Sequence to Sequence Learning: Fast Training and Inference with Gated Convolutions

Michael Auli

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

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

Facebook AI Research (FAIR)

Facebook AI Research

Sequence to Sequence Learning

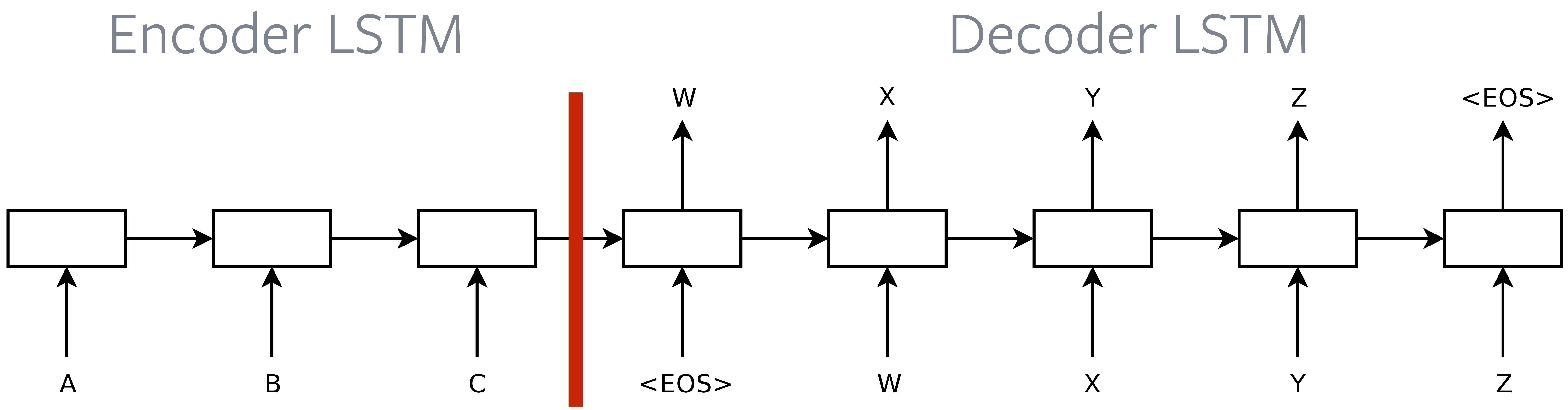


Figure 1: Our model reads an input sentence "ABC" and produces "WXYZ" as the output sentence. The model stops making predictions after outputting the end-of-sentence token. Note that the LSTM reads the input sentence in reverse, because doing so introduces many short term dependencies in the data that 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, ...
- Translation, e.g., ″La maison de Léa.″ -> ″Léa′s house.″

- ″Models basis for 25% of posters at ACL″,
  Lapata at keynote ACL′17

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

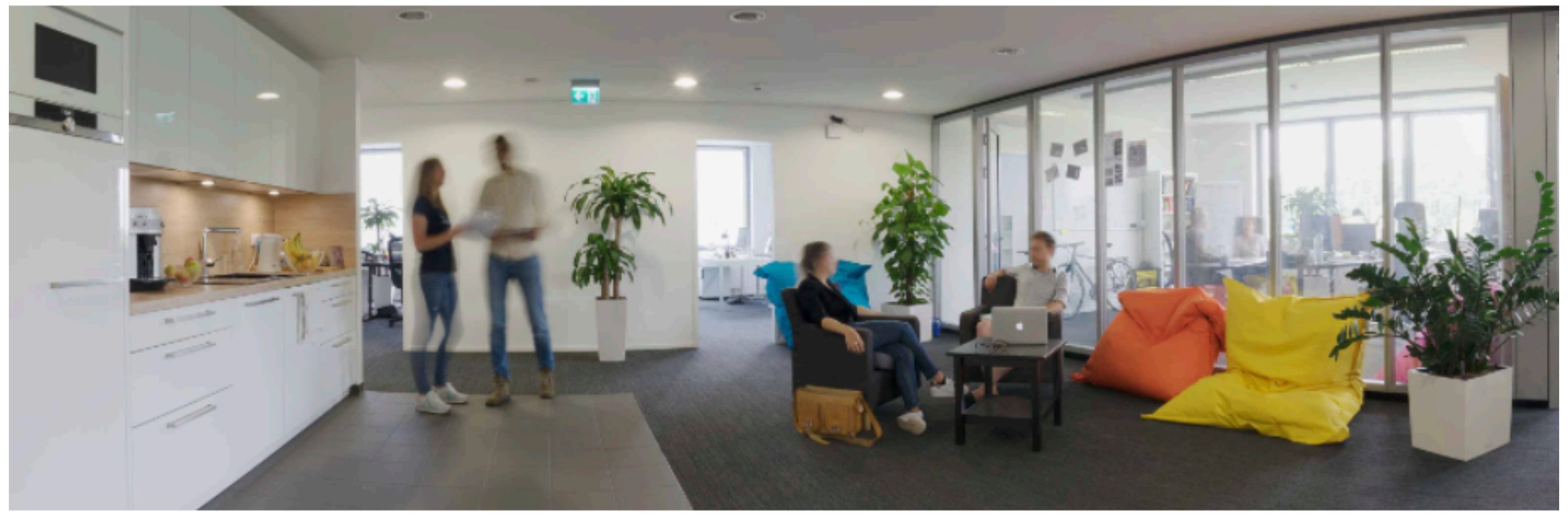
Ashish Vaswani* Google Brain avaswani@google.com
Noam Shazeer* Google Brain noam@google.com
Niki Parmar* Google Research nikip@google.com
Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com
Aidan N. Gomez*† University of Toronto aidan@cs.toronto.edu
Łukasz Kaiser* 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



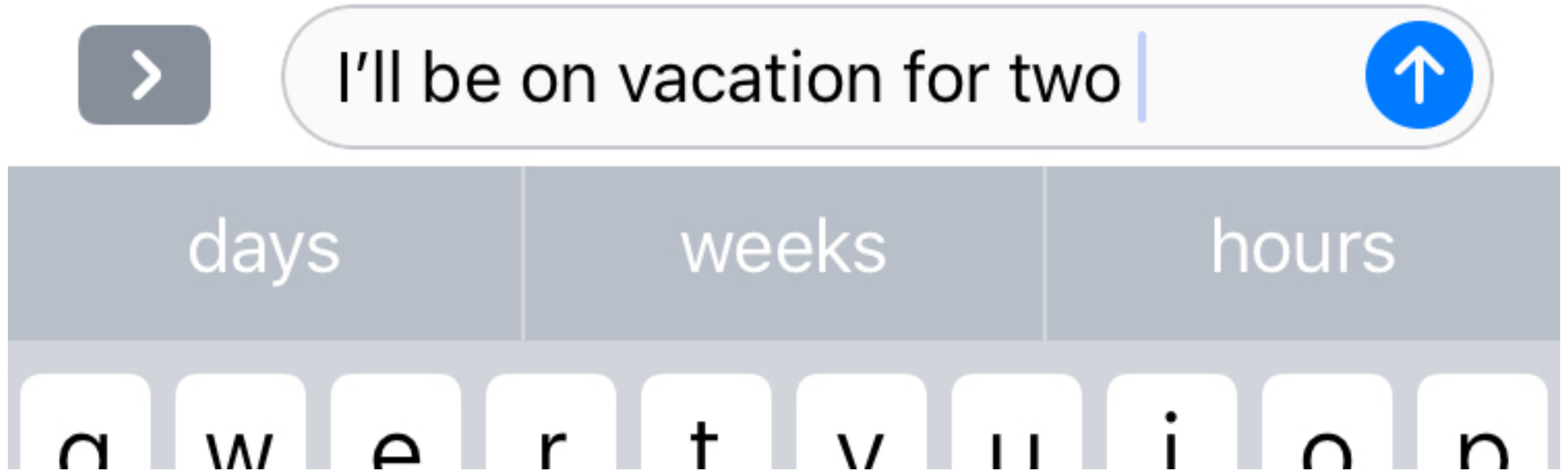
Press Information – DeepL Translator Launch

Overview

- Gated convolutions for Language Modeling
- Convolutional Sequence to Sequence Learning
- Analyzing beam search for seq2seq

Gated Convolutional Models for Language Modeling

Language Modeling



- Estimate probability of a sequence of words

$$P(w_0, \ldots, w_N) = P(w_0) \prod_{i=1}^{N} P(w_i | w_0, \ldots, w_{i-1})$$

- Good language models help in speech (Mikolov et al, 2010) and translation
- LSTMs achieve state-of-the-art performance by processing sentences left to right

CNNs & RNNs

Vision → Convolutional neural networks

NLP/Speech → Recurrent neural networks

CNNs & RNNs

Vision → Convolutional neural networks

NLP/Speech → Recurrent neural networks

- Architectures complex: bi-directional, reverse processing
- Fail to model long-range dependencies in language: need attention

CNNs & RNNs

Vision → Convolutional neural networks

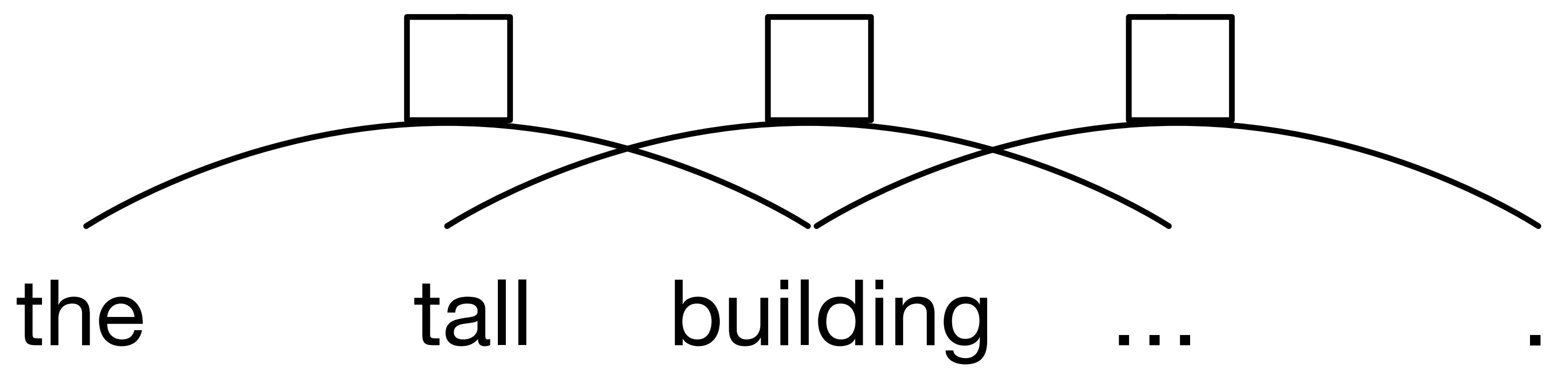
NLP/Speech → Recurrent neural networks

- Architectures complex: bi-directional, reverse processing
- Fail to model long-range dependencies in language: need attention

This talk: **model sequences well without RNNs**

What is a CNN?

- a linear projection taking several input vectors (embeddings, hidden states)
- that maps them to a single output vector (of same or different size)
- which is applied repeatedly to the input sequence at a given stride (=1 here) to yield an output sequence

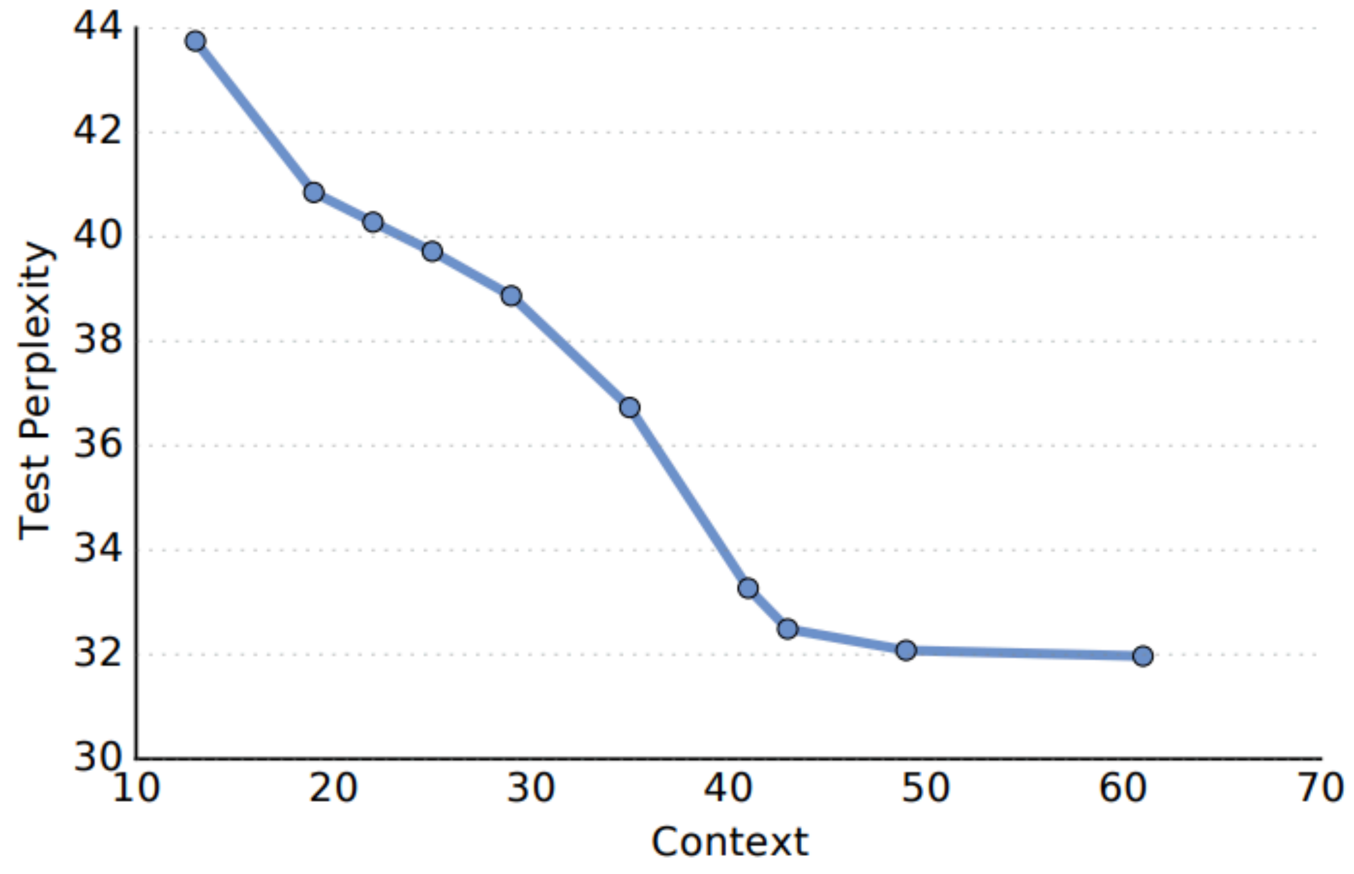


CNNs for Sequence Modeling

- **Hierarchical:** bottom-up vs. left-right
- **Homogeneous:** all elements processed in same way
- **Efficient:** parallelizable over number of sequences **& time** dimension

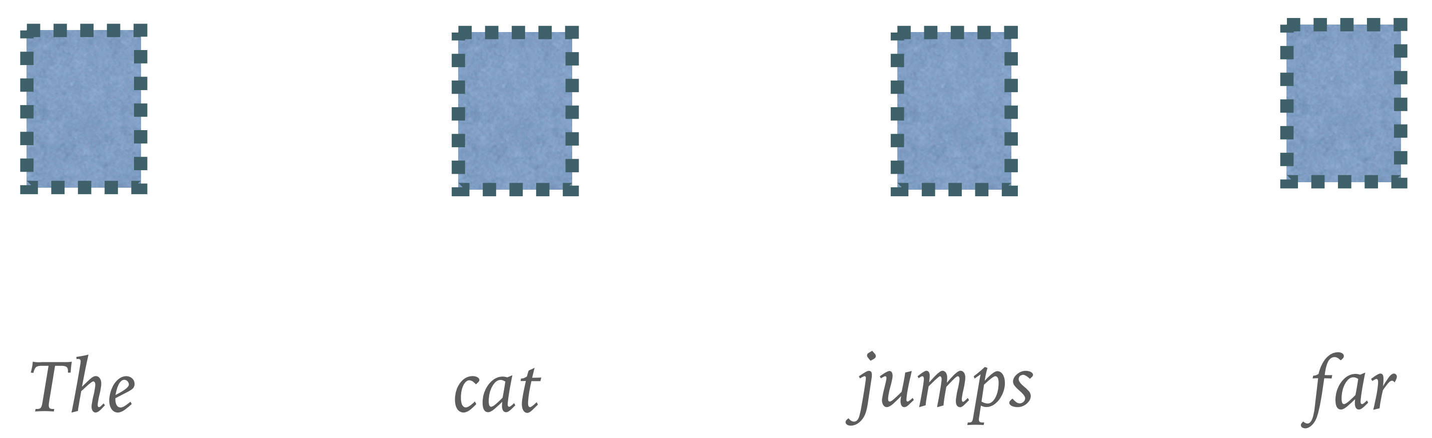
CNNs for Sequence Modeling

- **In practice:**
  - dependencies are not arbitrarily long
  - e.g. Dauphin et al. ICML'17

