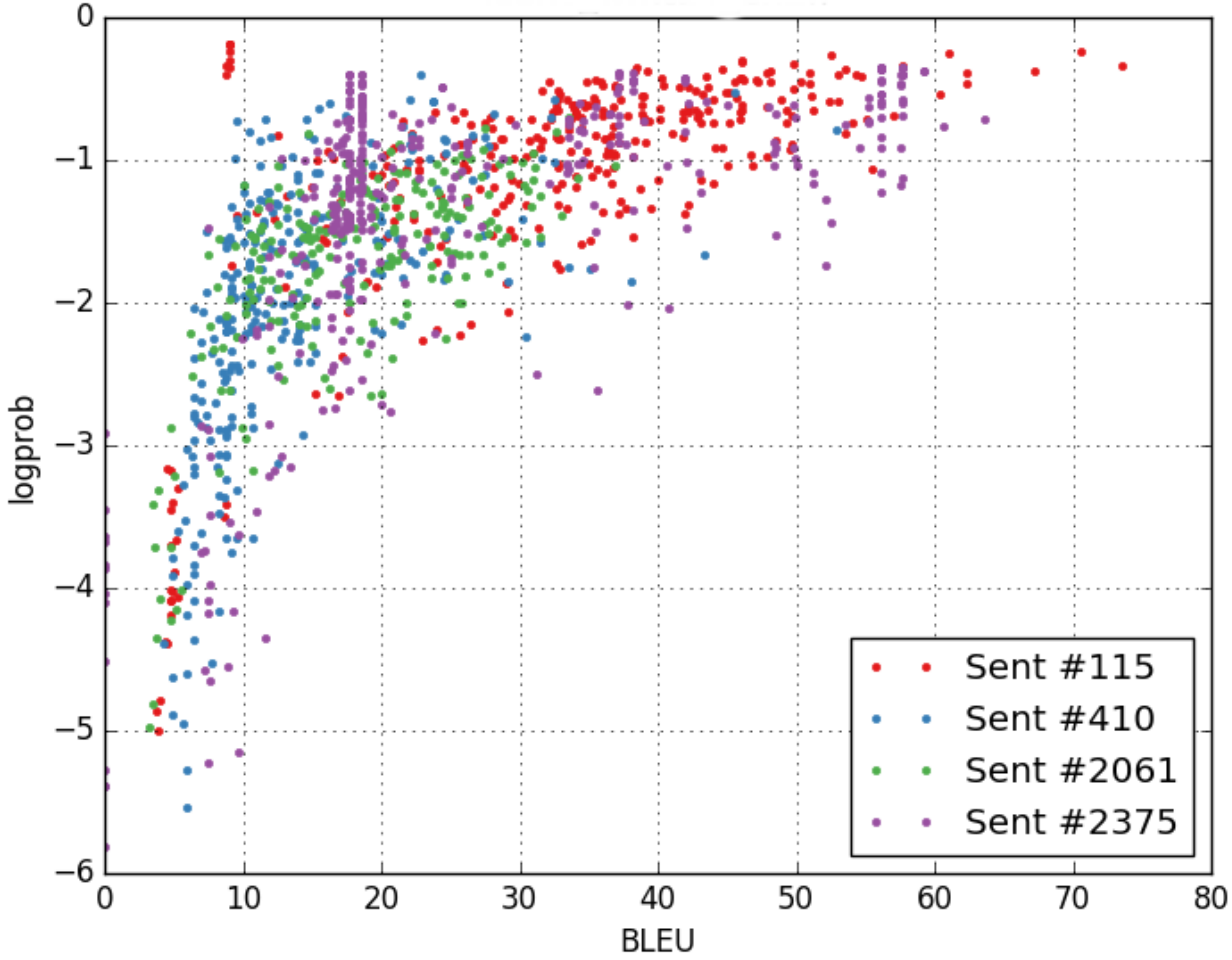
Model distribution has a lot of uncertainty.

Puzzling Observations

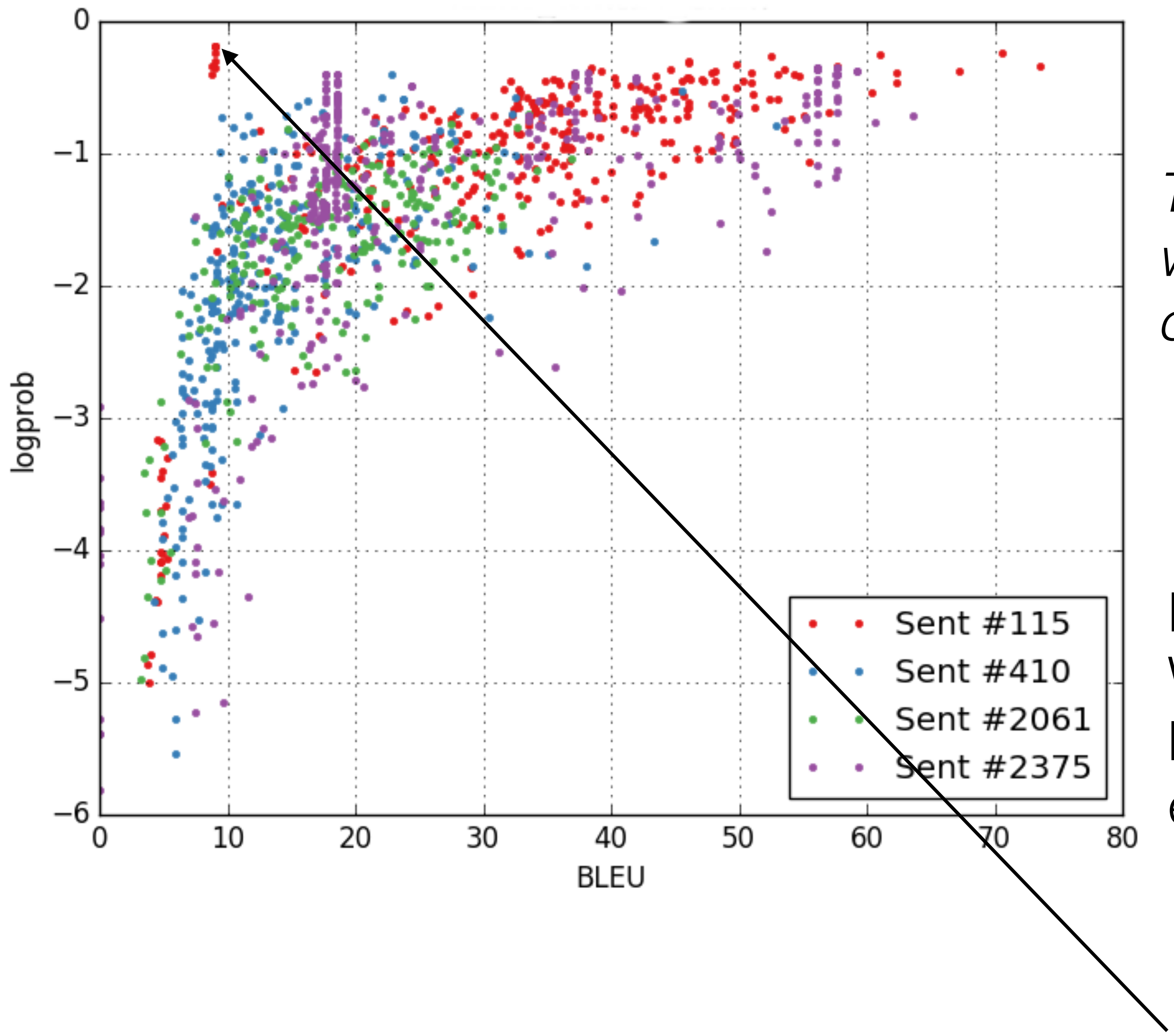
- Increasing beam size hurts performance in terms of BLEU.

- Large beam accounts only for fraction of total probability mass.

Scatter Plot of Samples



Scatter Plot of Samples



Source #115 (red):

The first nine episodes of Sheriff [unk]'s Wild West will be available from November 24 on the site [unk] or via its application for mobile phones and tablets.