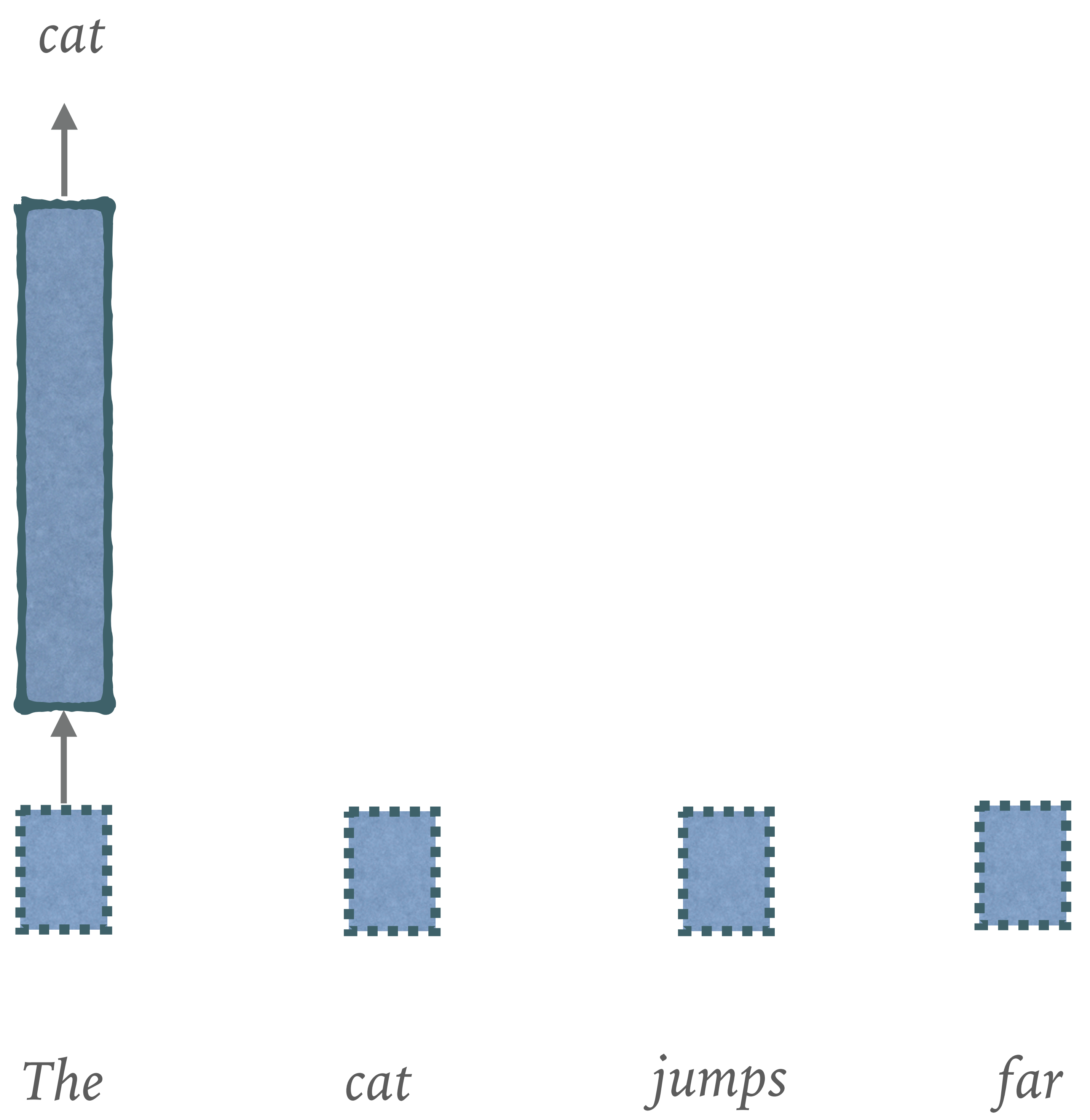


  - CNNs are much more efficient than LSTMs on GPU
  - e.g. Baidu DeepBench, github.com/baidu-research/DeepBench '16

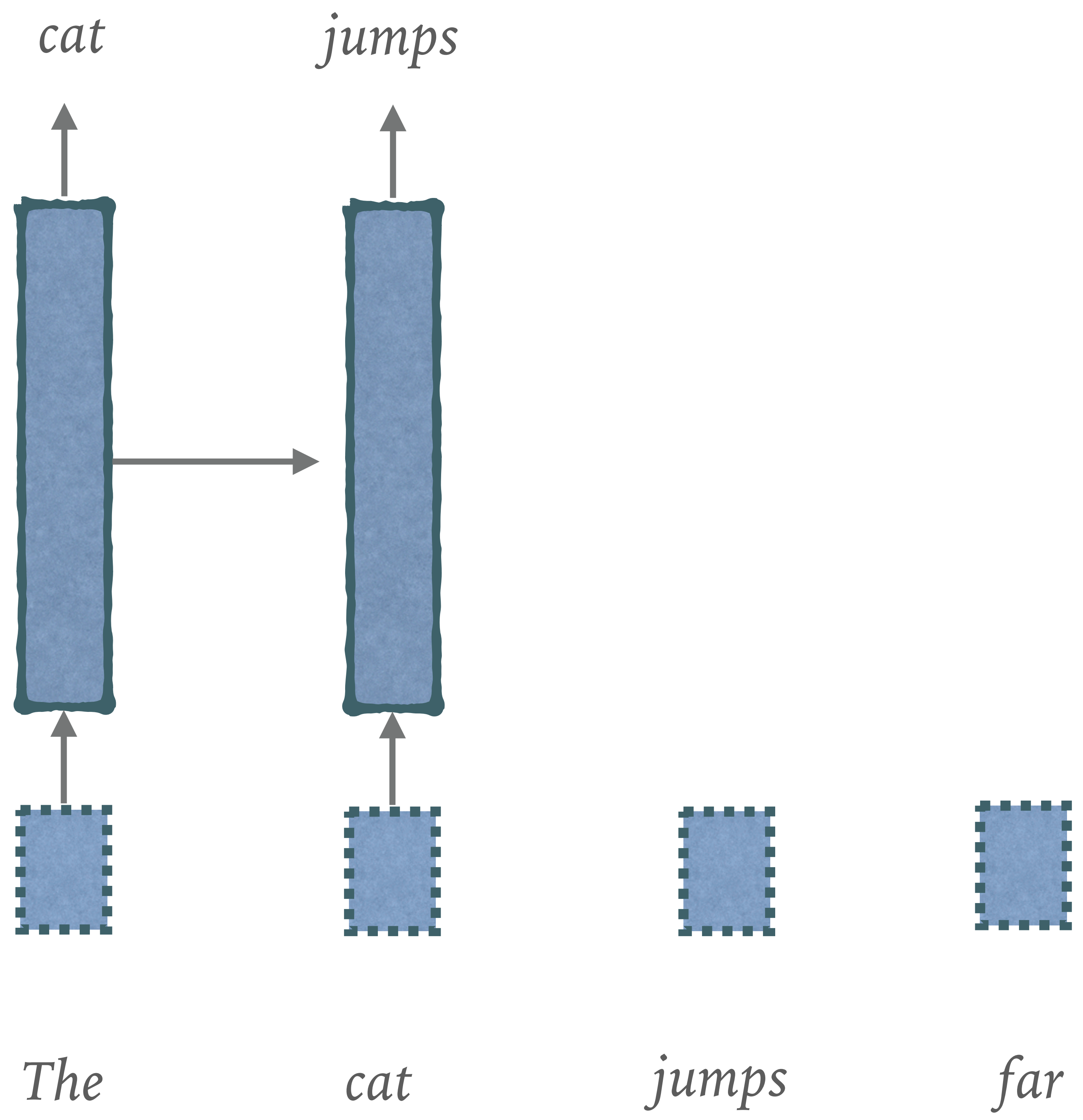
Recurrent Neural Network



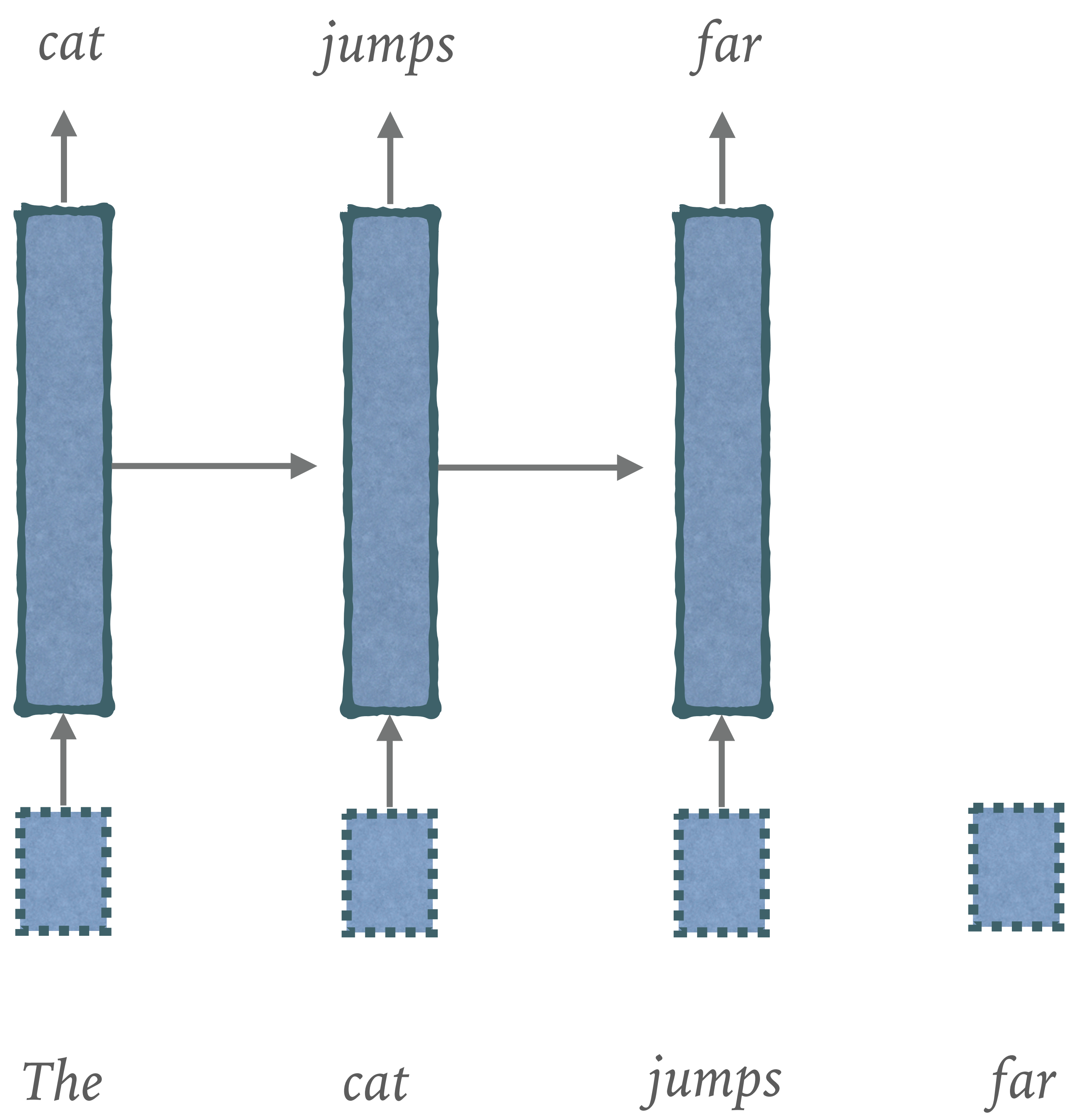
Recurrent Neural Network



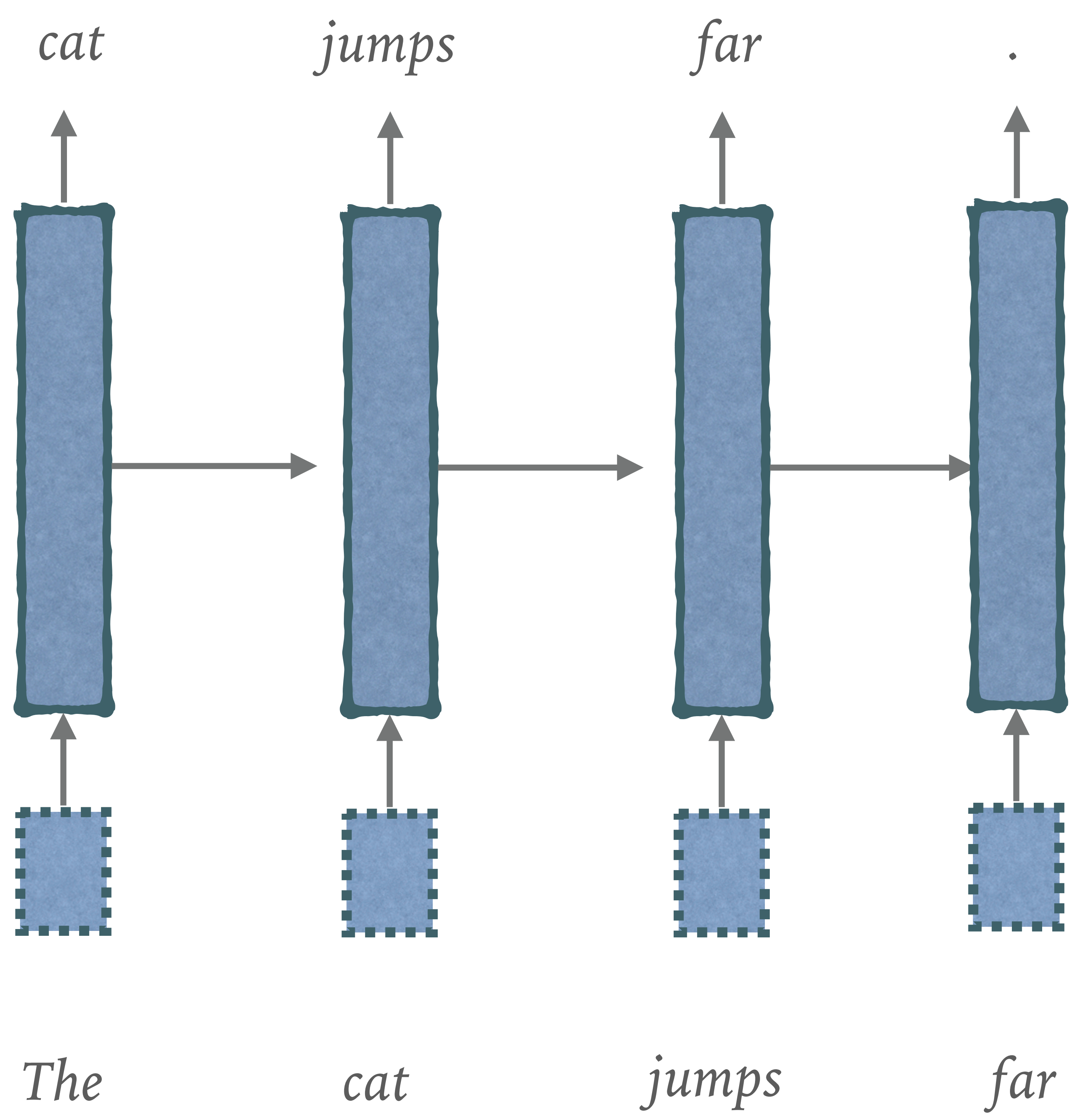
Recurrent Neural Network



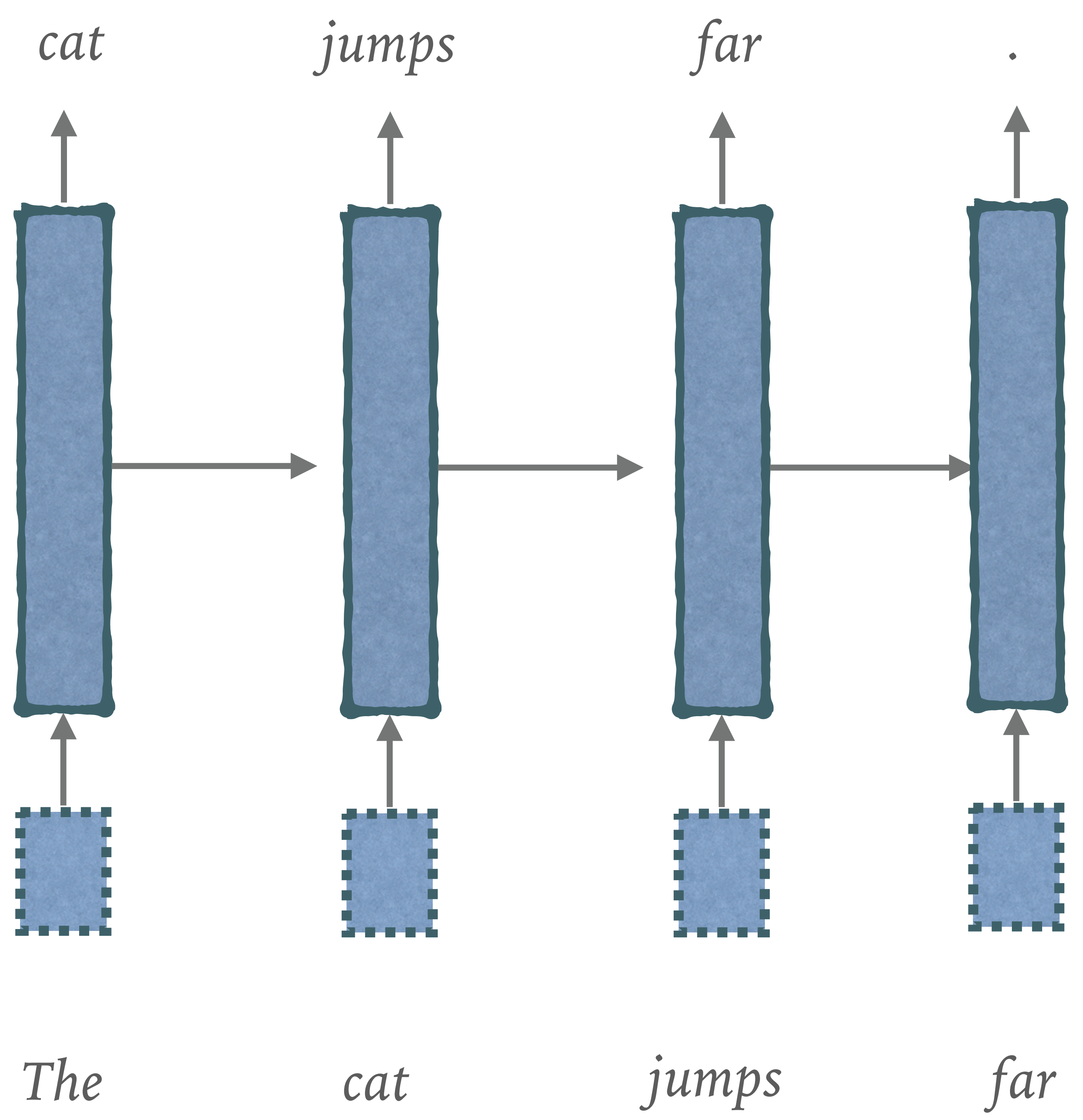
Recurrent Neural Network



Recurrent Neural Network

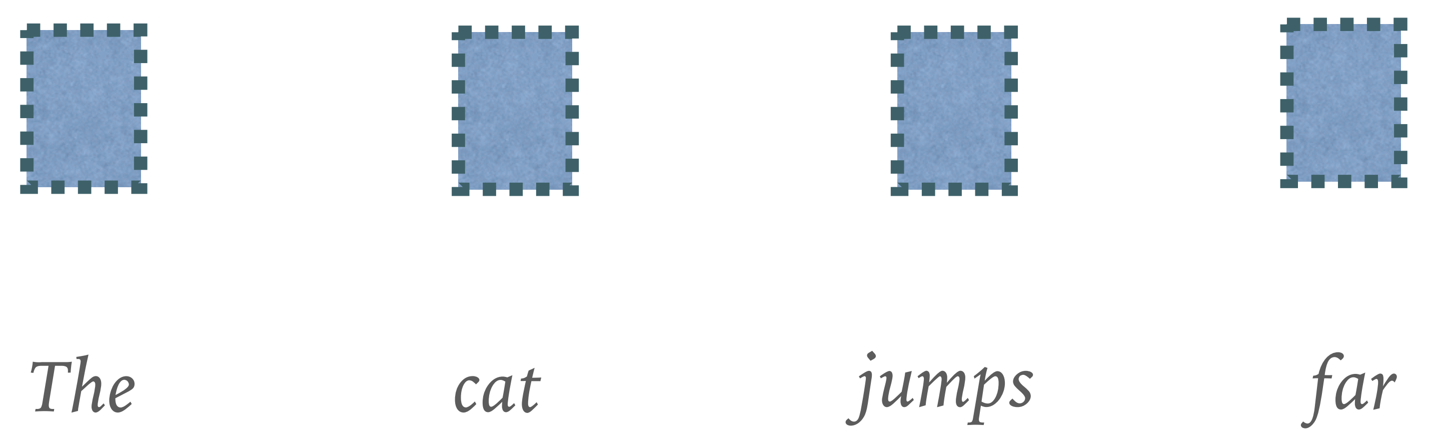


Recurrent Neural Network

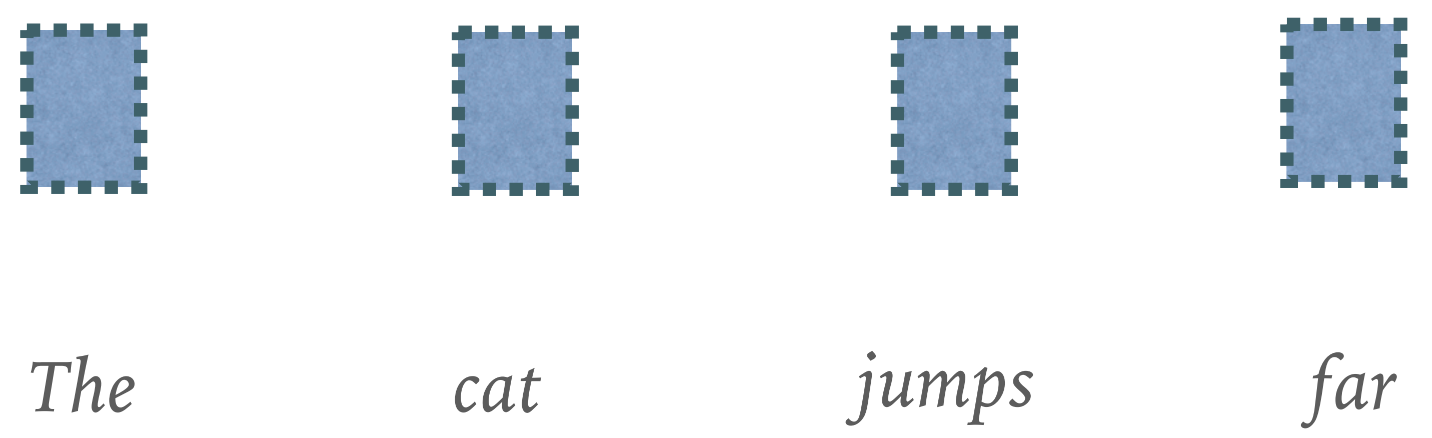


- O(T) sequential steps
- Recurrent connection causes vanishing gradient
- Are the recurrent connections necessary?

Multi-Layer Perceptron

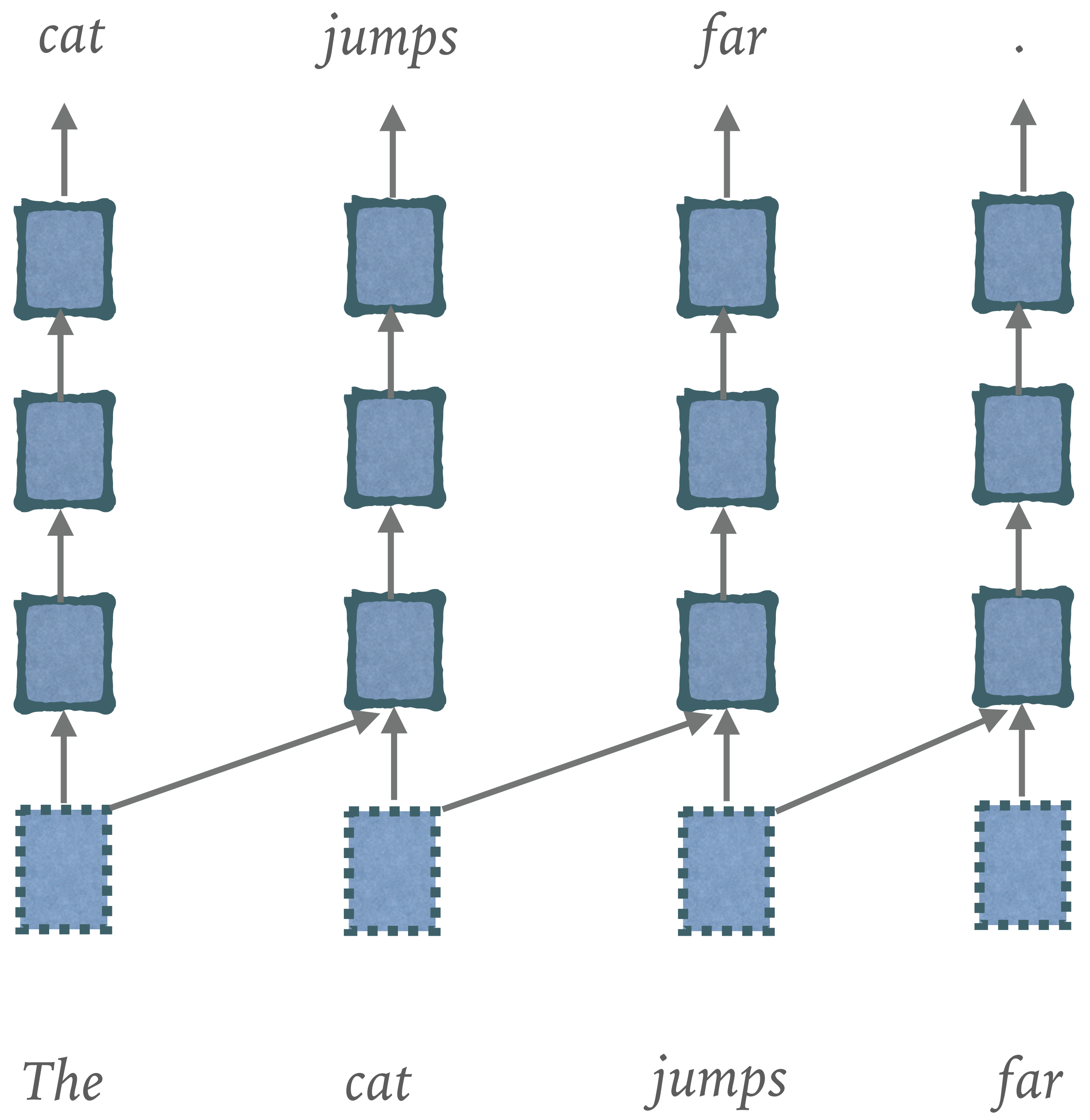


Multi-Layer Perceptron



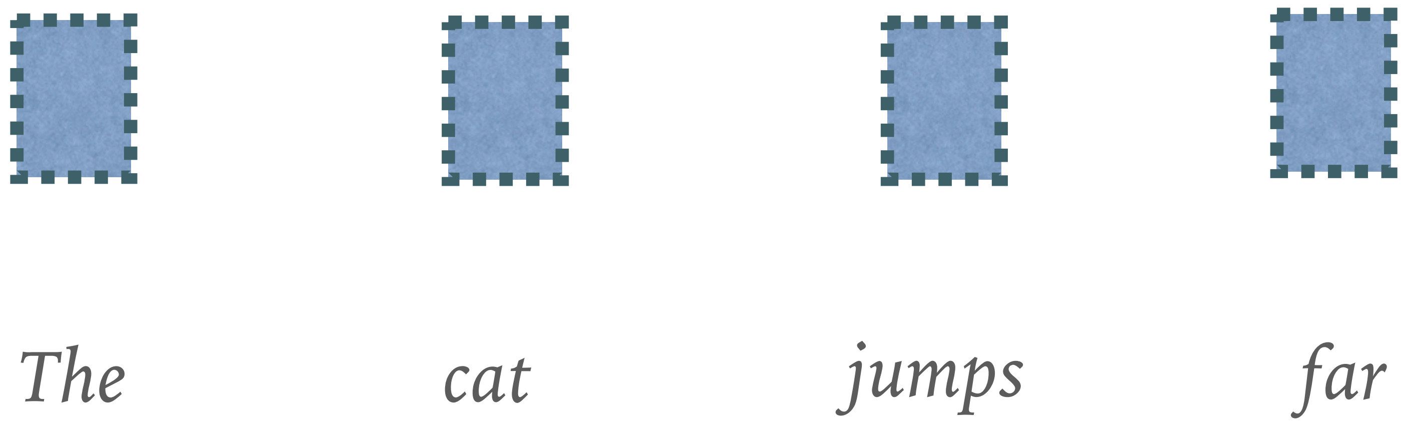
- O(1) sequential steps
- Proposed by (Bengio et al, 2001)
- Inefficient because no computation is shared between time steps
- Bad experimental results

Multi-Layer Perceptron

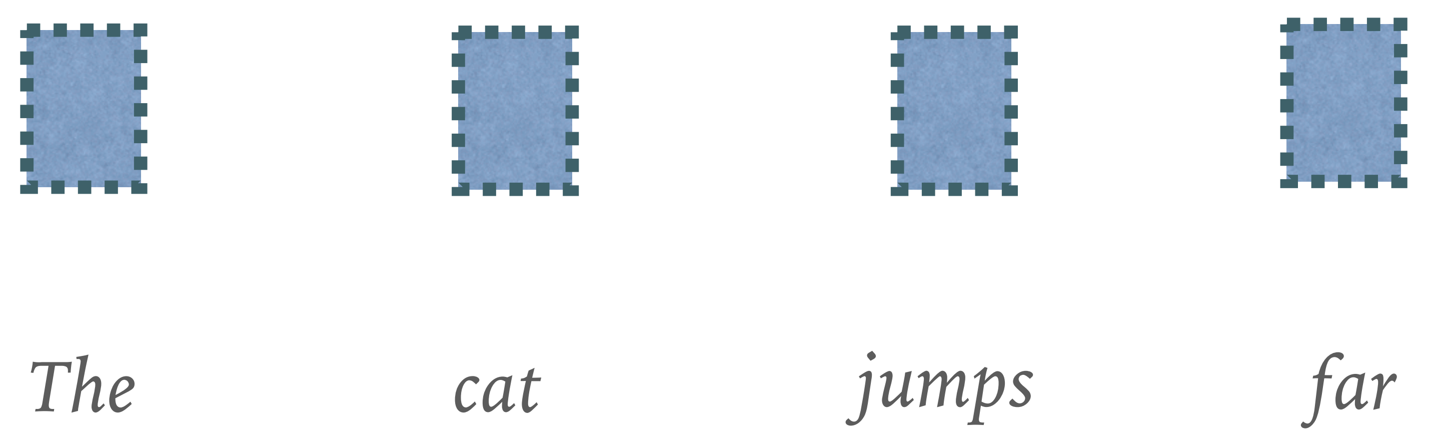


- O(1) sequential steps
- Proposed by (Bengio et al, 2001)
- Inefficient because no computation is shared between time steps
- Bad experimental results

Convolutional Neural Network

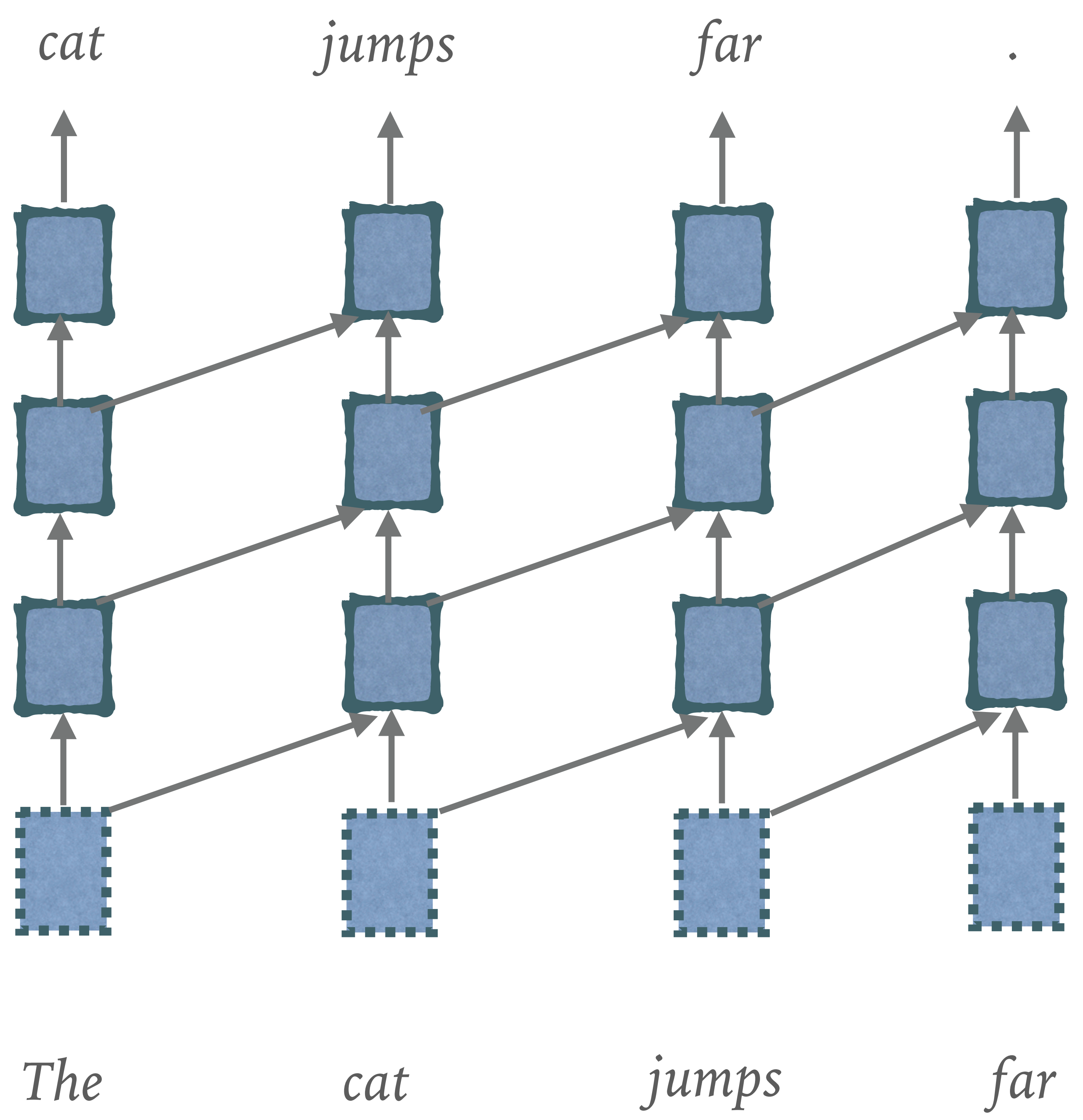


Convolutional Neural Network



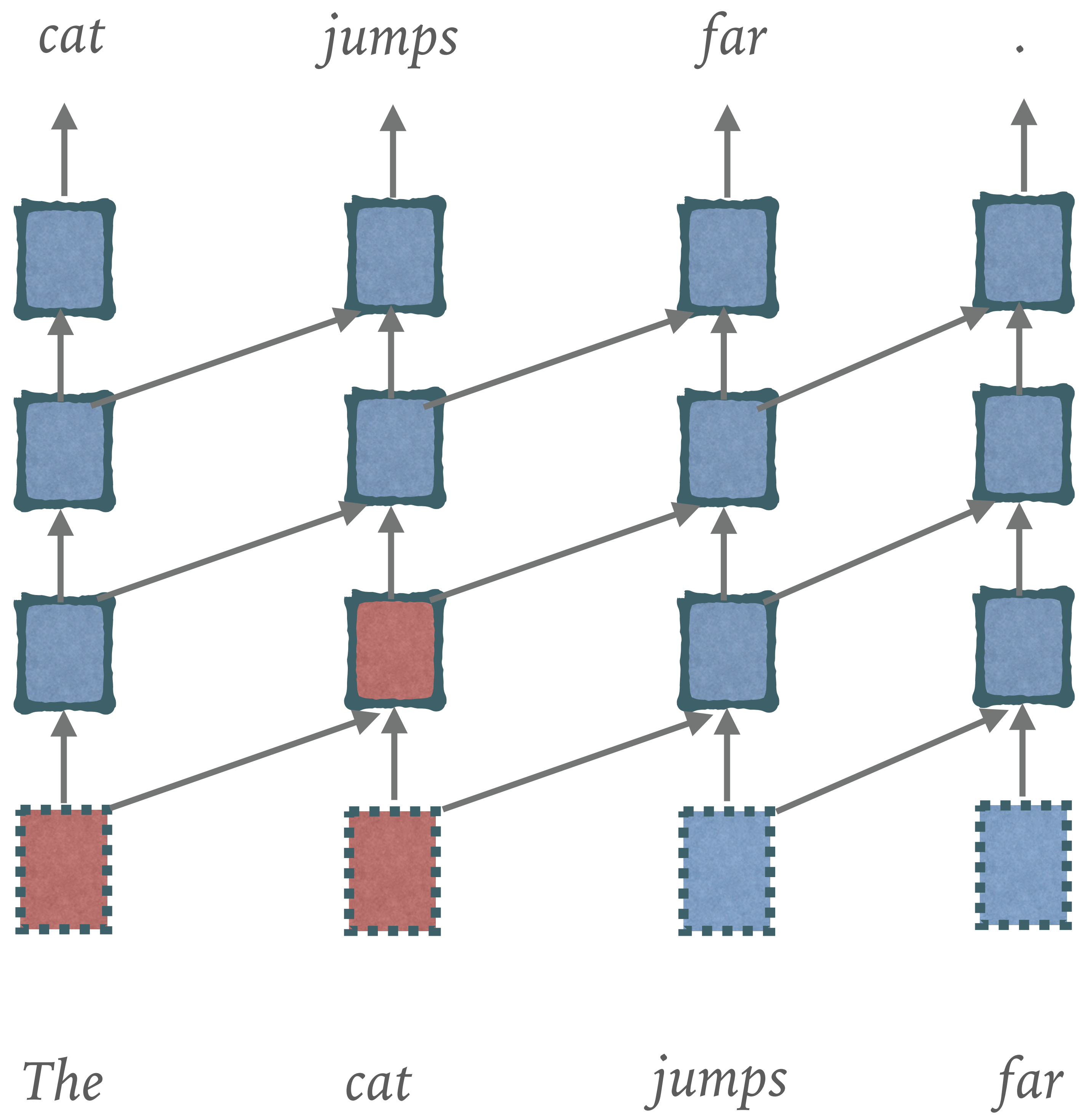
- O(1) sequential steps
- Incrementally build context of context windows