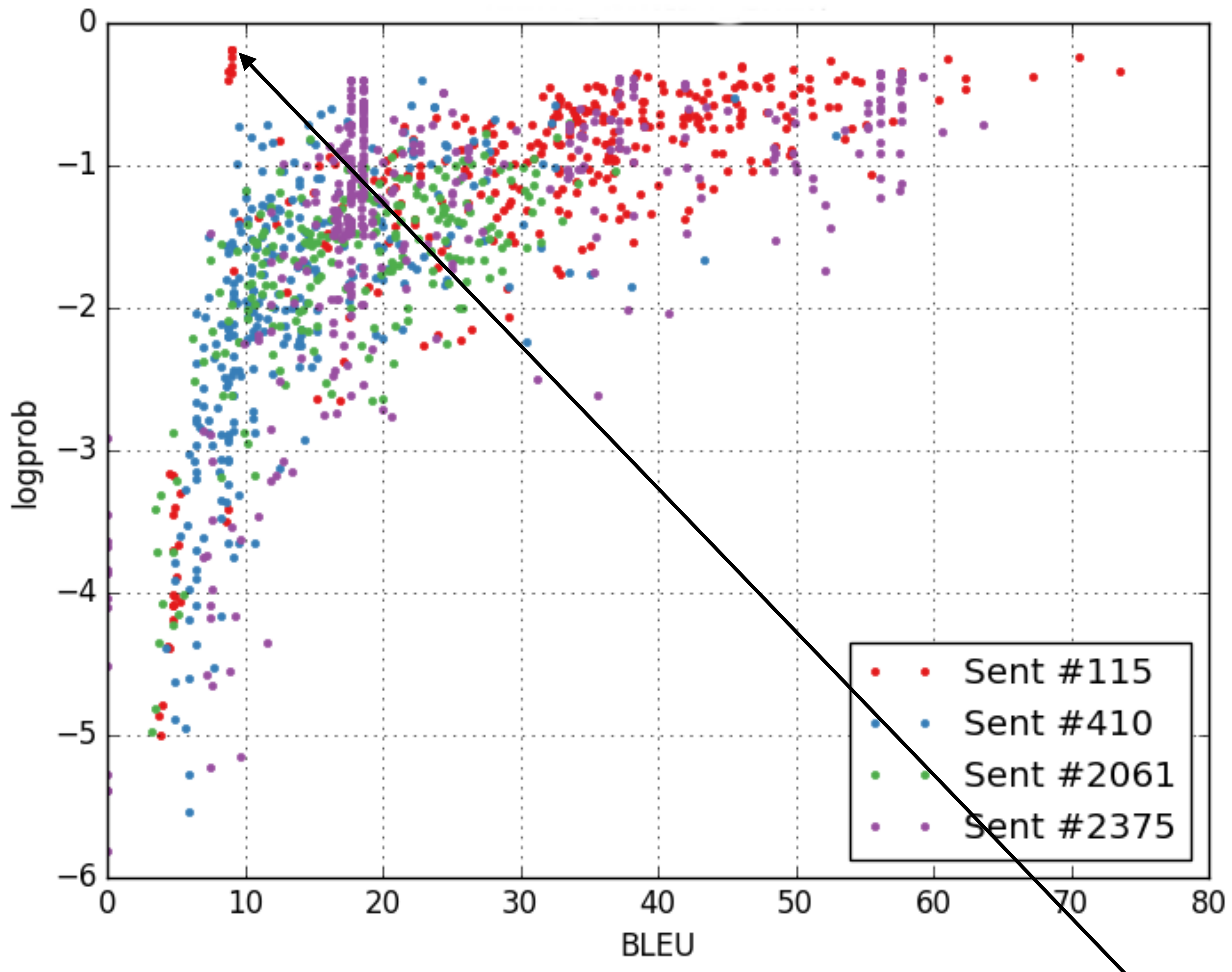
Target #115 (red):

Les neuf premiers épisodes de [unk] [unk] s Wild West seront disponibles à partir du 24 novembre sur le site [unk] ou via son application pour téléphones et tablettes.

High-logp low BLEU sample:

The first nine episodes of Sheriff [unk] s Wild West will be available from November 24 on the site [unk] or via its application for mobile phones and tablets.

Scatter Plot of Samples



Source #115 (red):

The first nine episodes of Sheriff [unk]'s Wild West will be available from November 24 on the site [unk] or via its application for mobile phones and tablets.

Target #115 (red):

Les neuf premiers épisodes de [unk] [unk] s Wild West seront disponibles à partir du 24 novembre sur le site [unk] ou via son application pour téléphones et tablettes.

High-logp low BLEU sample:

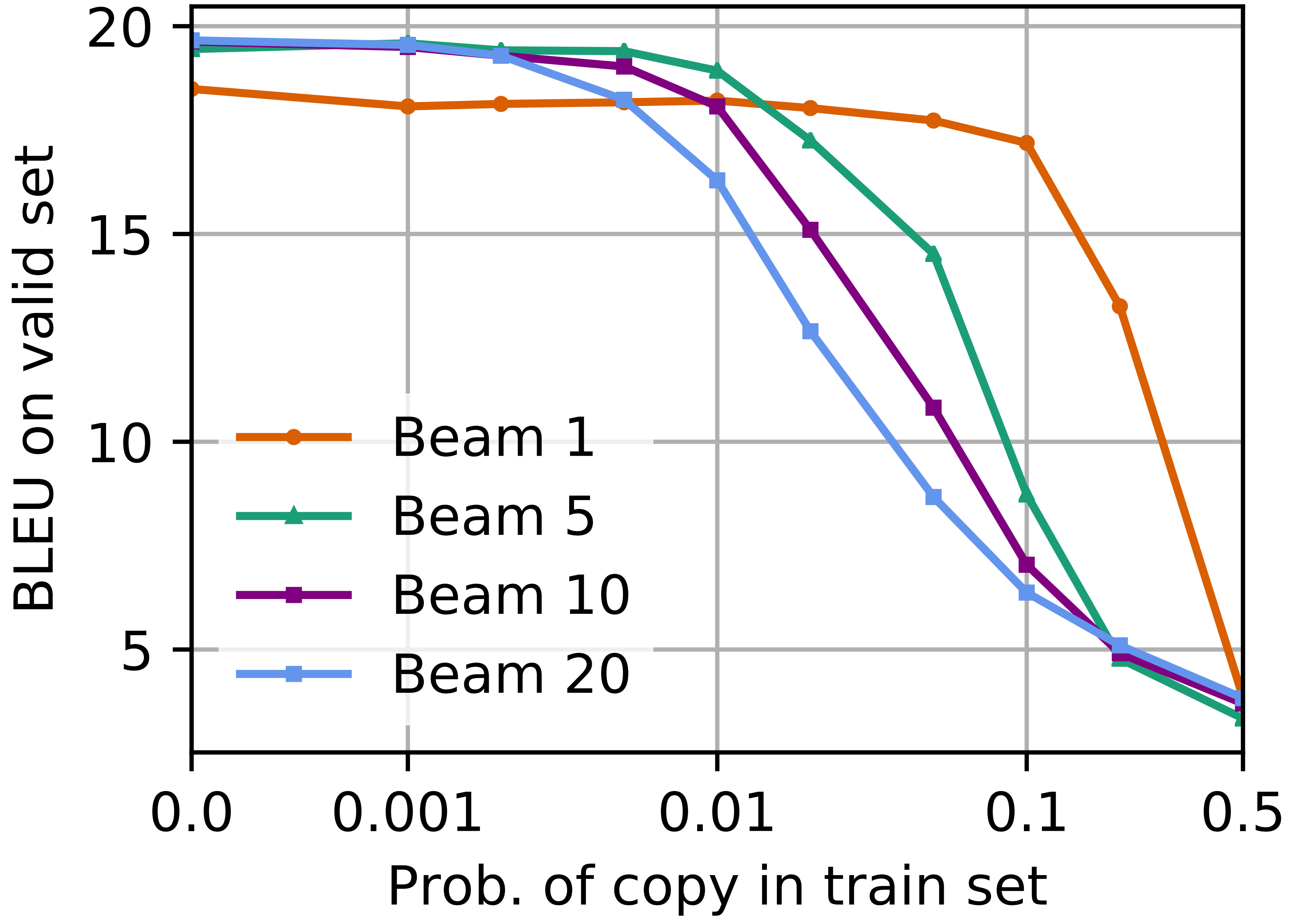
The first nine episodes of Sheriff [unk] s Wild West will be available from November 24 on the site [unk] or via its application for mobile phones and tablets.

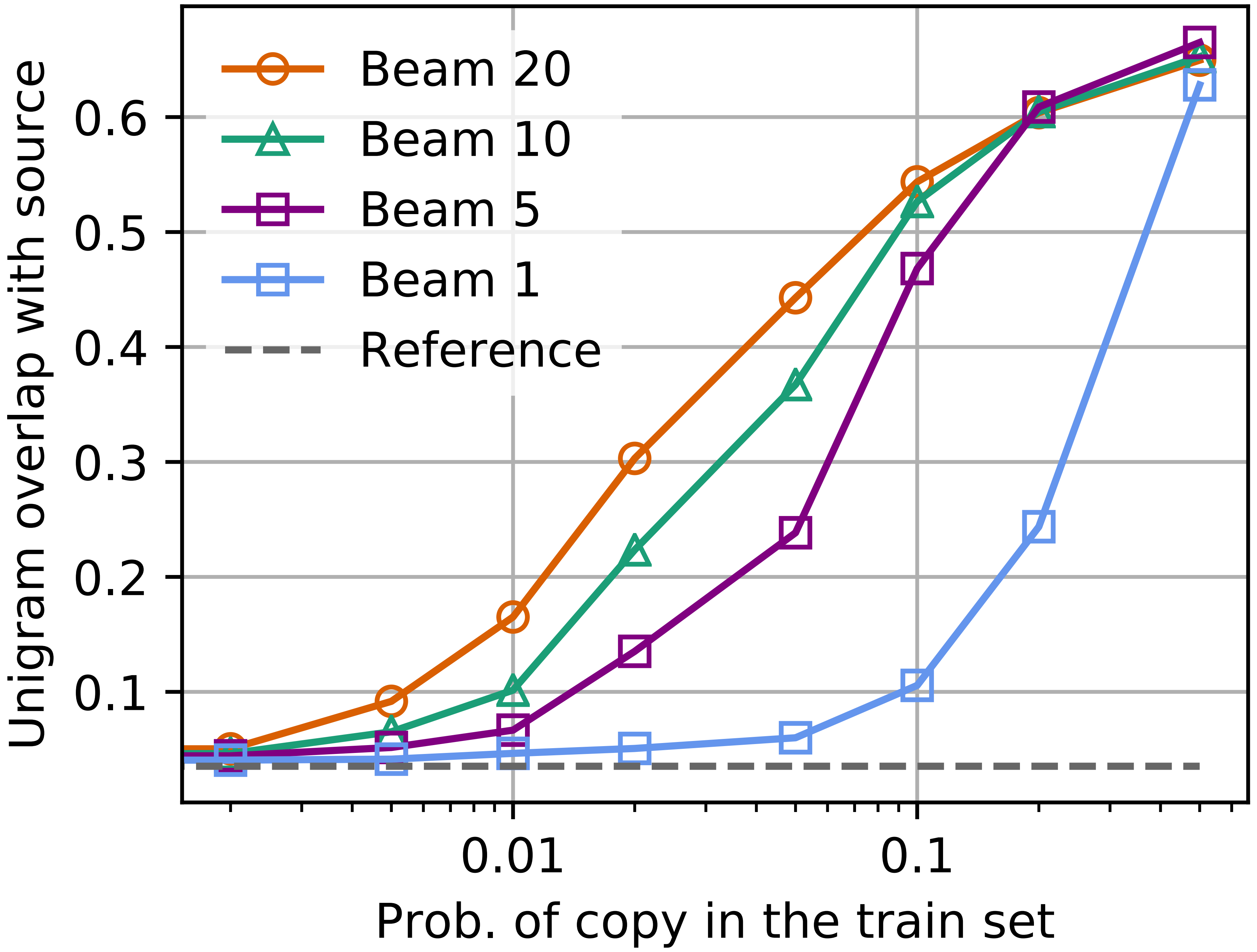
Model generates copies of source sentence!
Why does beam find this?

Uncertainty <—> Search

- Hard to characterize how uncertainty affects search in general.
- We can however simulate (extrinsic) uncertainty:
  - add fraction of “copy noise” and check effects on search.

Uncertainty <—> Search





Large beams are more prone to copy the source, hence the lower BLEU.

Uncertainty <—> Search

- Source: The first nine episodes of Sheriff <unk> ’s Wild West will be available from November 24 on the site <unk> or via its application for mobile phones and tablets .
- Target (reference): Les neuf premiers épisodes de <unk> <unk> s Wild West seront disponibles à partir du 24 novembre sur le site <unk> ou via son application pour téléphones et tablettes .
- Sample: The first nine episodes of Sheriff <unk> s Wild West will be available from November 24 on the site <unk> or via its application for mobile <unk> and tablets .

Uncertainty <—> Search

- **Source**: The first nine episodes of Sheriff <unk> ′s Wild West will be available from November 24 on the site <unk> or via its application for mobile phones and tablets .
- **Target (reference)**: Les neuf premiers épisodes de <unk> <unk> s Wild West seront disponibles à partir du 24 novembre sur le site

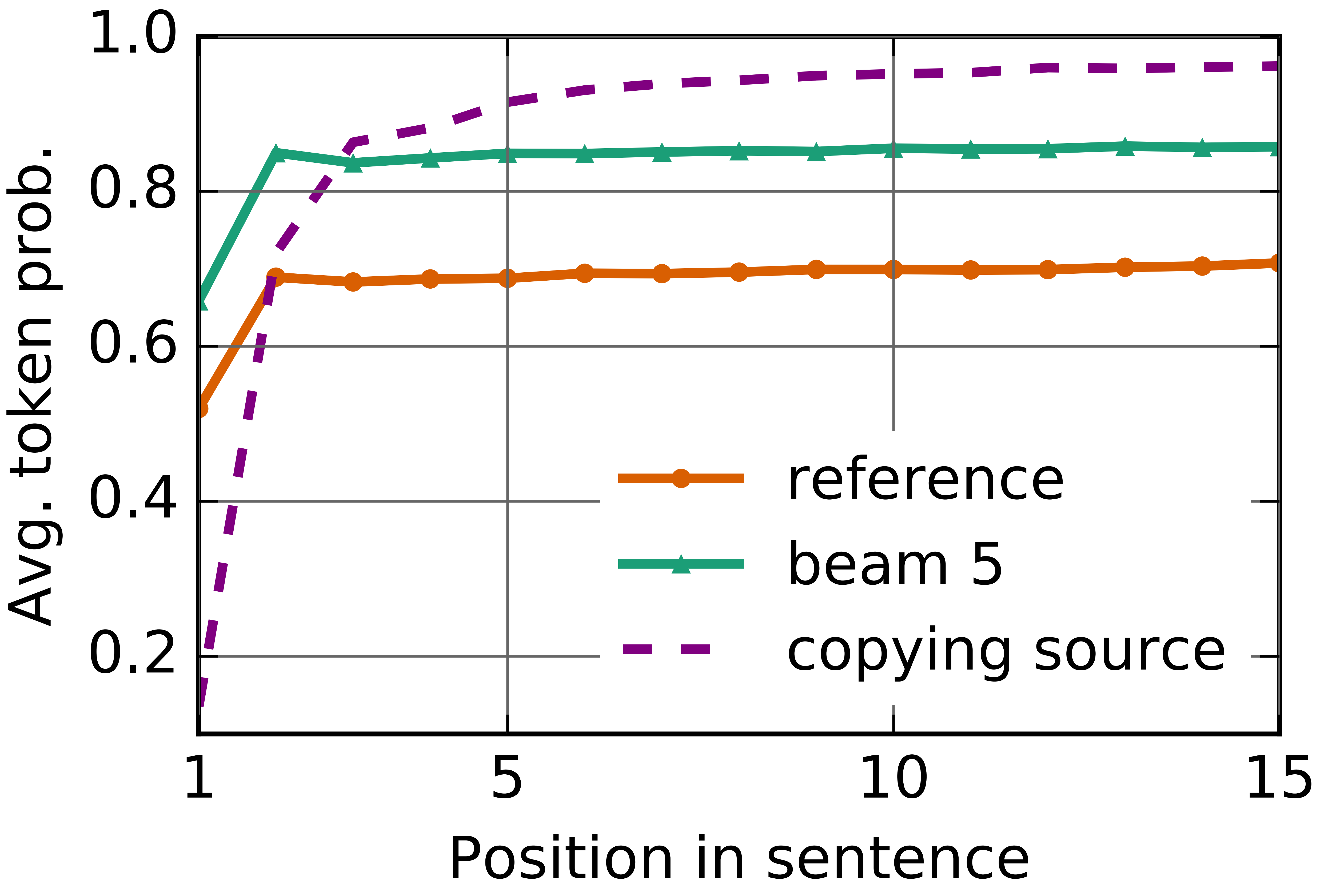
log probs: -4.53 -0.02 -0.28 -0.11 -0.01 -0.001 -0.004 -0.002 -0.001 -0.005

- **Sample**: The first nine episodes of Sheriff <unk> s Wild West will be available from November 24 on the site <unk> or via its application for mobile <unk> and tablets .

Inductive bias:

NMT + attention has easy time to learn how to copy!

Uncertainty <—> Search



Initial tokens pay big penalty, but afterwards copying the source is cheap. Only large beams can discover this.