Convolutional Neural Network



- O(1) sequential steps
- Incrementally build context of context windows

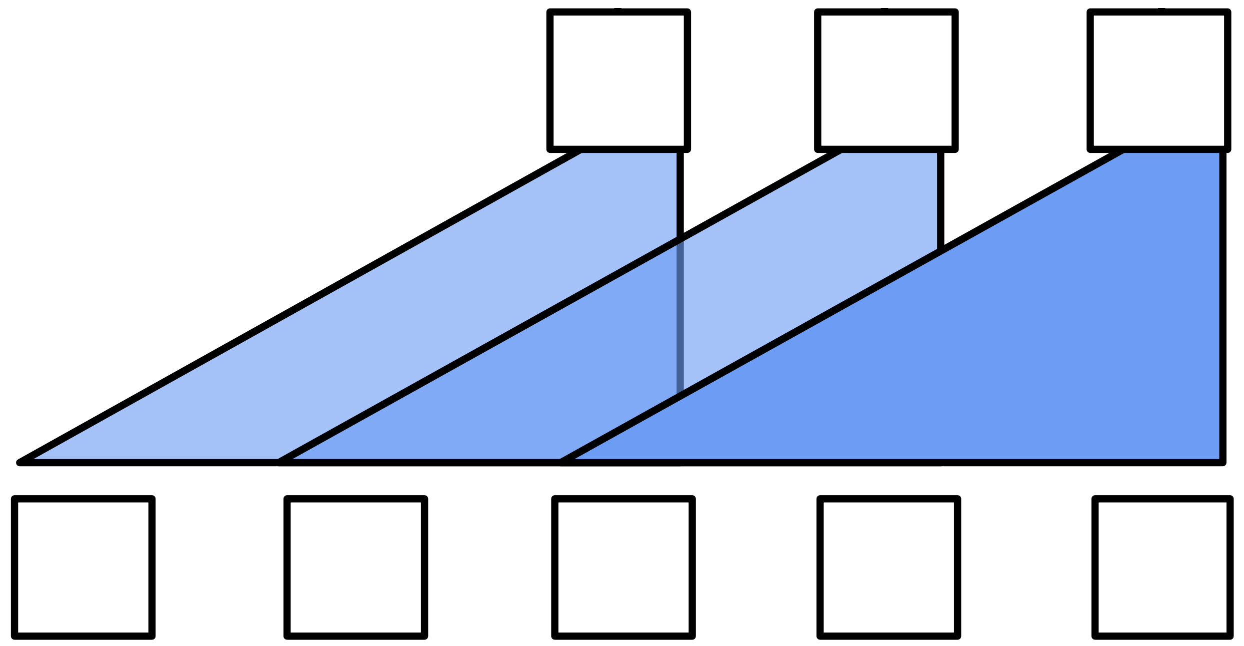
Convolutional Neural Network



- O(1) sequential steps
- Incrementally build context of context windows
- Builds **hierarchical** structure