Uncertainty <—> Search

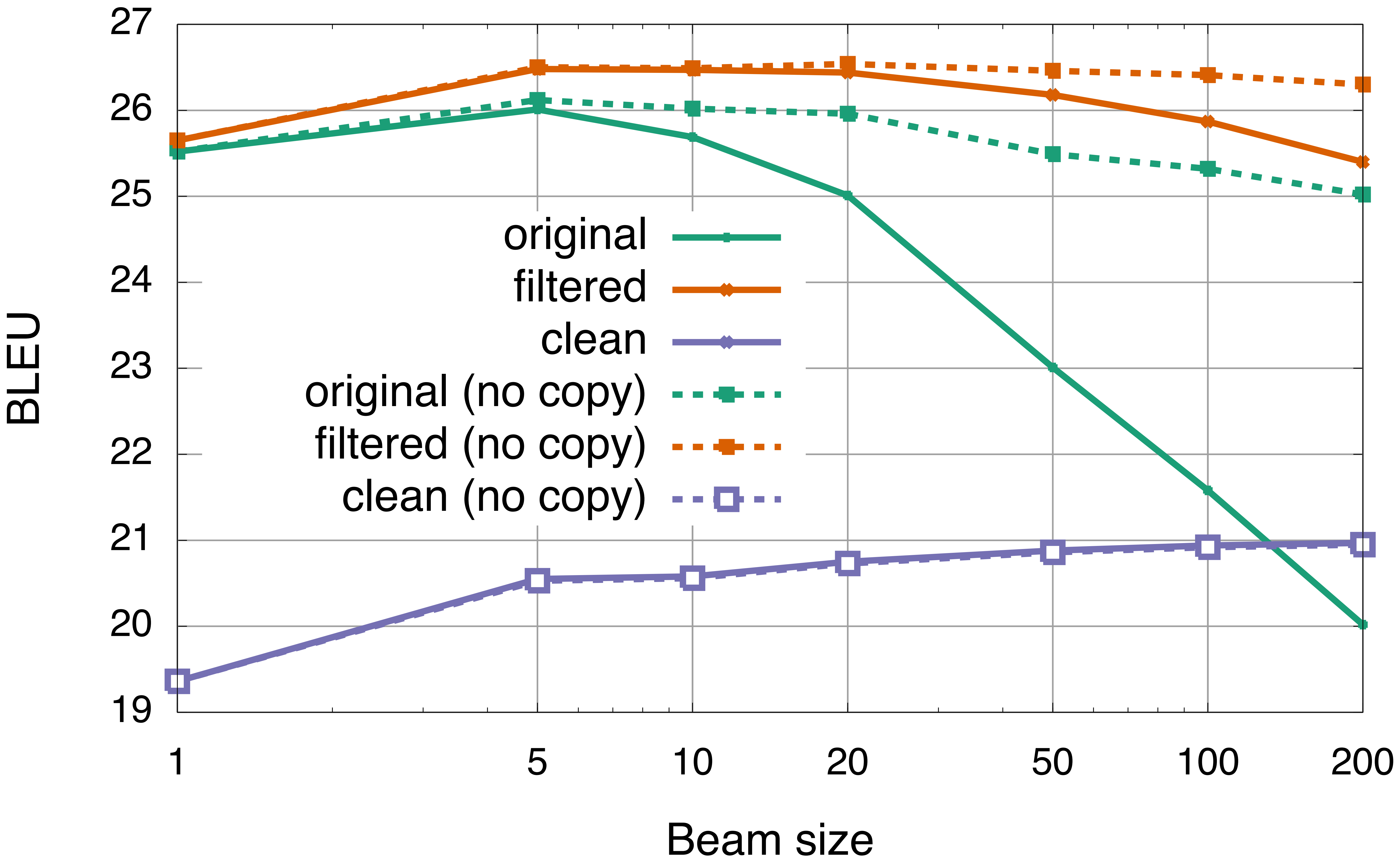
- On WMT’14 En-Fr: ~2% of the training target sentences are copies of the corresponding source.

- Beam@1 yields copies 2.6% of the time.
- Beam@20 yields copies 3.5% of the time.

Fixing Search

- **Filtering the data** with model trained on “clean data” to remove copies from training set.
- **Constrain beam search** not to output too many words from the source sentence.

Fixing Search



Search & Uncertainty

- Search works very well, i.e. beam finds likely model hypotheses.
- However, it can find noisy sentences (model is wrong), that are merely due to noise in the data collection process.
- This explains why BLEU deteriorates for large beams.
- There are easy fixes.

Puzzling Observations

- Increasing beam size hurts performance in terms of BLEU.

Understood

- Large beam accounts only for fraction of total probability mass.

Outline

- Data uncertainty
- Search
- **Analyzing the model distribution**

Model Distribution

Check match between model and data distribution is challenging:

- For a given source sentence, we typically observe only one sample from the data distribution (the provided reference).
- Enumeration of all possible sequences using the data distribution is intractable anyway.

We would like to:

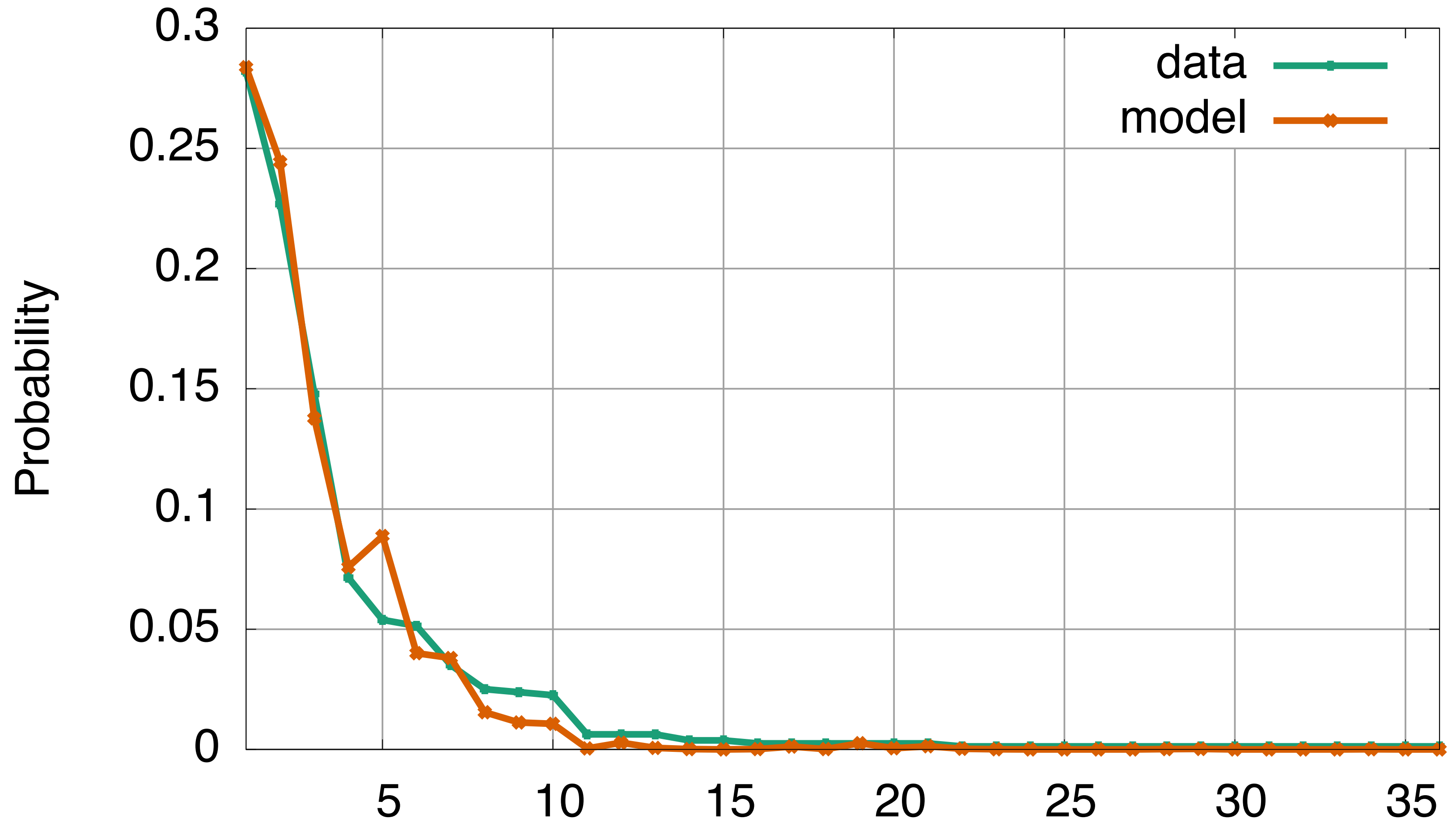
- check how closely model and data distribution match
- understand when they differ and why

Anecdotal Example

In the training set, some source sentences appear many times.
Use corresponding targets to estimate the underlying data distribution!

EXAMPLE
Source: (The president cutoff the speaker) .

Appears 798 times on the training set with 36 unique translations.