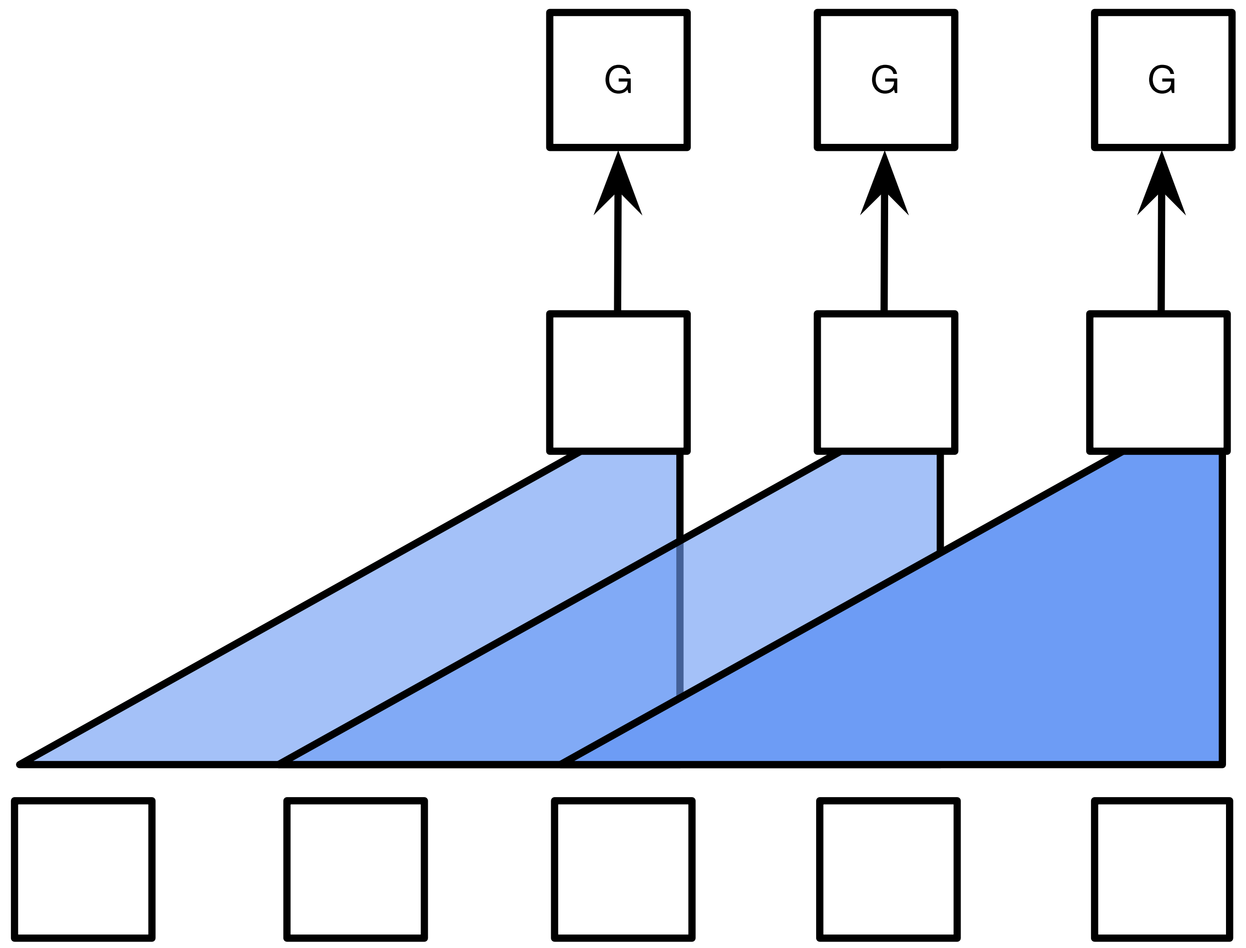
Gated Convolutional Neural Network

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



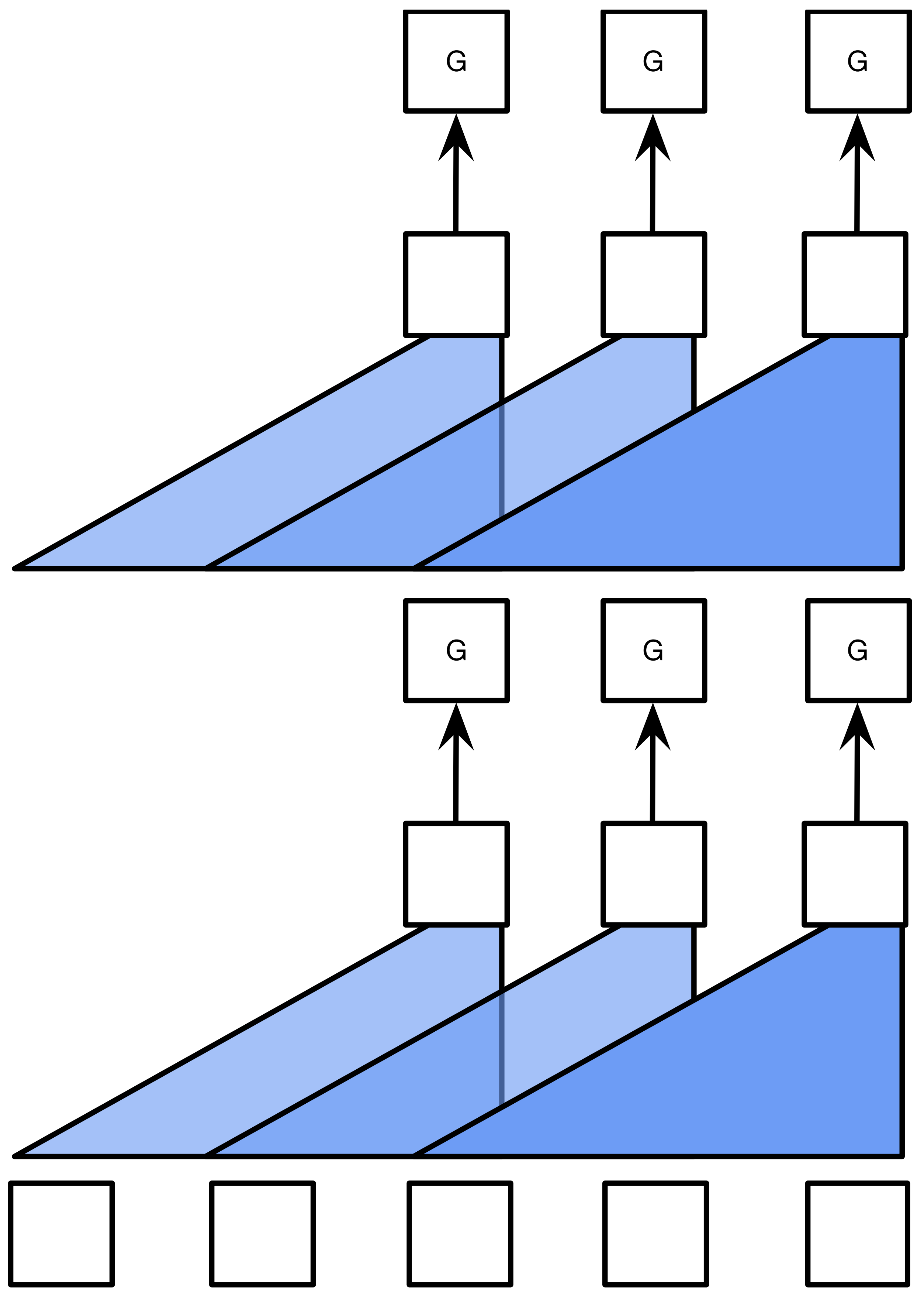
Gated Convolutional Neural Network

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



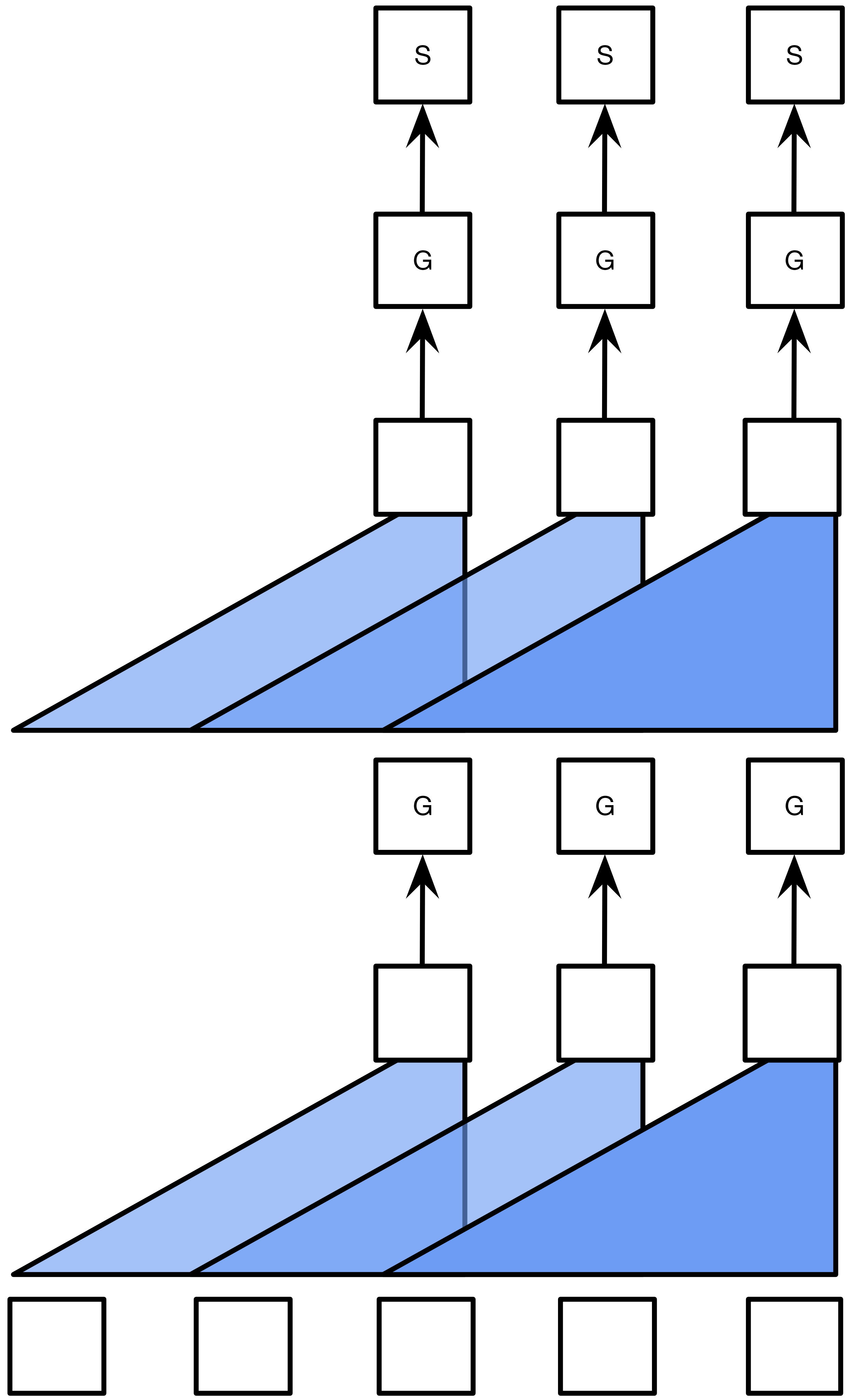
Gated Convolutional Neural Network

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



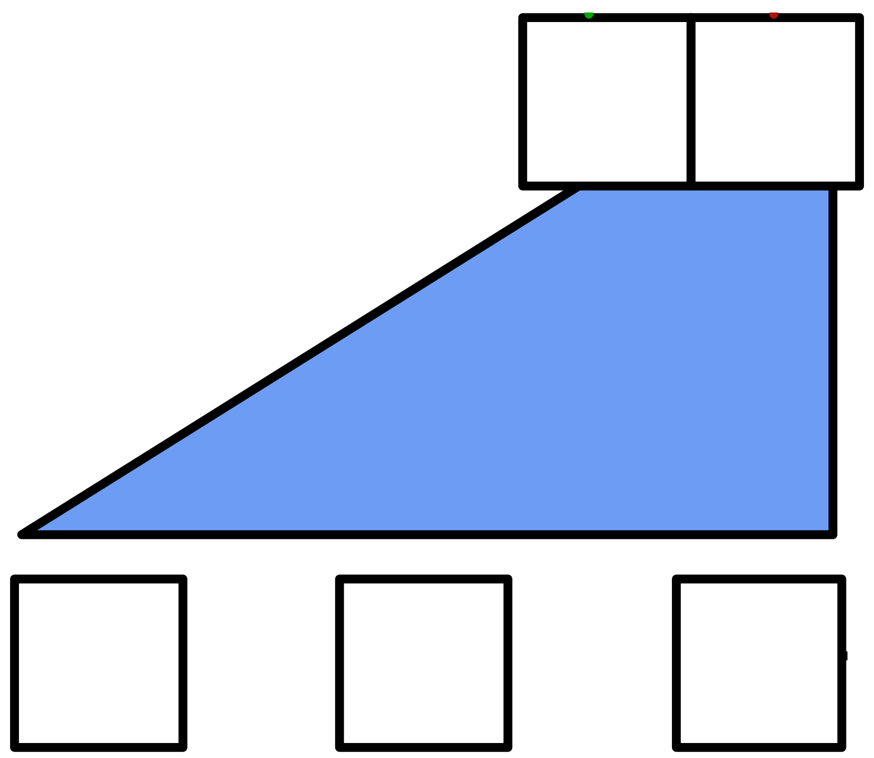
Gated Convolutional Neural Network

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



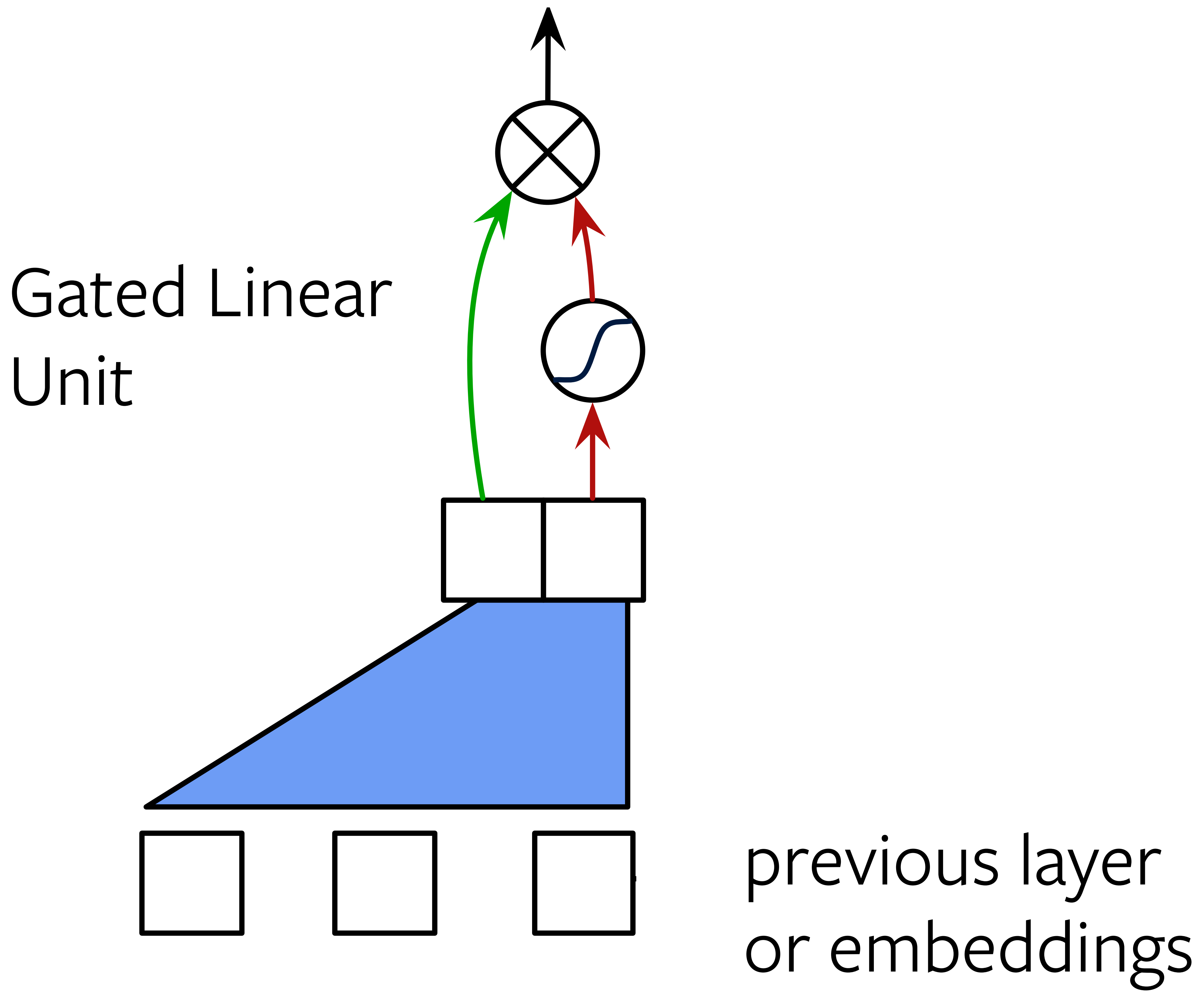
Gated Linear Unit

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



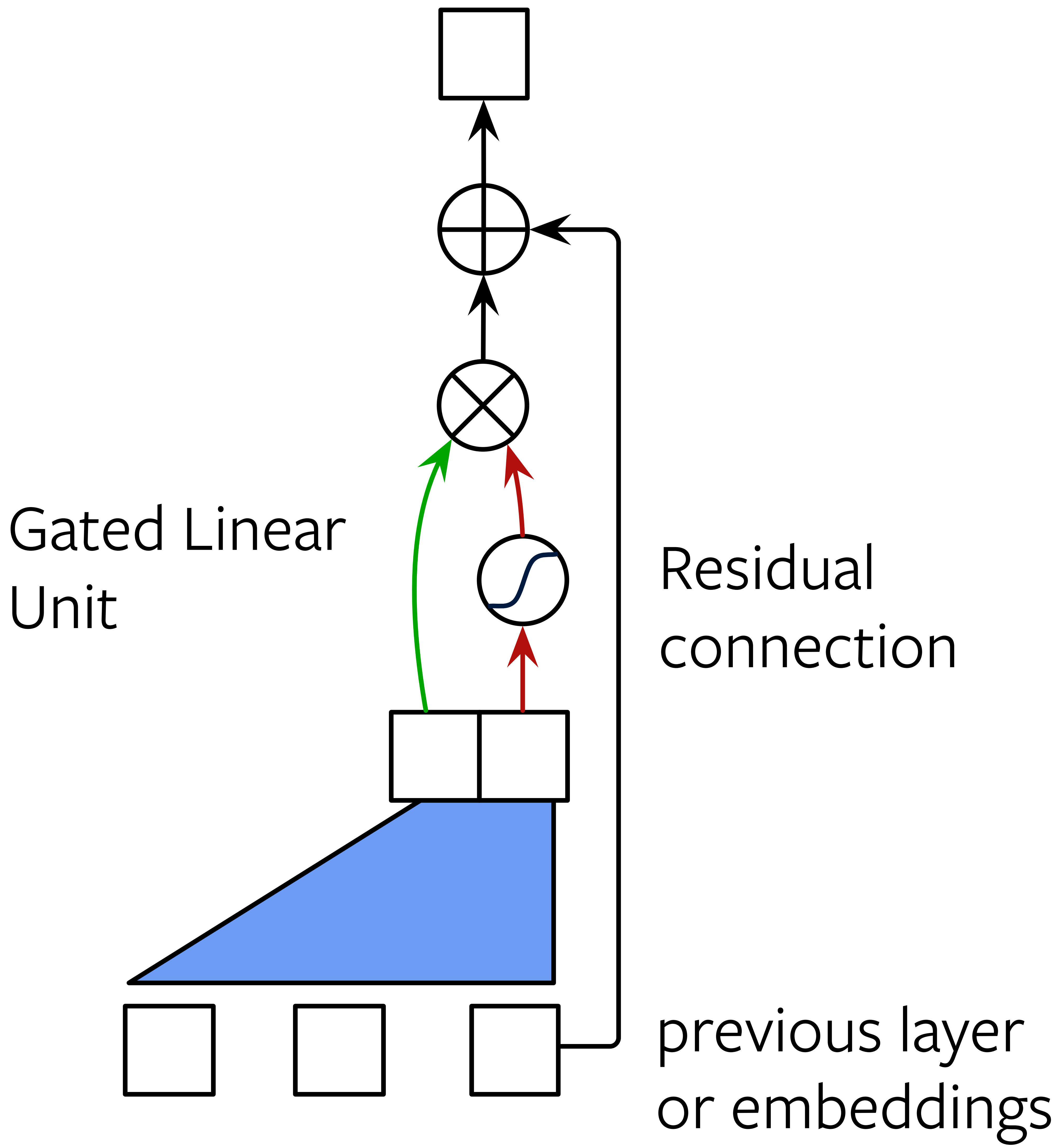
Gated Linear Unit

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



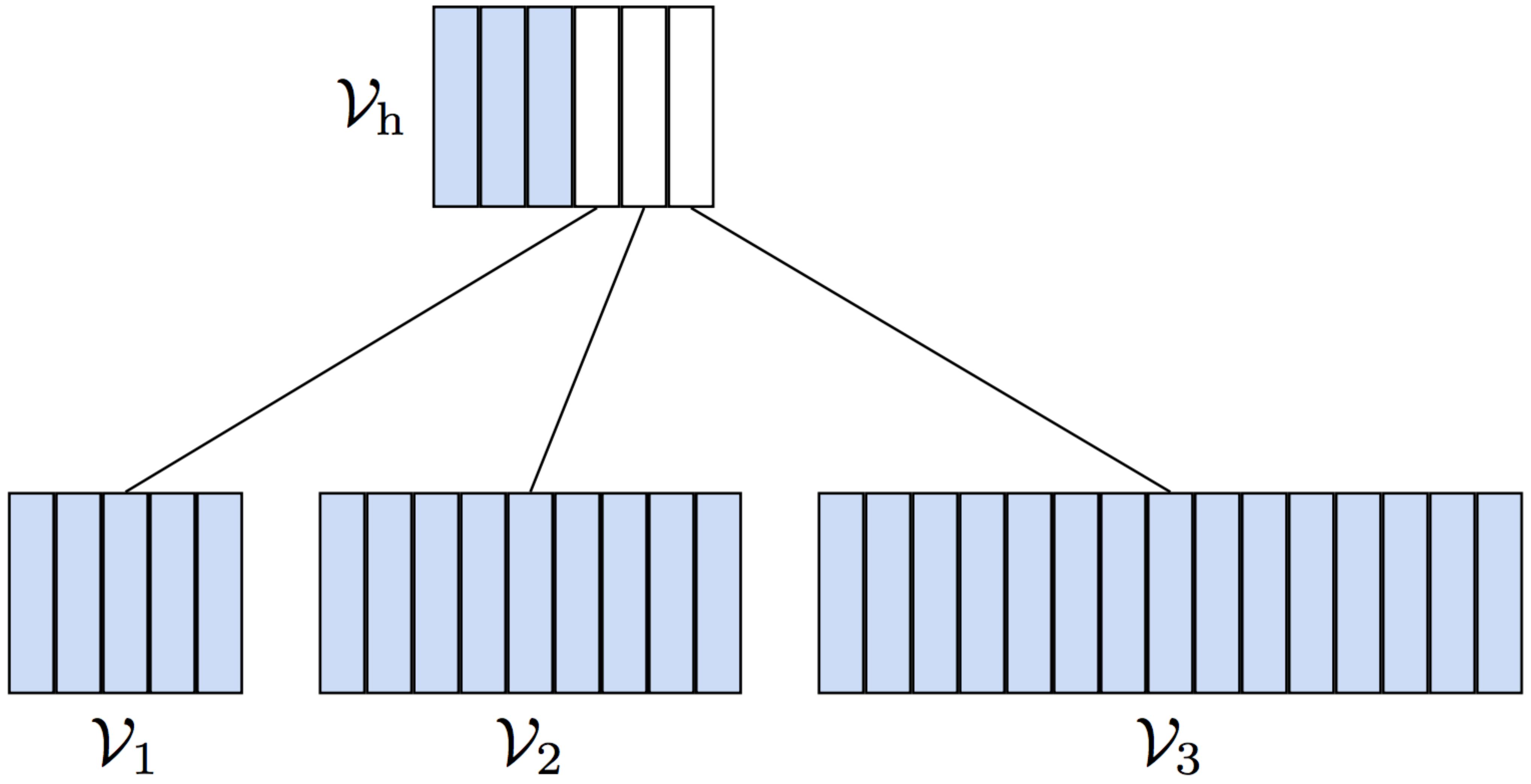
Gated Linear Unit

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



Training

- We use SGD with Nesterov's momentum and weight normalization (Salimans & Kingma, 2016)
- Clipping for convnets (Pascanu et al. 2013)
- Adaptive Softmax (Grave et al, 2016) for very large vocabularies



Datasets

- **Google billion words** (Chelba et al, 2013):
  - ~800k vocabulary with ~800M tokens
  - independent sentences (~20 tokens)

- **WikiText-103** (Bradbury et al, 2016)
  - ~200k vocabulary with ~100M tokens
  - wikipedia articles (~4000 tokens)

Results: Google billion words

Model	Test PPL
Sigmoid-RNN-2048 (Ji et al., 2015)	68.3
Interpolated KN 5-Gram (Chelba et al., 2013)	67.6
Sparse Non-Negative Matrix LM (Shazeer et al., 2014)	52.9
RNN-1024 + MaxEnt 9 Gram Features (Chelba et al., 2013)	51.3
LSTM-2048-512 (Jozefowicz et al., 2016)	43.7
2-layer LSTM-8192-1024 (Jozefowicz et al., 2016)	30.6
LSTM-2048 (Grave et al., 2016a)	43.9
2-layer LSTM-2048 (Grave et al., 2016a)	39.8
GCNN-13	38.1
GCNN-14 Bottleneck	31.9

- GatedCNN manages to match the LSTM with comparable output approximation and computational budget for training

Results: Google billion words

Model	Test PPL
Sigmoid-RNN-2048 (Ji et al., 2015)	68.3
Interpolated KN 5-Gram (Chelba et al., 2013)	67.6
Sparse Non-Negative Matrix LM (Shazeer et al., 2014)	52.9
RNN-1024 + MaxEnt 9 Gram Features (Chelba et al., 2013)	51.3
LSTM-2048-512 (Jozefowicz et al., 2016)	43.7
2-layer LSTM-8192-1024 (Jozefowicz et al., 2016)	30.6
LSTM-2048 (Grave et al., 2016a)	43.9
2-layer LSTM-2048 (Grave et al., 2016a)	39.8
GCNN-13	38.1
GCNN-14 Bottleneck	31.9

- GatedCNN manages to match the LSTM with comparable output approximation and computational budget for training

Results: Google billion words

Model	Test PPL
Sigmoid-RNN-2048 (Ji et al., 2015)	68.3
Interpolated KN 5-Gram (Chelba et al., 2013)	67.6
Sparse Non-Negative Matrix LM (Shazeer et al., 2014)	52.9
RNN-1024 + MaxEnt 9 Gram Features (Chelba et al., 2013)	51.3
LSTM-2048-512 (Jozefowicz et al., 2016)	43.7
2-layer LSTM-8192-1024 (Jozefowicz et al., 2016)	30.6
BIG GLSTM-G4 (Kuchaiev & Ginsburg, 2017)	23.3*
LSTM-2048 (Grave et al., 2016a)	43.9
2-layer LSTM-2048 (Grave et al., 2016a)	39.8
GCNN-13	38.1
GCNN-14 Bottleneck	31.9

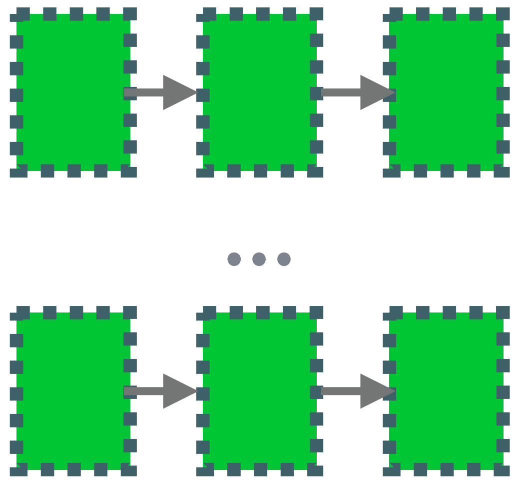
- GatedCNN manages to match the LSTM with comparable output approximation and computational budget for training

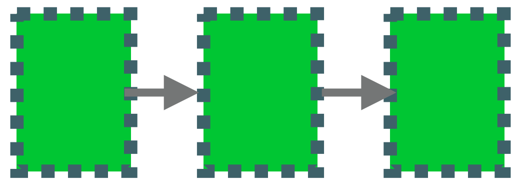
Results: Wikitext-103

Model	Test PPL
LSTM-1024 (Grave et al., 2016b)	48.7
GCNN-8	44.9
GCNN-14	37.2

- SOTA accuracy despite limited context size (25 & 32 words)

Speed

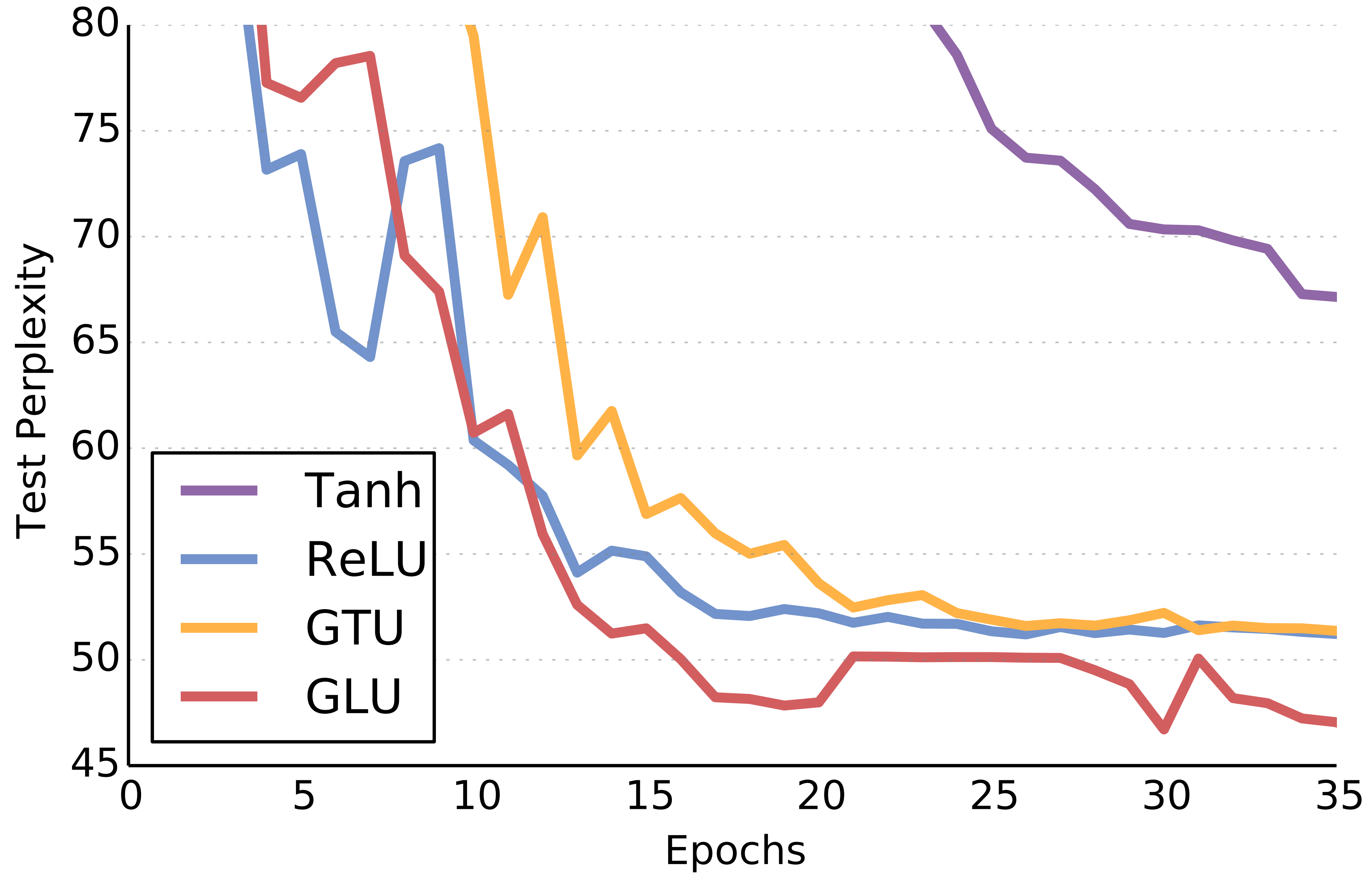




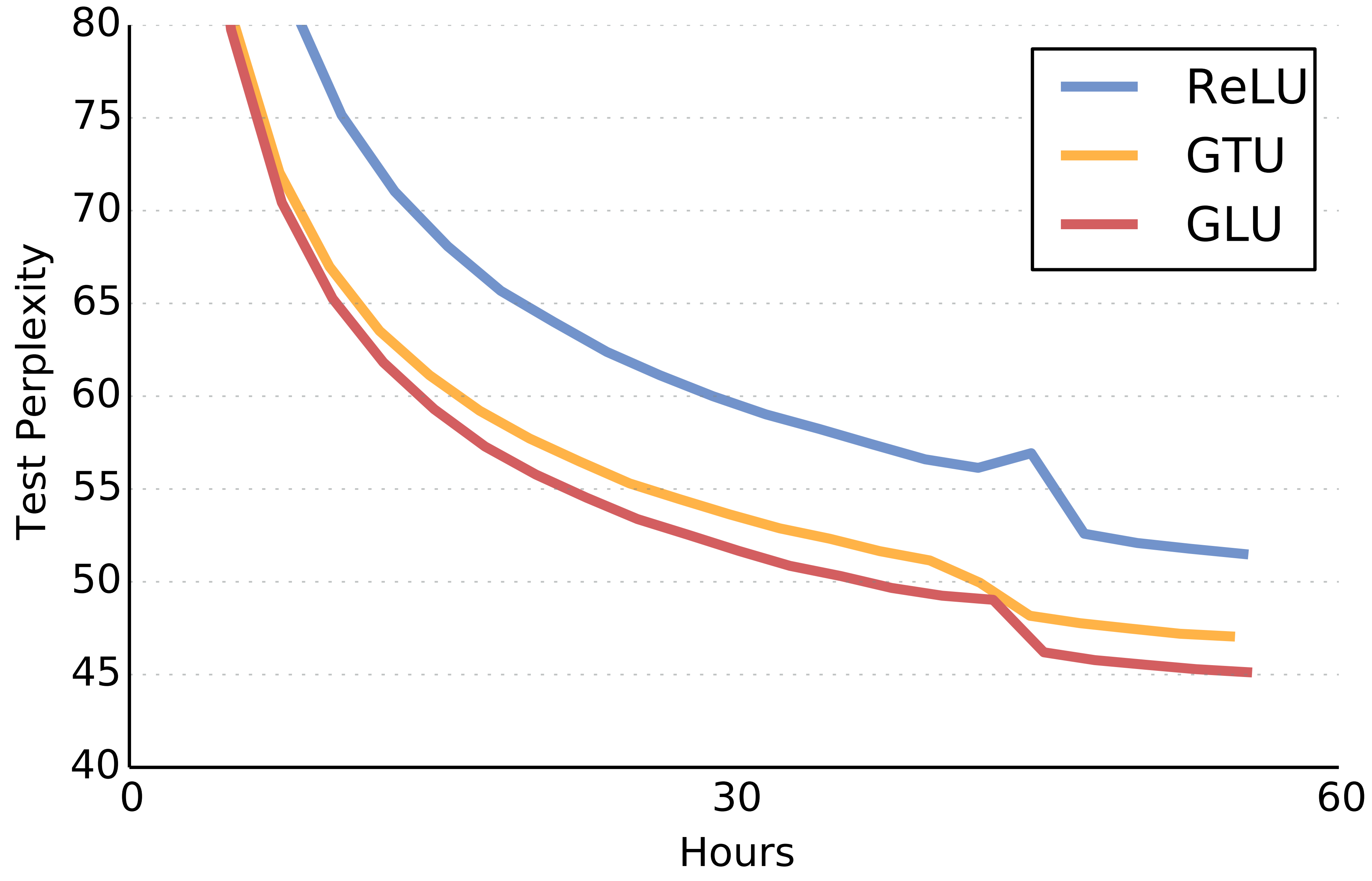
	Throughput (CPU)	Throughput (GPU)	Responsiveness (GPU)
LSTM-2048	169	45,622	2,282
GCNN-9	121	29,116	29,116
GCNN-8 Bottleneck	**179**	**45,878**	**45,878**

- Throughput is the number of tokens per second
- Responsiveness is the number of sequential tokens per second

Gating



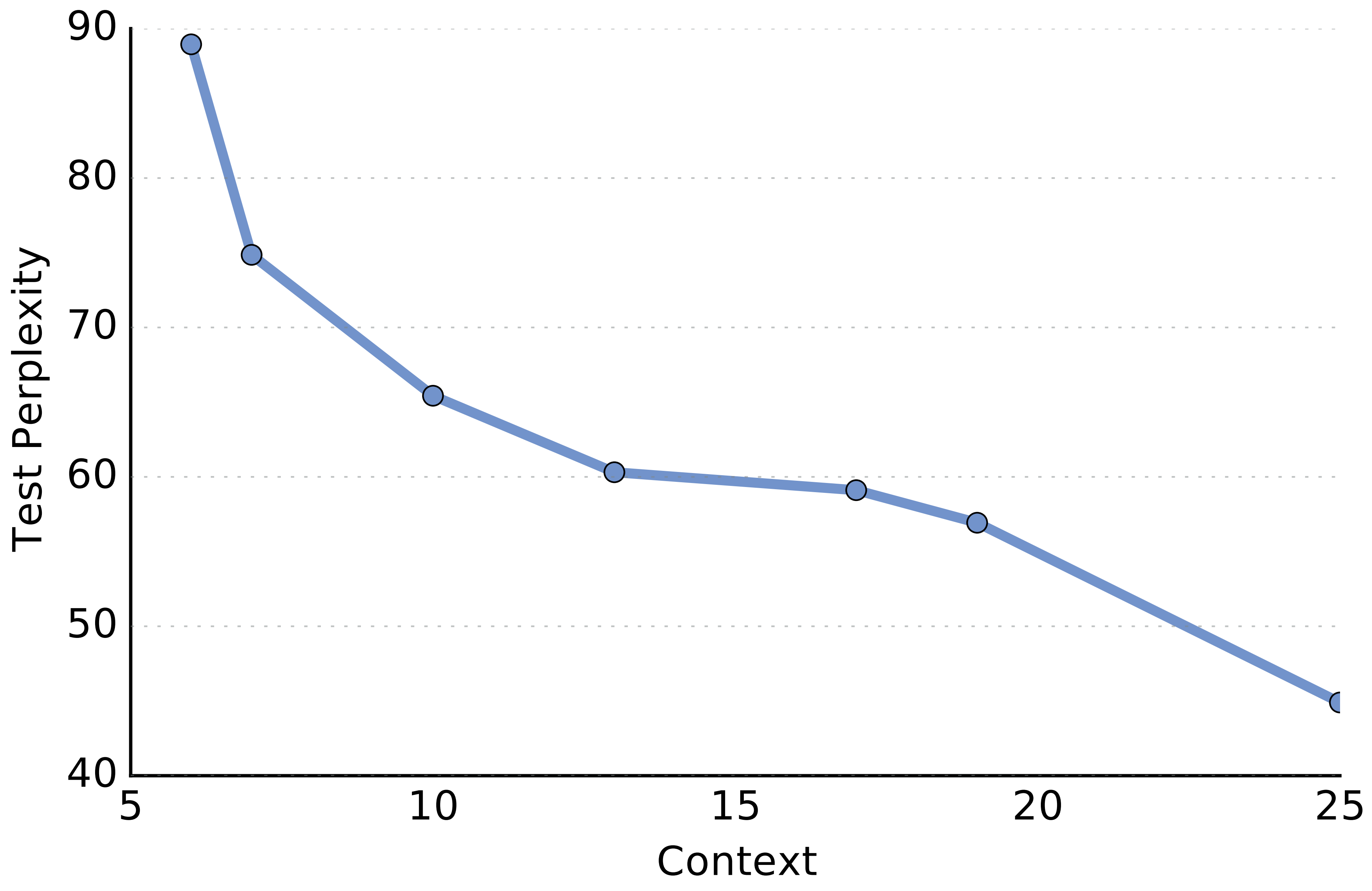
WikiText-103



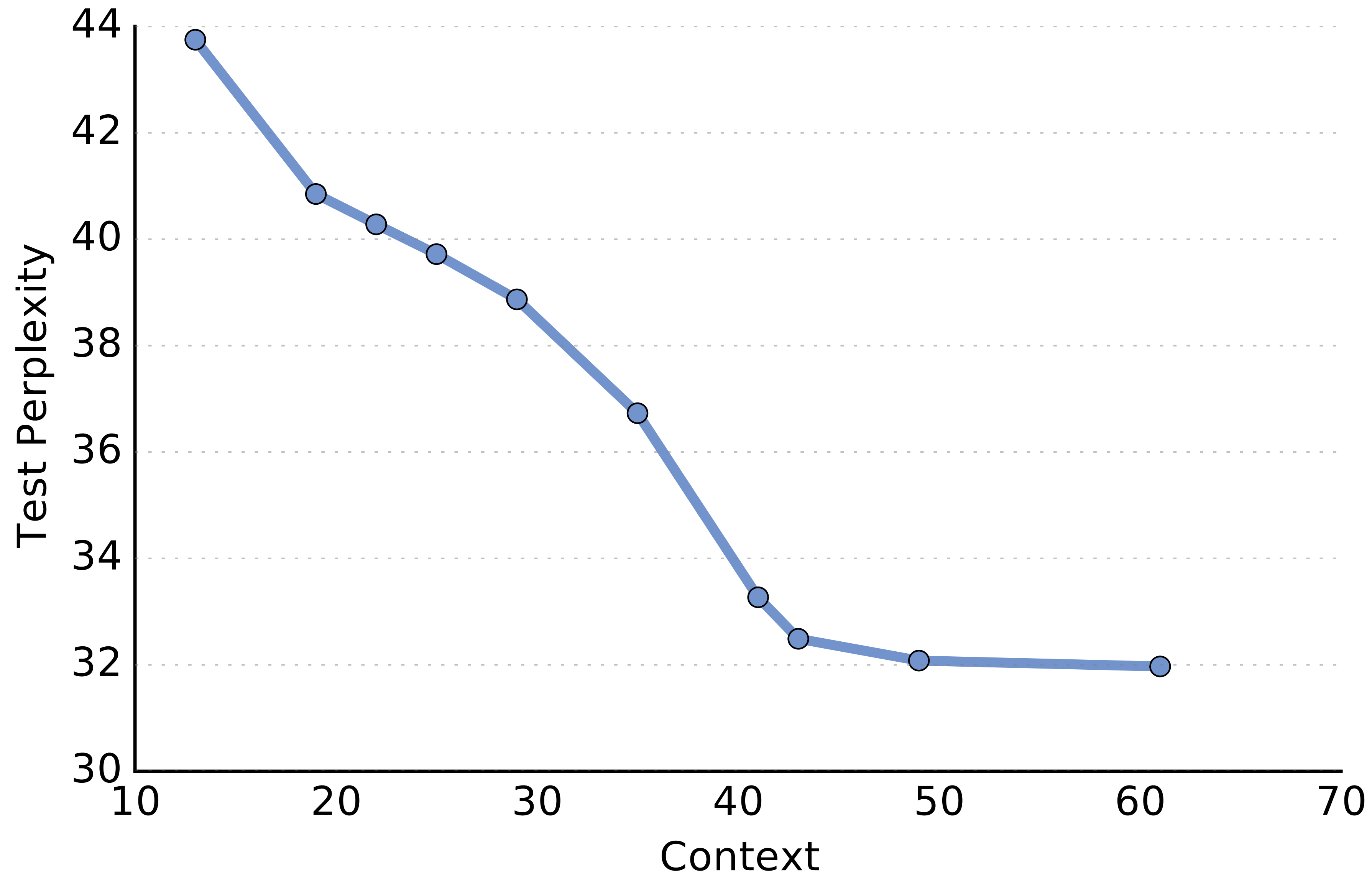
Google Billion Words

- Gated linear units (GLU in red) converge faster
- GTU is LSTM style gating of (Oord et al, 2016)

Context



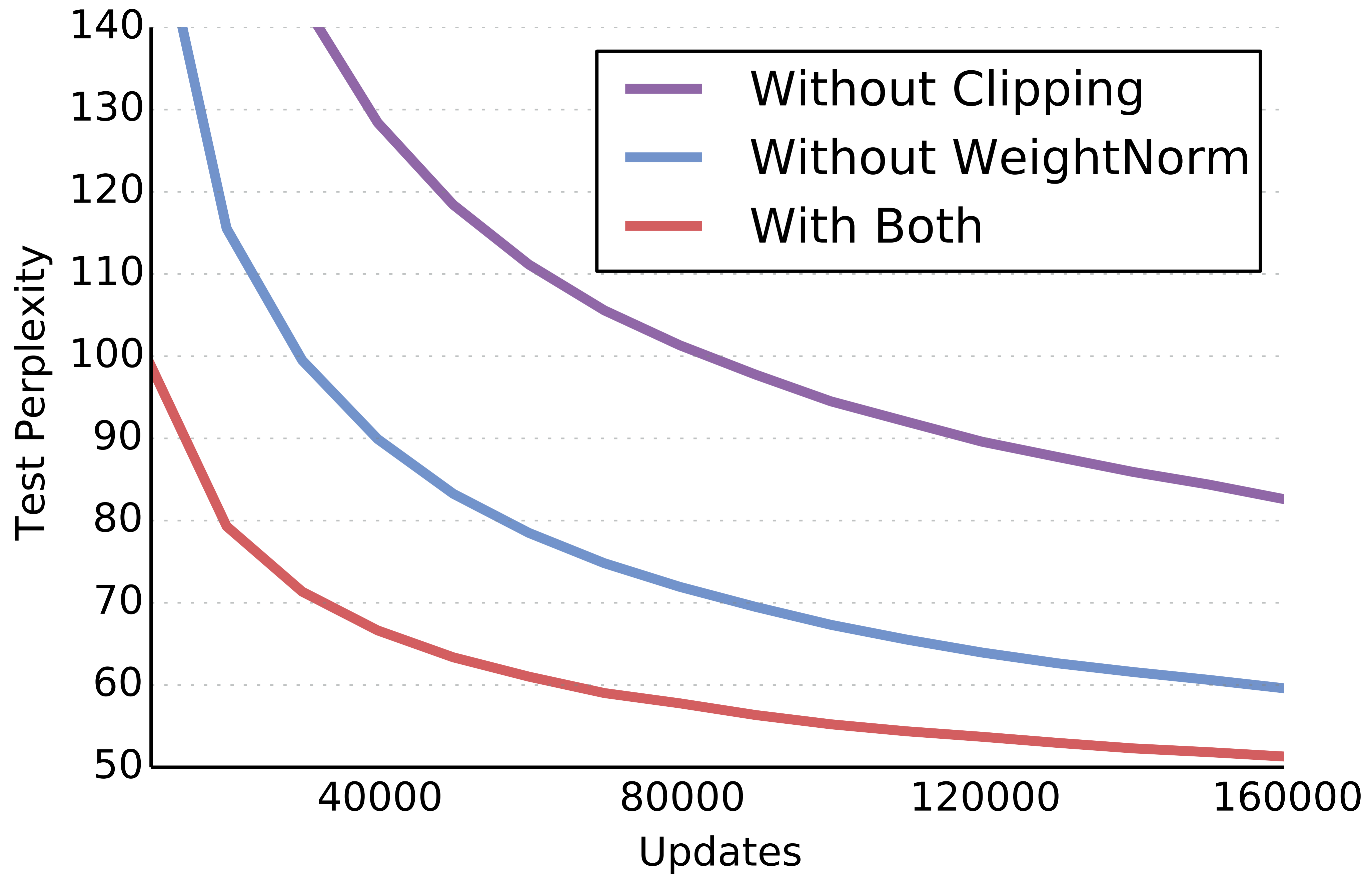
WikiText-103



Google Billion Words

- Competitive performance can be achieved with context of less than 40 tokens.

Training algorithm



- Clipping and weight normalization speed up convergence by allowing large learning rates without divergence

Summary

- Fully convolutional model of language that is competitive with LSTMs.
- Demonstrated impact of gating mechanisms for this task.
- Shown faster response times with this approach.

Convolutional Sequence to Sequence Learning

Convolutional Sequence to Sequence Learning

- non-RNN models can outperform very well-engineered RNNs on large translation benchmarks
- Multi-hop attention
- Approach with very fast inference speed: >9x faster than RNN

Code and pre-trained models available!
Lua/Torch: https://github.com/facebookresearch/fairseq
PyTorch: https://github.com/facebookresearch/fairseq-py

Previous work

- ByteNet (Kalchbrenner et al. 2016)
  Characters, dilated convolutions, no attention

- Quasi-RNNs (Bradburry et al., 2016)
  Recurrent pooling of CNN outputs, but still an RNN

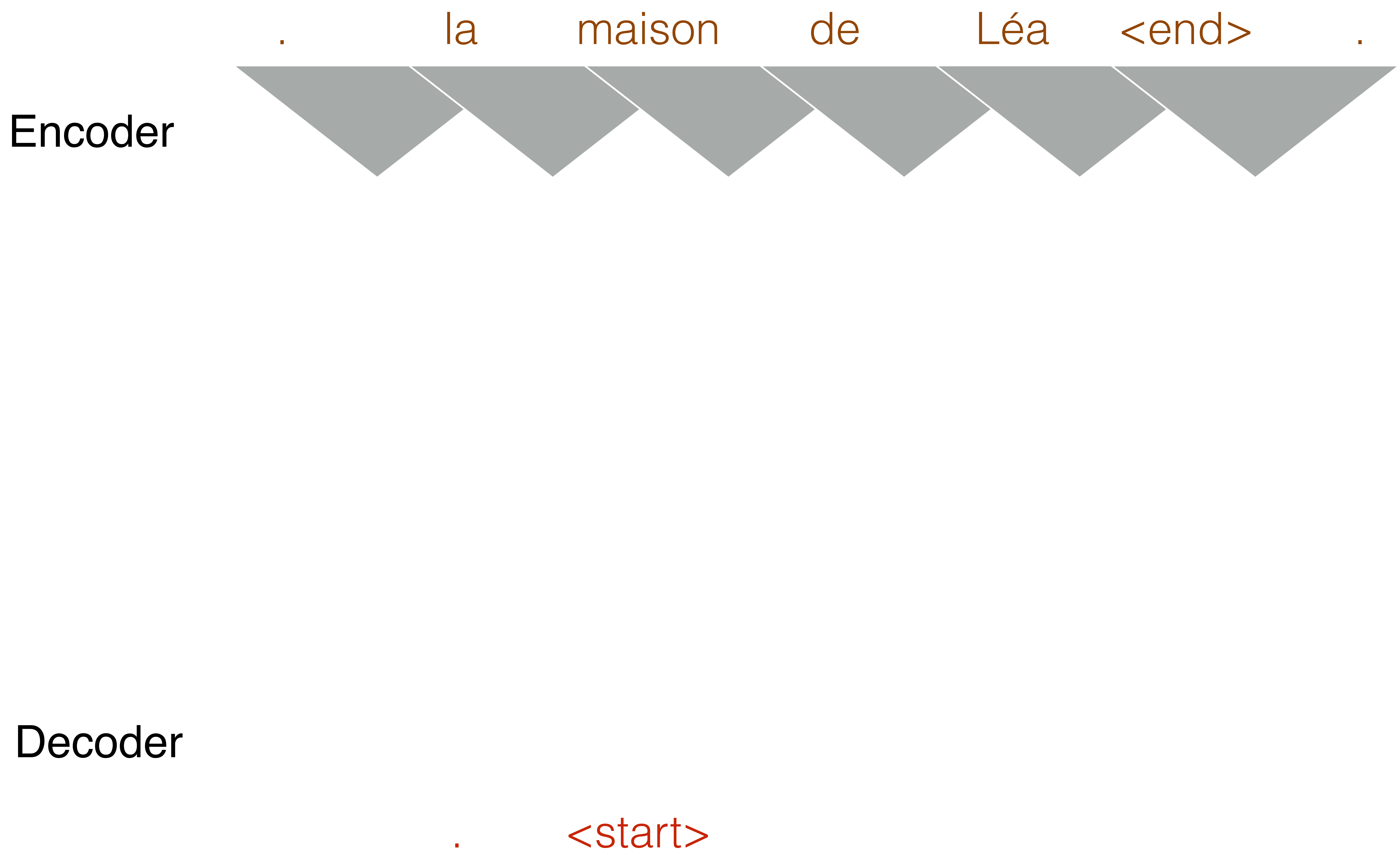
- Convolutional encoders (Gehring et al., 2016)
  CNN encoder, LSTM decoder

. la maison de Léa <end> .

. la maison de Léa <end> .