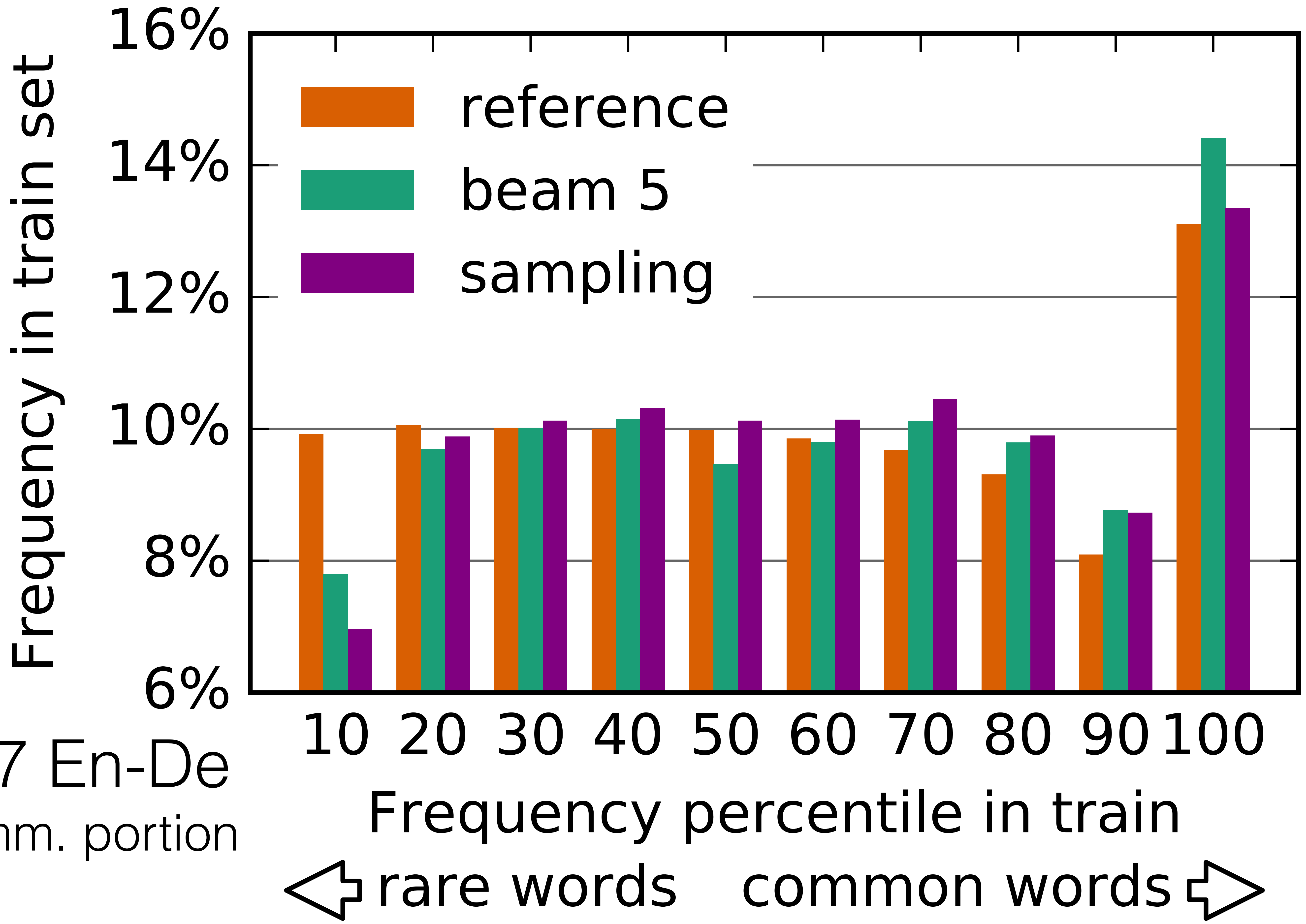


For this source sentence, model and data distribution match very well!

Analysis Tools

- Token level fitting
- Sentence level calibration
- Set level calibration
- Other necessary conditions

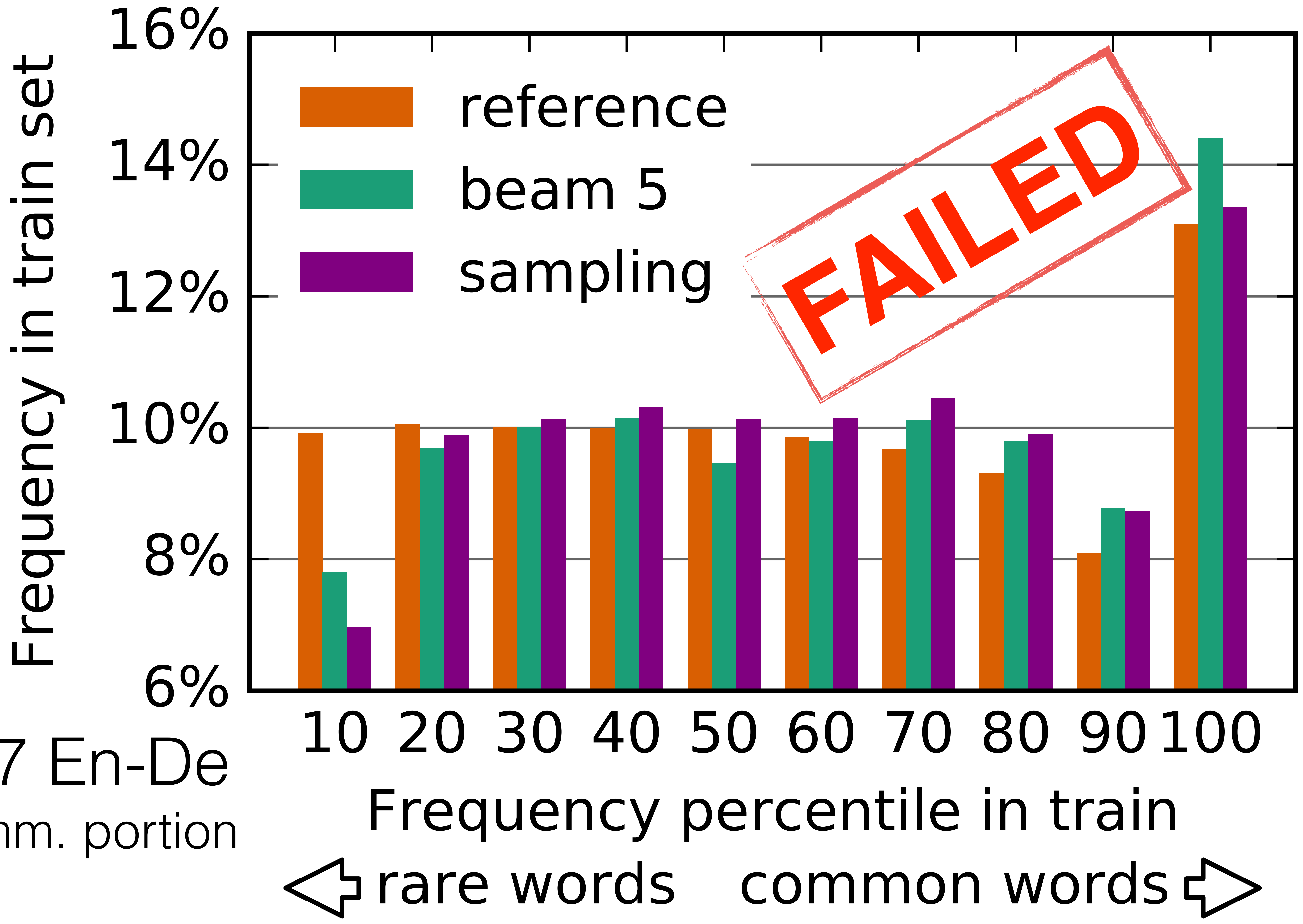
Token Level: Matching Unigram Stats



WMT'17 En-De
news-comm. portion

Model grossly under-estimate rare words.
Beam over-estimates frequent words, as expected.

Token Level: Matching Unigram Stats



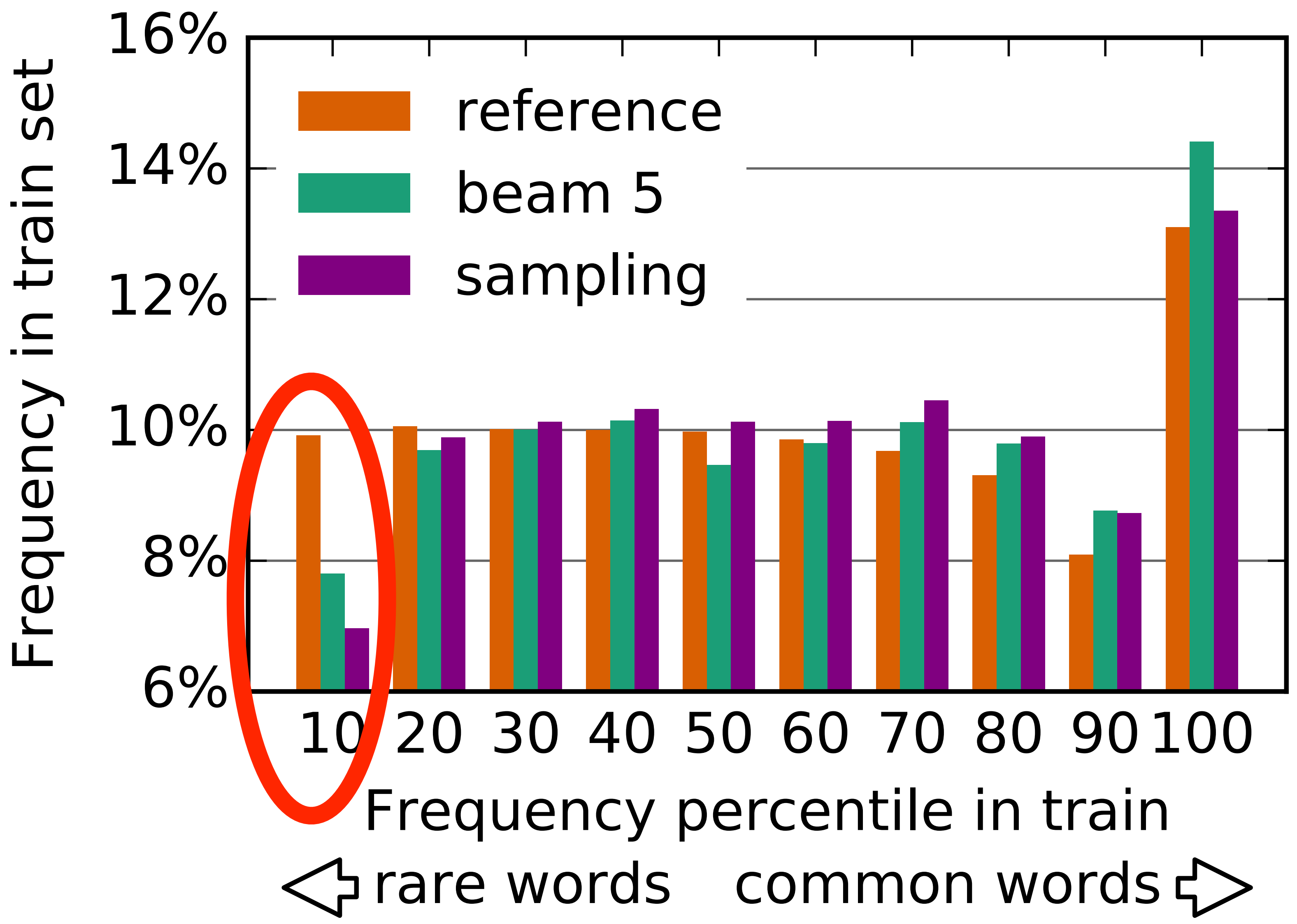
WMT'17 En-De

news-comm. portion

Model grossly under-estimate rare words.
Beam over-estimates frequent words, as expected.