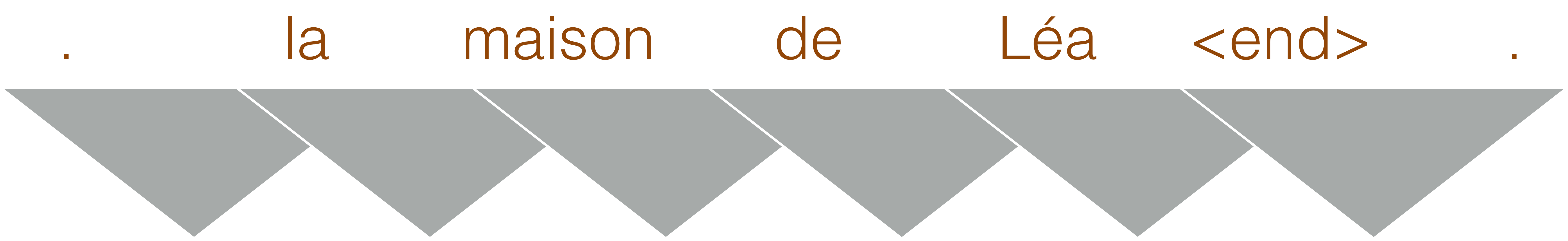
Encoder


. la maison de Léa <end> .
Encoder



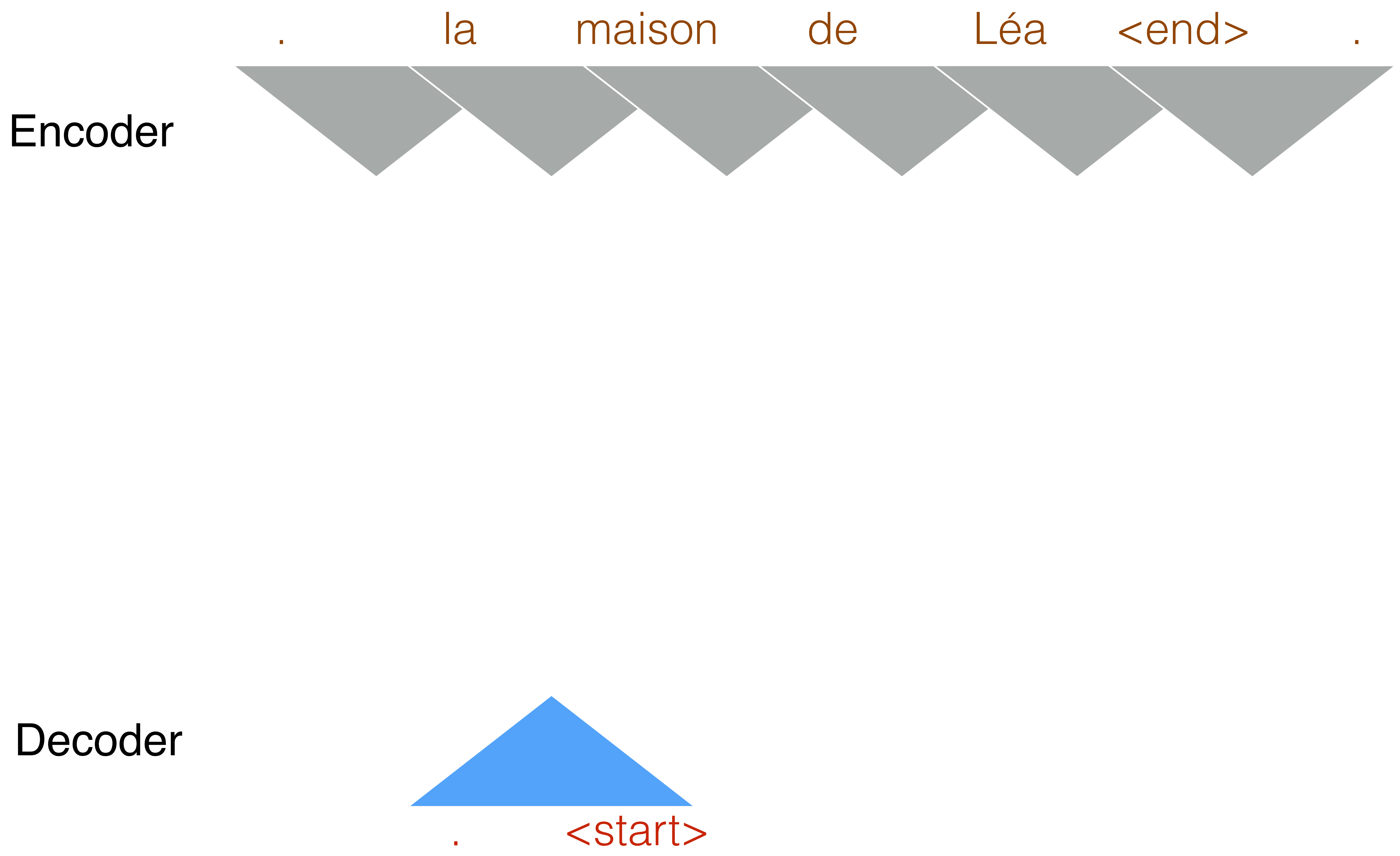
.
la
maison
de
Léa
<end>
.
Encoder
Decoder
.
<start>

Encoder

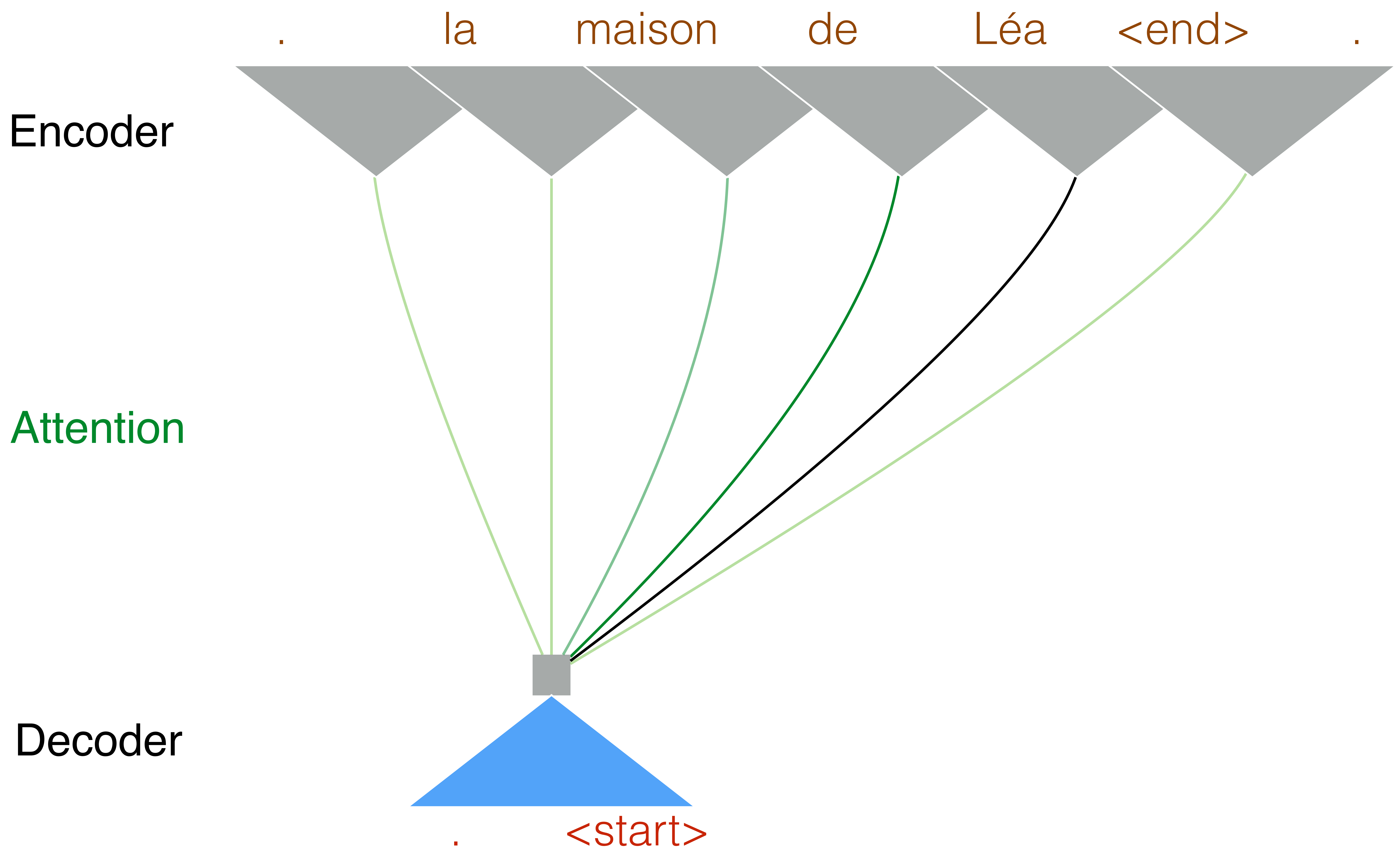


Decoder

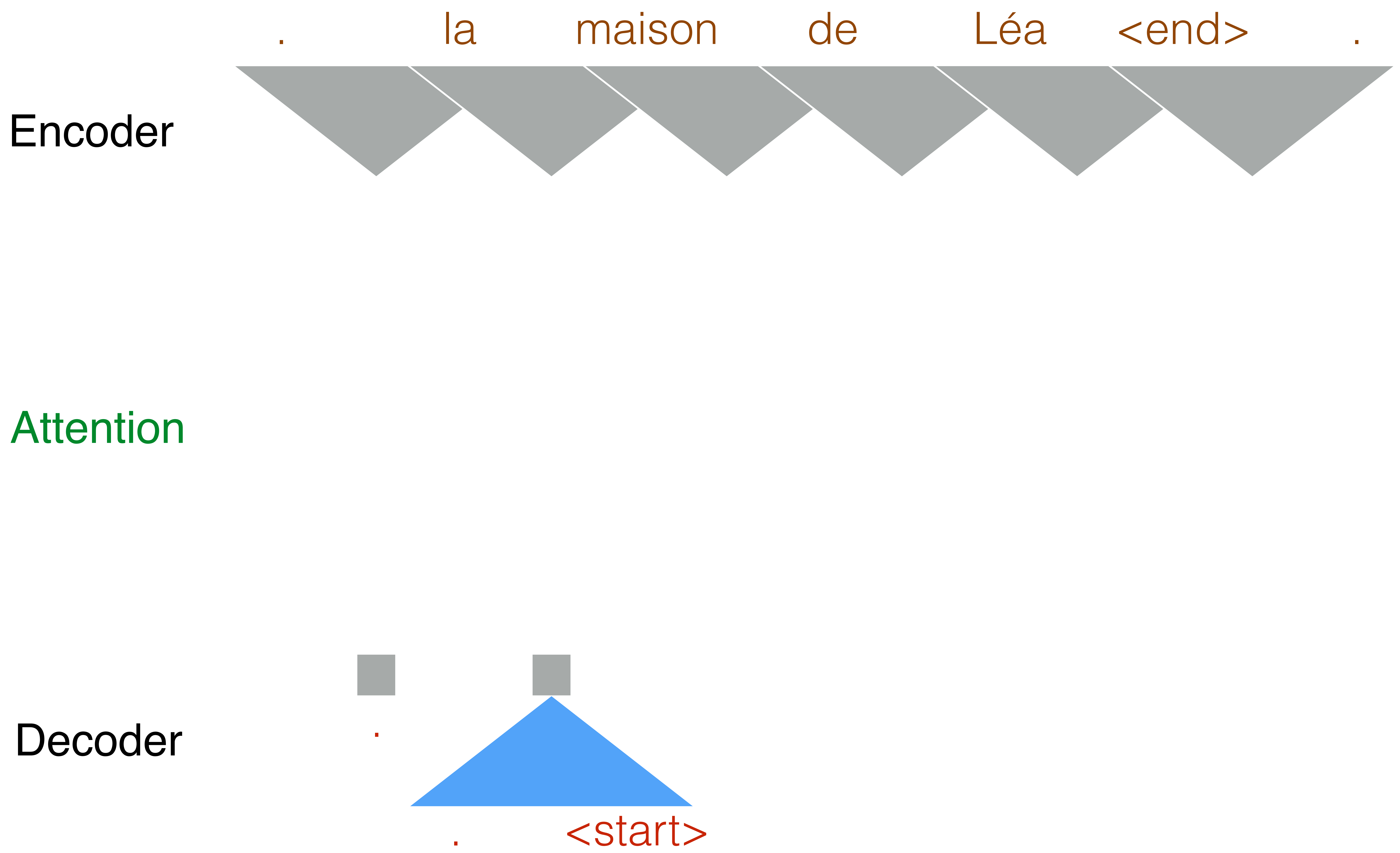
. <start>



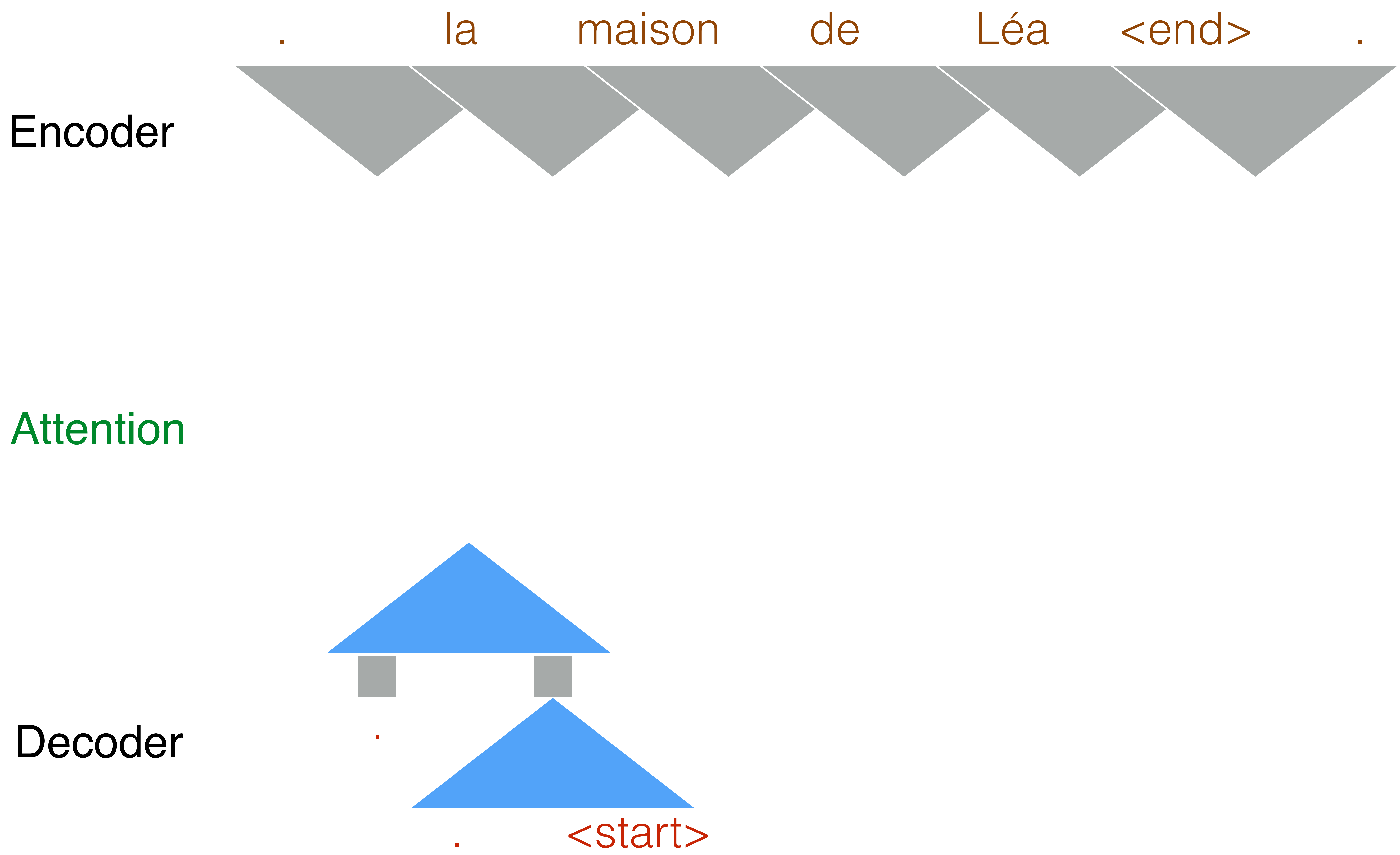
. la maison de Léa <end> .
Encoder
Decoder
. <start>