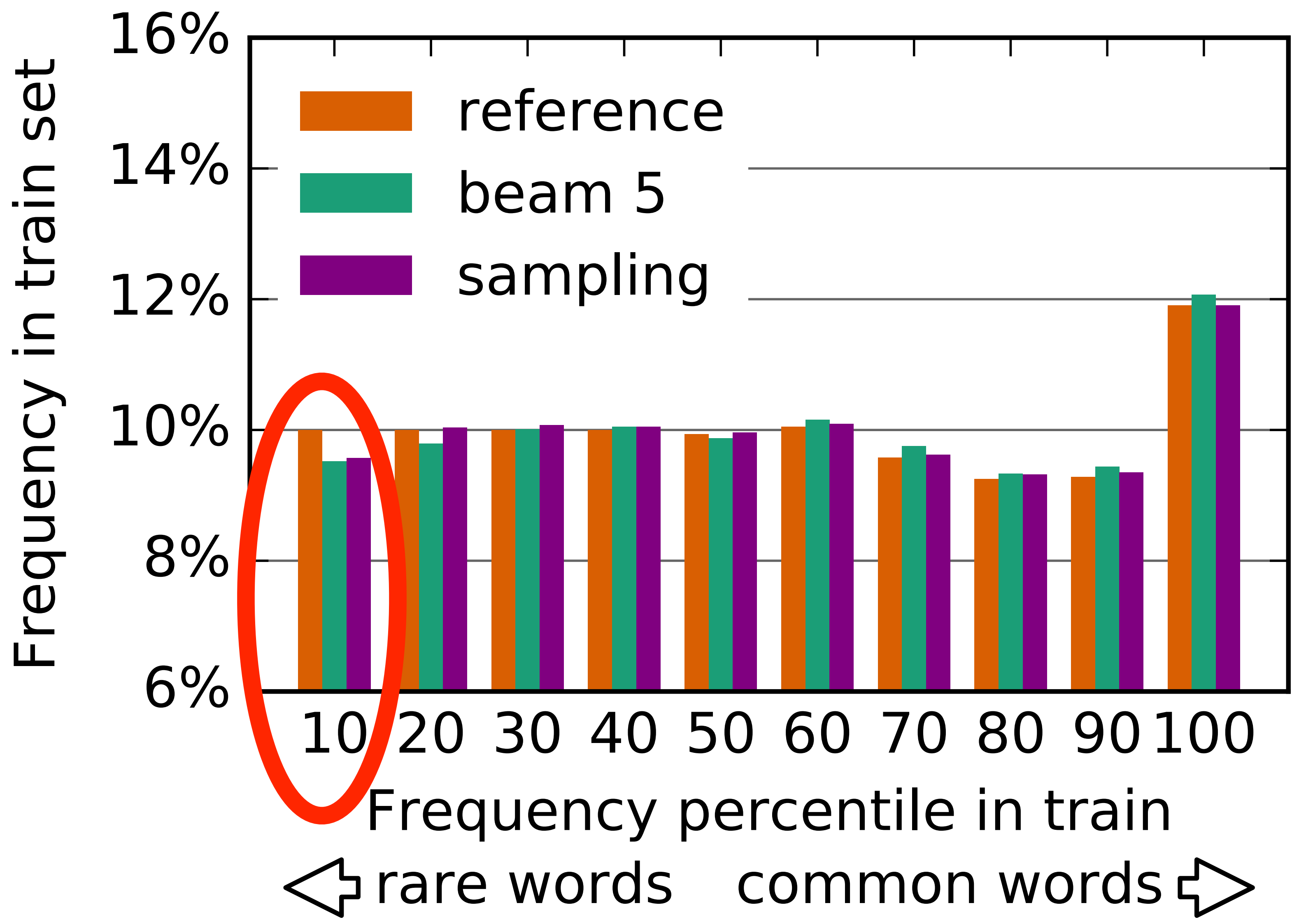
Token Level: Matching Unigram Stats

WMT'17 En-De news-comm. portion



- ~300K parallel sentences
- 21 BLEU on test
- median freq. in 10% bin: 12

WMT'14 En-Fr

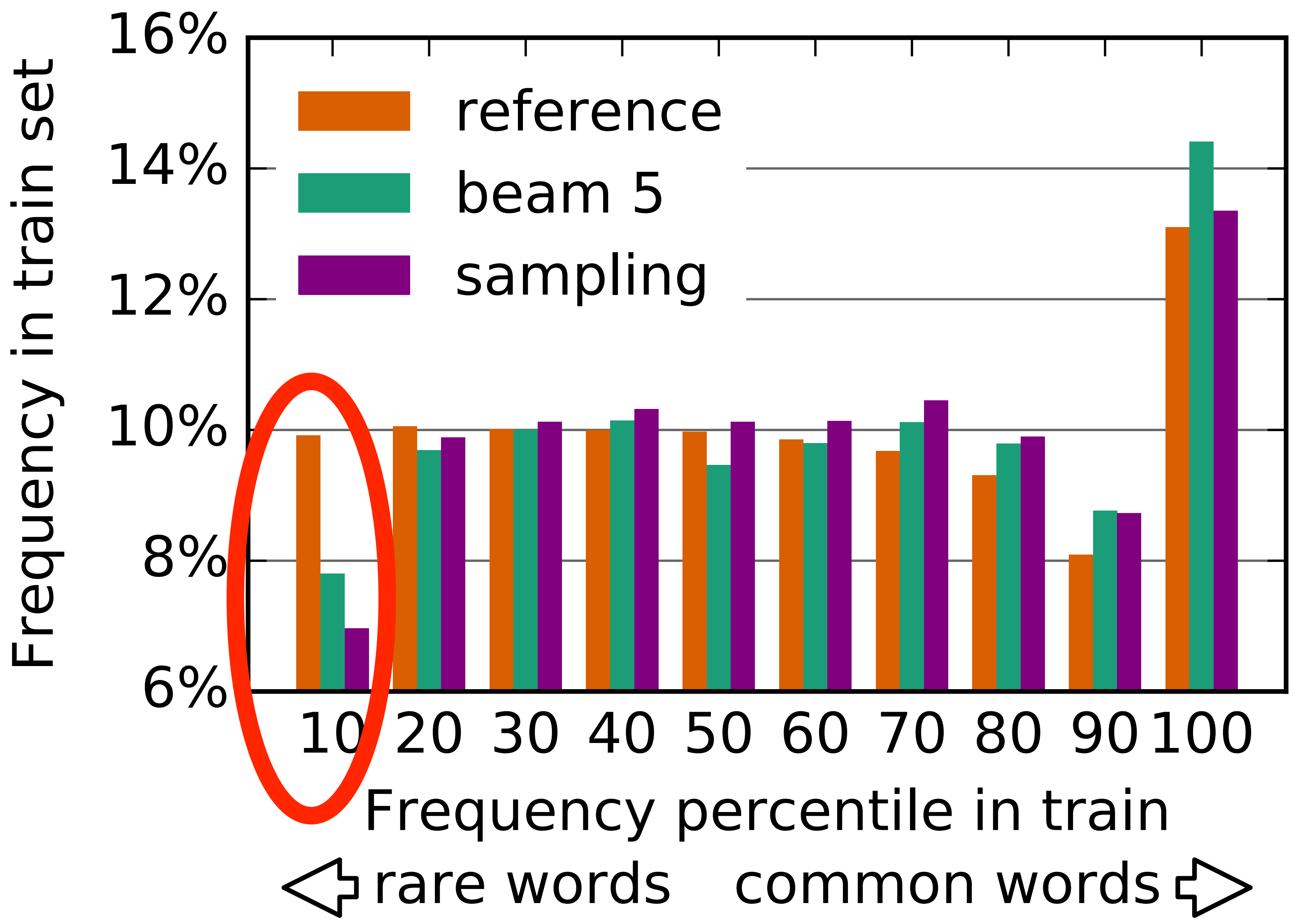


- ~35M parallel sentences
- 41 BLEU on test
- median freq. in 10% bin: 2500

More data & better model close the gap, but rare words are still under-estimated.

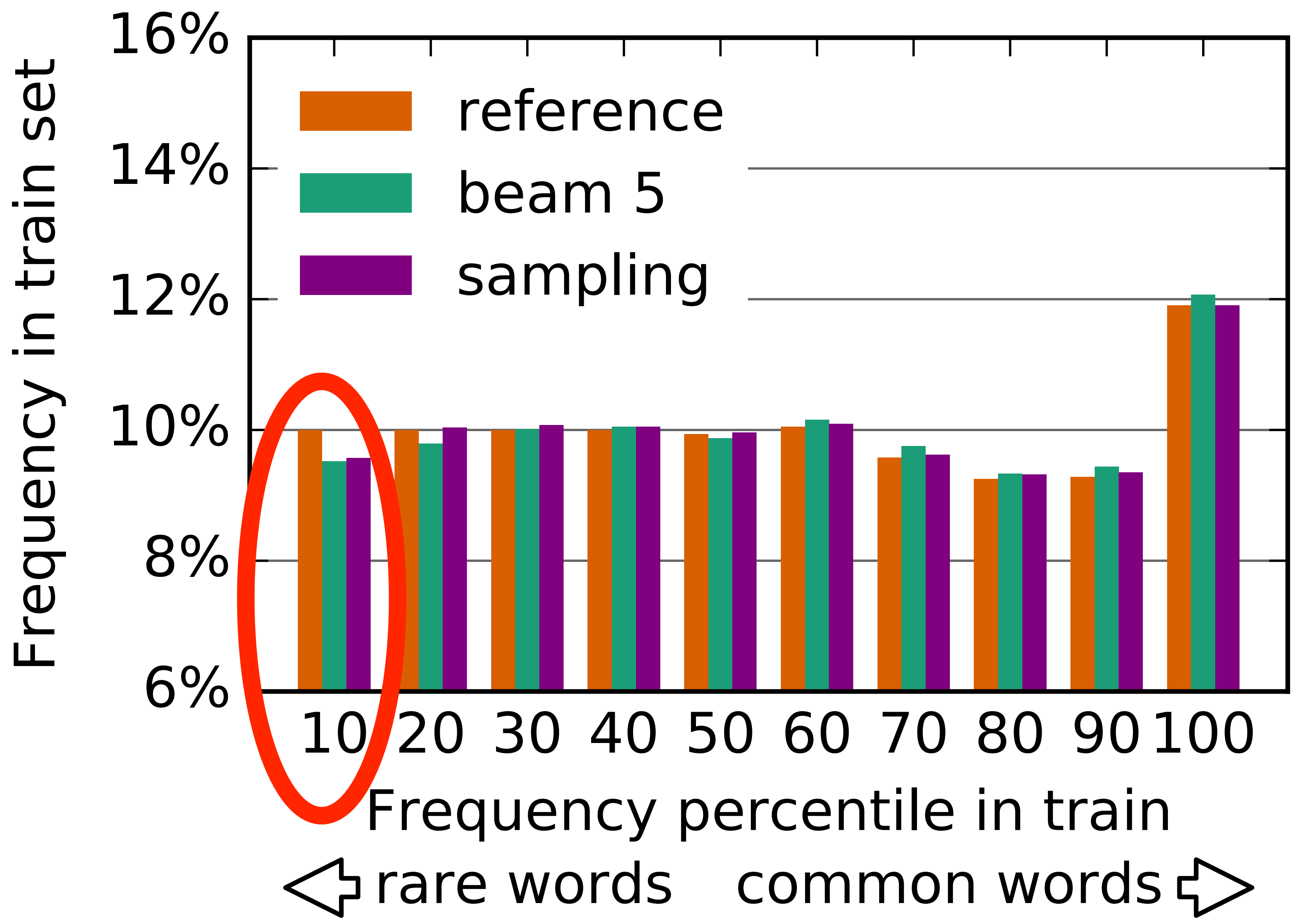
Token Level: Matching Unigram Stats

WMT'17 En-De news-comm. portion



- ~300K parallel sentences
- 21 BLEU on test
- median freq. in 10% bin: 12

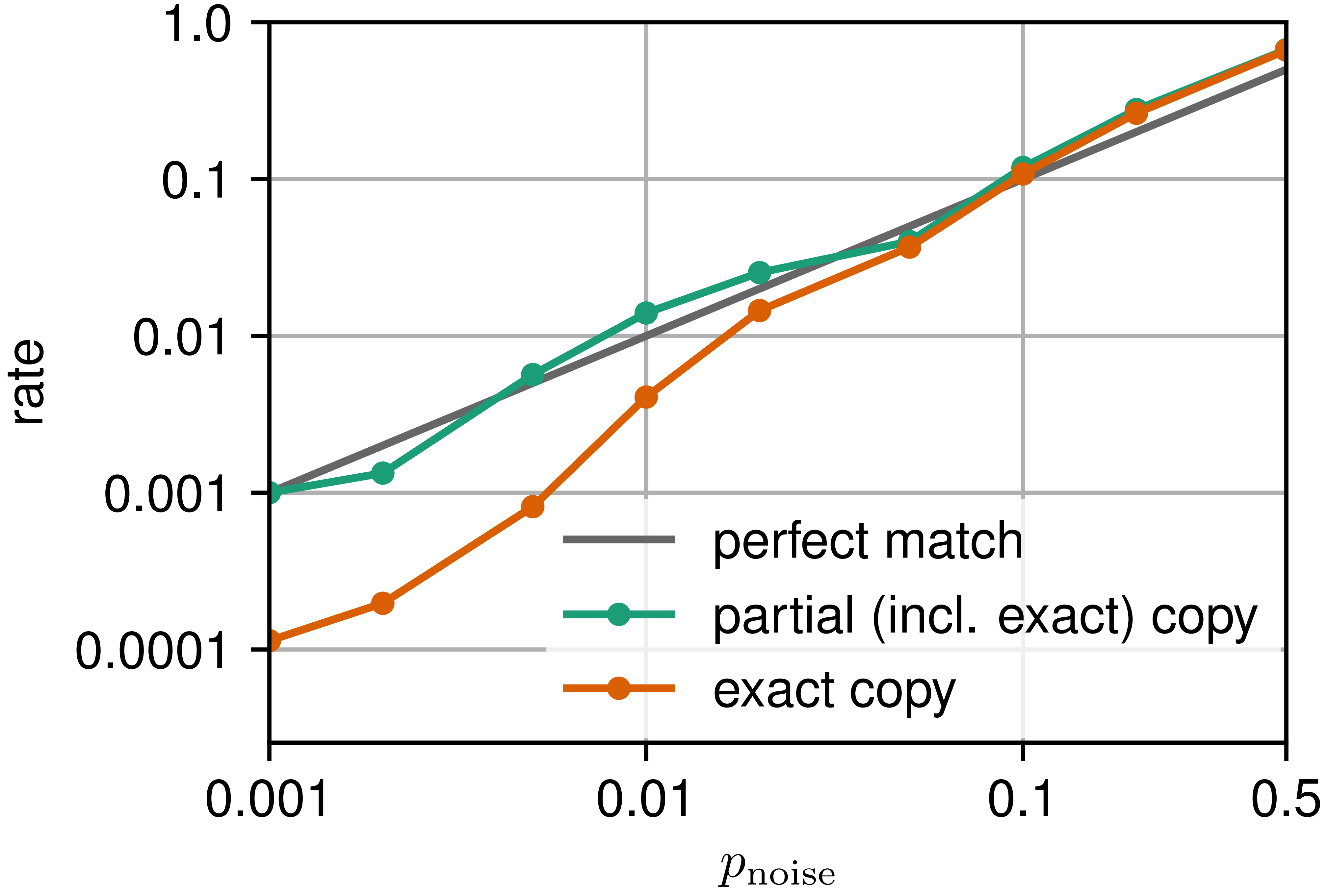
WMT'14 En-Fr



- ~35M parallel sentences
- 41 BLEU on test
- median freq. in 10% bin: 2500

Match may look better than it is if model shifts probability mass within each of these buckets, let's take a closer look then…

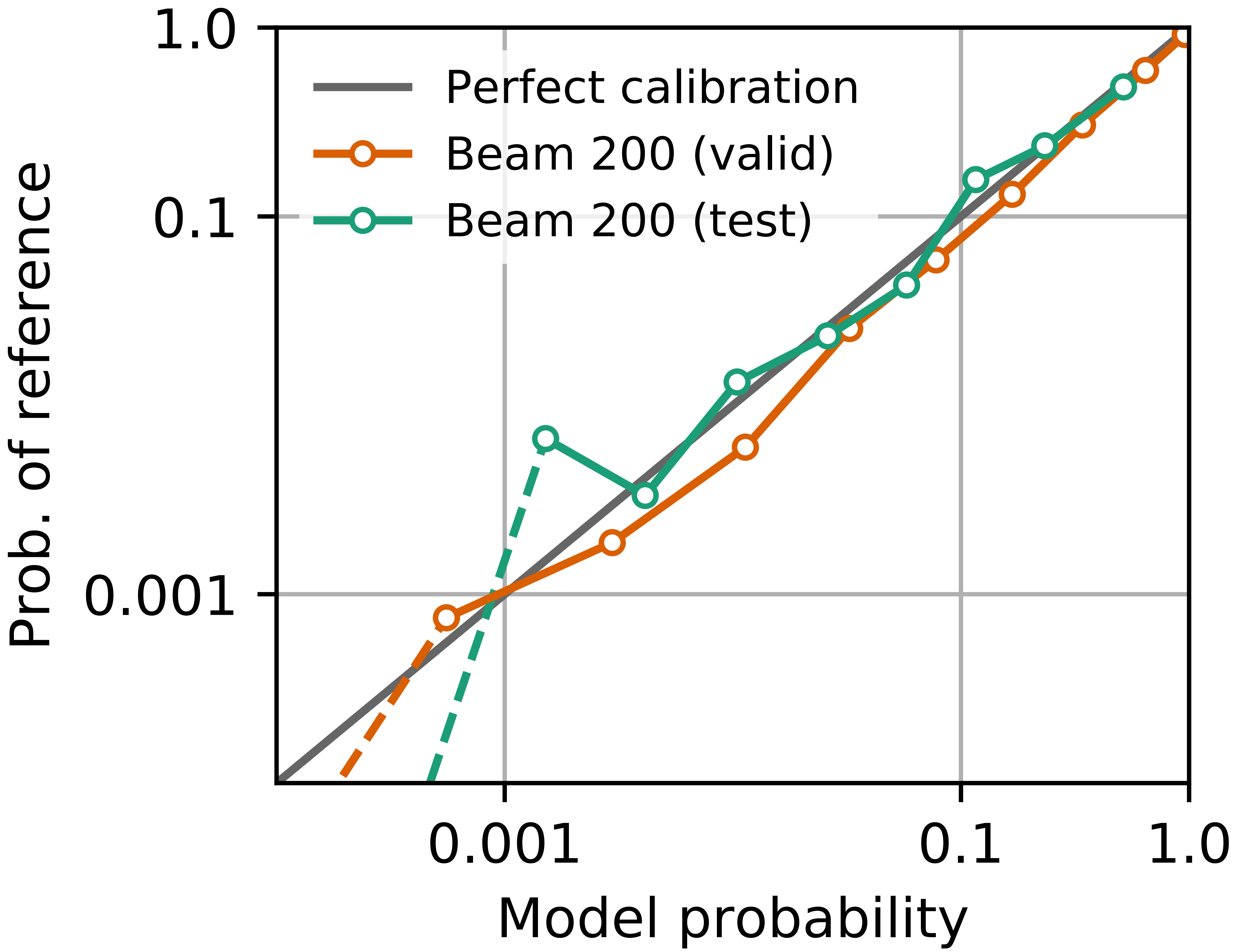
Sentence Level Calibration



Copy source sentences at a given rate during training, check whether probability assigned by the model to copies matches the copy production rate.

NMT model under-estimates copy probability at low rates, while it over-estimates it at high rates.
Model spills probability mass on partial copies.

Set Level Calibration



$$\mathop{\mathbb{E}}_{x\sim p_d}[\mathbb{I}\{x \in S\}] = p_m(S)$$

where S is the set of hypotheses produced by beam.

NMT model is very well calibrated at the set level.

Distance Matching

$$\mathop{\mathbb{E}}_{y \sim p_d, y' \sim p_d}[BLEU(y, y')] \overset{?}{=} \mathop{\mathbb{E}}_{y \sim p_m, y' \sim p_m}[BLEU(y, y')]$$

	En-Fr	**En-De**
human	44.5	32.1
NMT	28.6	24.2

NMT model produces samples that have low BLEU and that are too diverse. Model spreads probability mass.

Distance Matching

$$\mathop{\mathbb{E}}_{y\sim p_d, y'\sim p_d}[BLEU(y,y')] \overset{?}{=} \mathop{\mathbb{E}}_{y\sim p_m, y'\sim p_m}[BLEU(y,y')]$$

	En-Fr	En-De
human	44.5	32.1
NMT	28.6	24.2

FAILED

NMT model produces samples that have low BLEU and that are too diverse. Model spreads probability mass.

Multi-Reference Experiments

	Beam@5	Beam@200	200 Samples
single reference	41.4	36.2	38.2
oracle reference	BLEU with reference yielding the largest BLEU score		
average oracle	average BLEU over all hypothesis of beam/sampling - with closest ref		
coverage	number of unique references using in matching		

We collected 10 additional references for 500 randomly selected source sentences from the test set.

Multi-Reference Experiments

	Beam@5	Beam@200	200 Samples
single reference	41.4	36.2	38.2
oracle reference	70.2	61.0	64.1
average oracle	65.7	56.4	39.1
coverage	1.9	5.0	7.4

Multi-Reference Experiments

	Beam@5	Beam@200	200 Samples
single reference	41.4	36.2	38.2
oracle reference	70.2	61.0	64.1
average oracle	65.7	56.4	39.1
coverage	1.9	5.0	7.4

Beam produces outputs close to an actual reference.
Lower scoring hypotheses are not far from a reference.
However, they often map to the same reference.

Multi-Reference Experiments

	Beam@5	Beam@200	200 Samples
single reference	41.4	36.2	38.2
oracle reference	70.2	61.0	64.1
average oracle	65.7	56.4	39.1
coverage	1.9	5.0	7.4

Sampling is more diverse but several samples poorly match any given reference. Mass is spread too much.

Conclusions

- Uncertainty in data: intrinsic/extrinsic
  - Search: works really well. For large beams, we find spurious modes, but we know how to fix it!
- Model & Data distribution: model is surprisingly well calibrated. In general, it spreads probability mass too much compared to the data distribution.

Collaborators

Sergey Edunov

Myle Ott

Jonas Gehring

Yann Dauphin

David Grangier

Marc'Aurelio Ranzato

Questions?

Come work with us!

Openings for internships, postdocs, research scientists