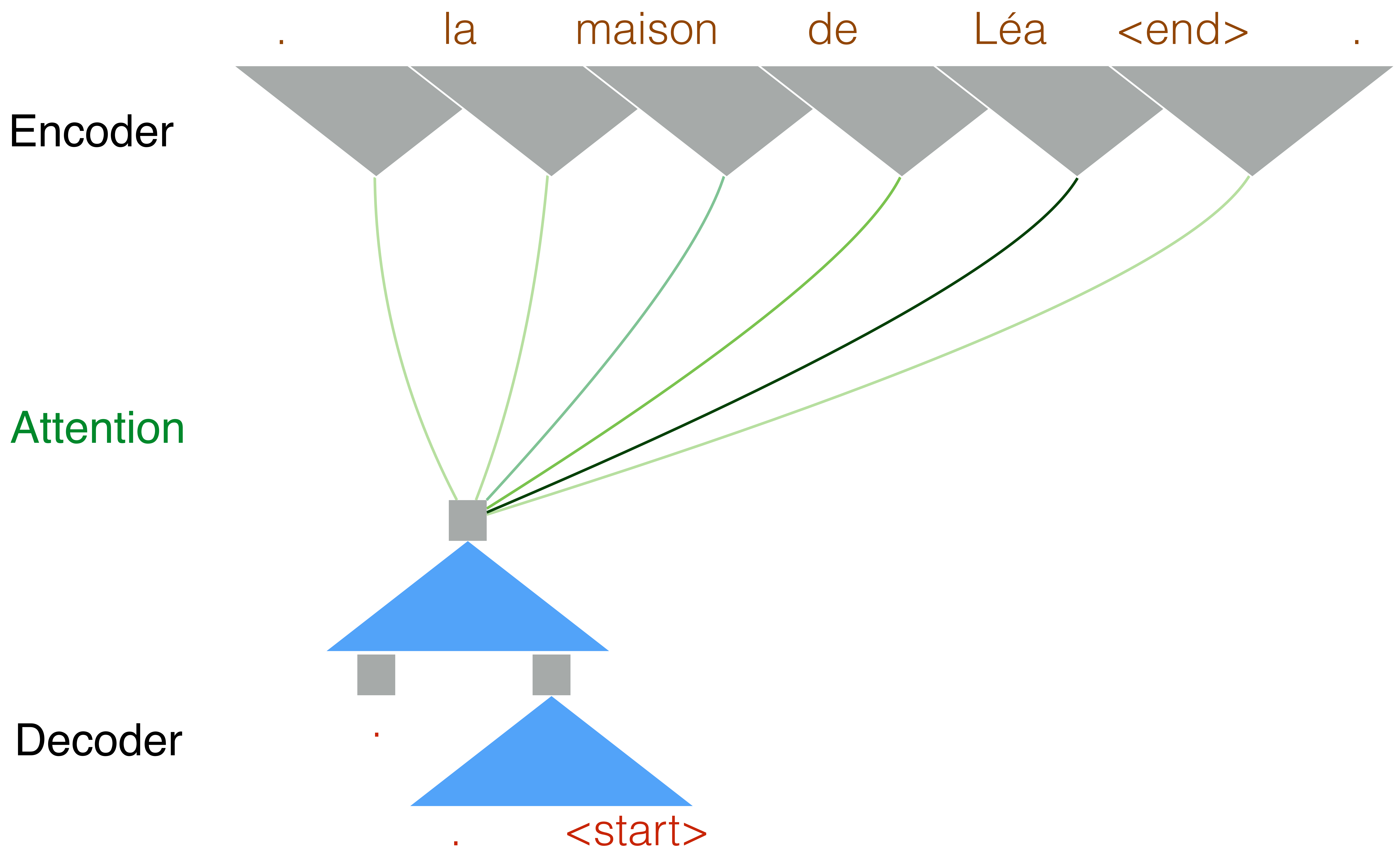


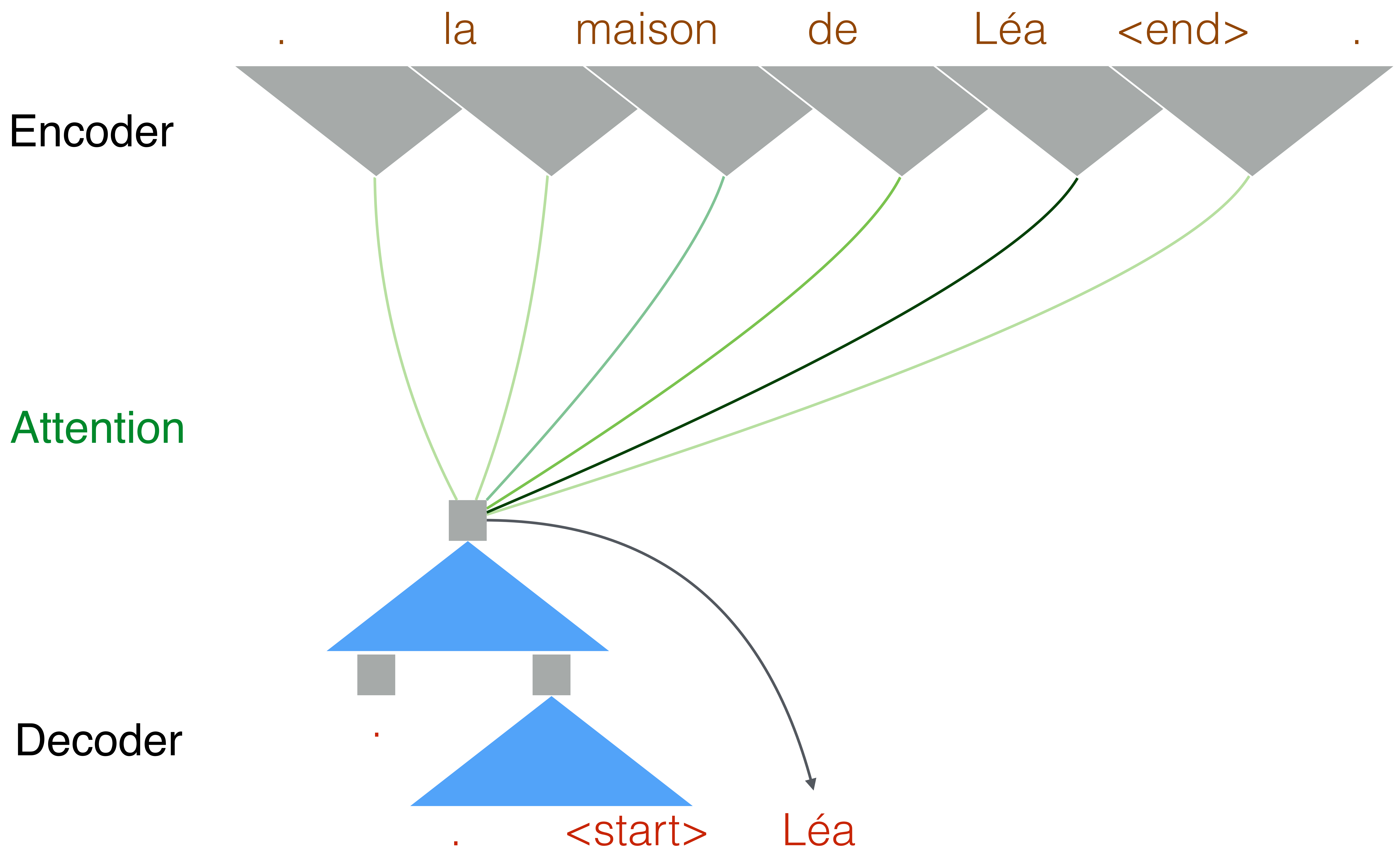
.
la
maison
de
Léa
<end>
.
Encoder
Attention
Decoder
.
<start>

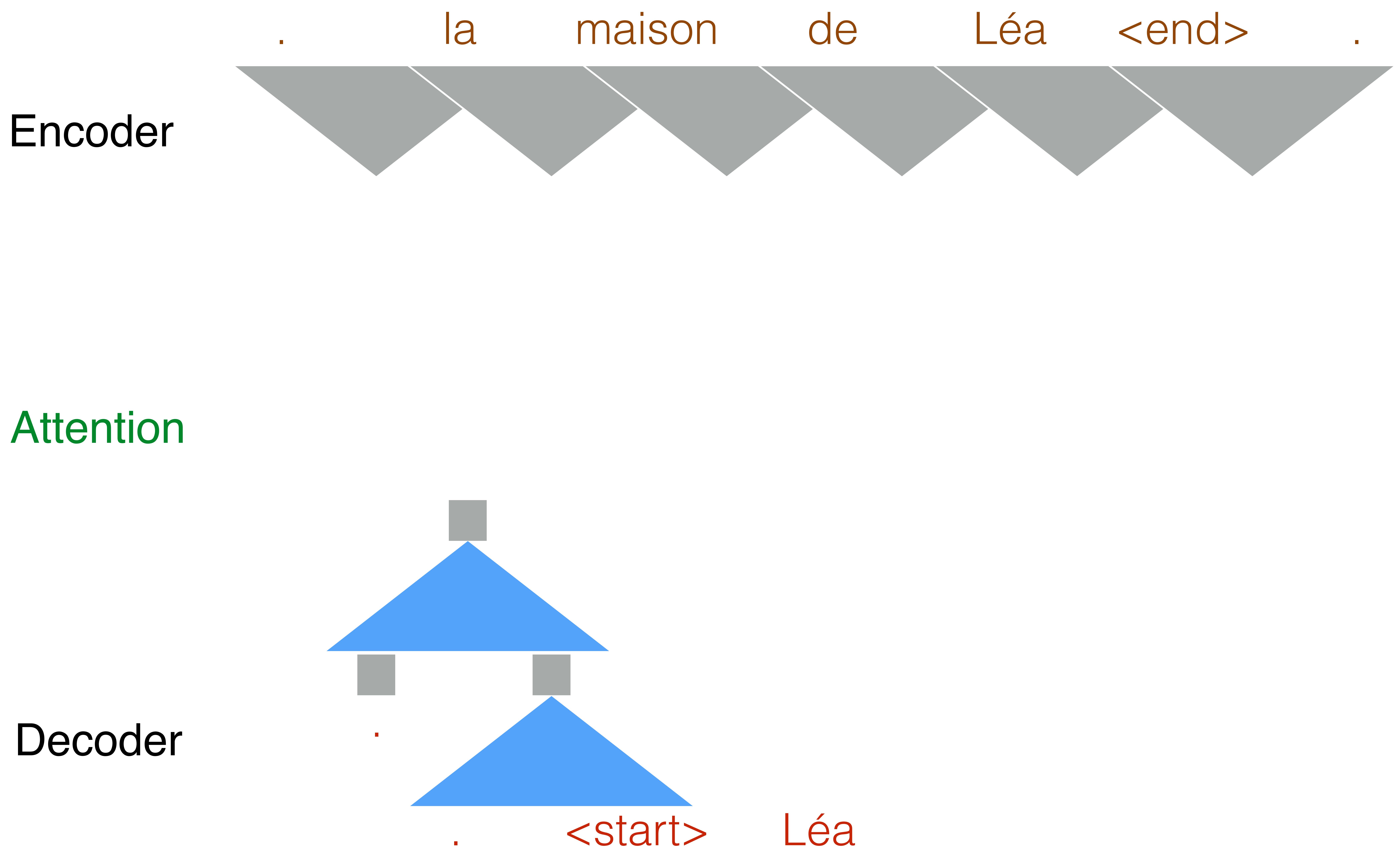


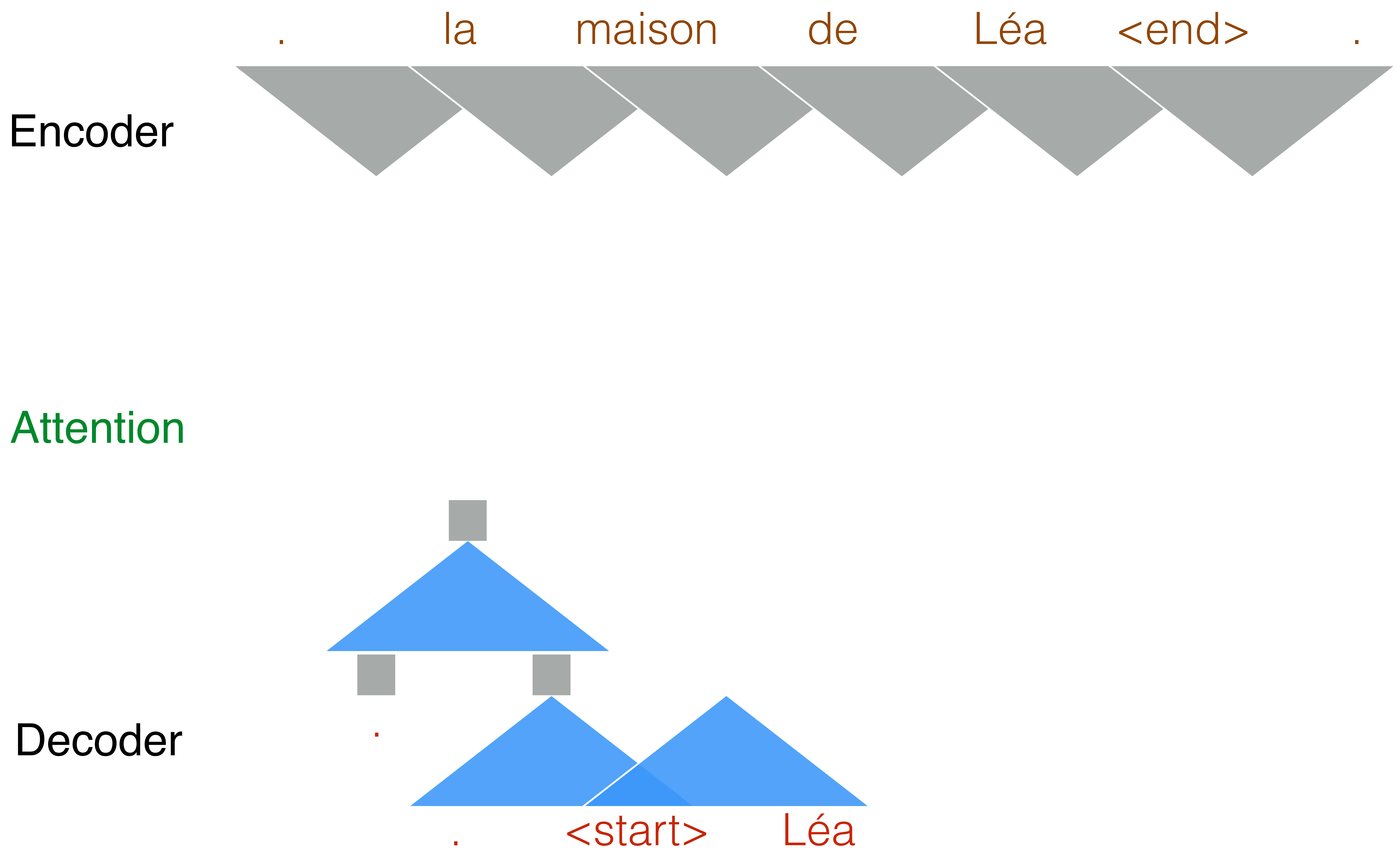
.
la
maison
de
Léa
<end>
.
Encoder
Attention
Decoder
.
.
<start>


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

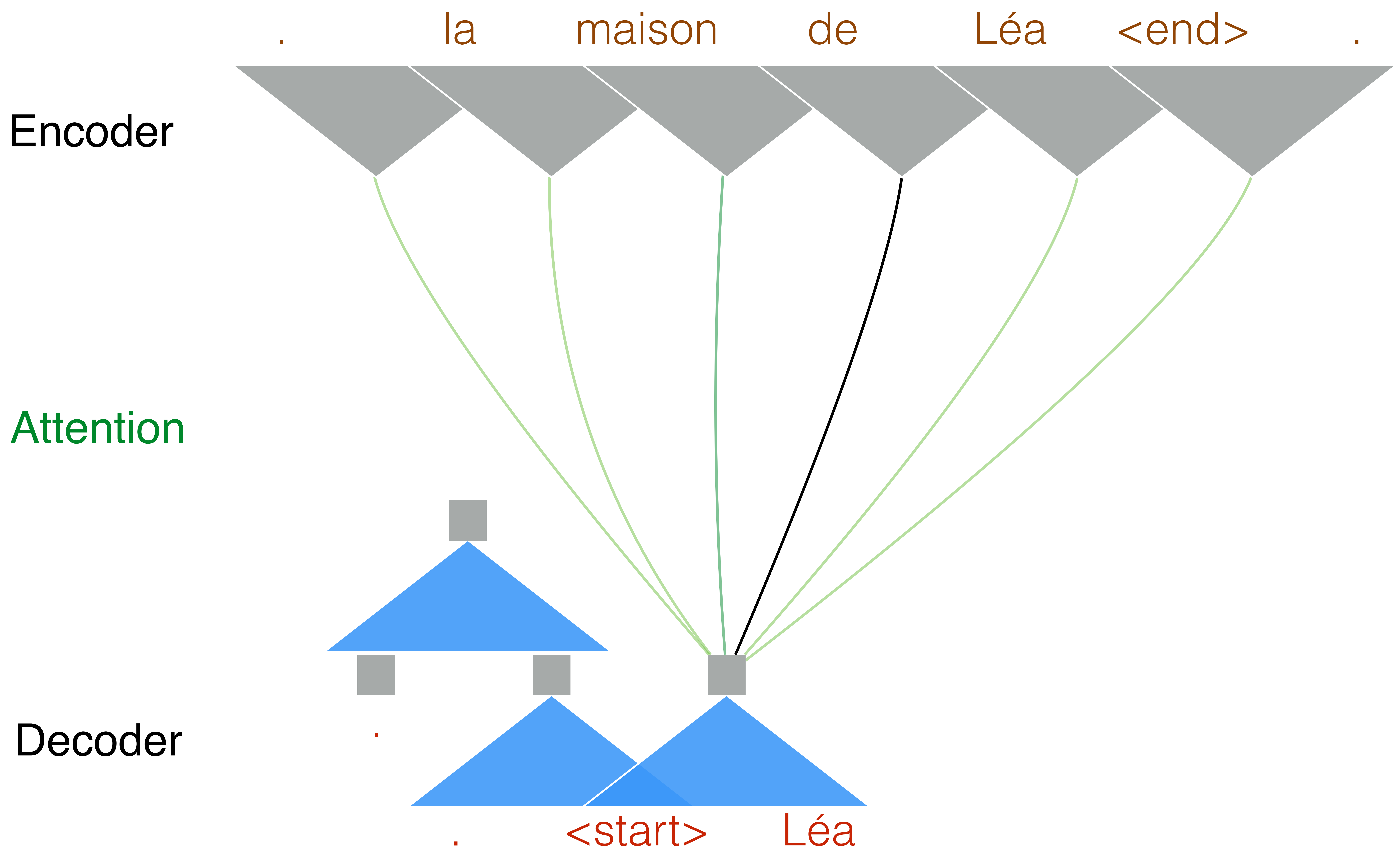

. la maison de Léa <end> .
Encoder
Attention
Decoder
.
. <start>


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

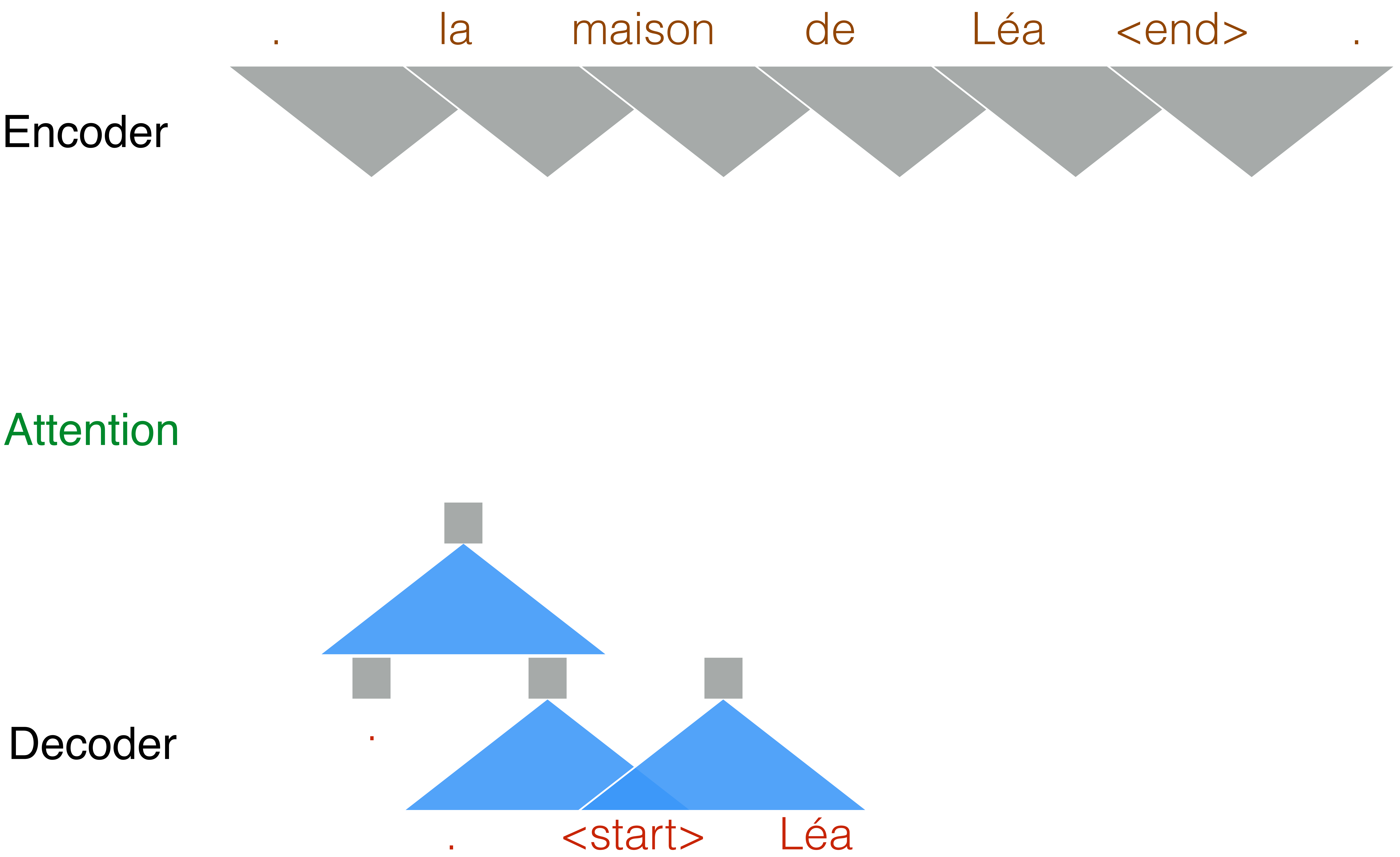

. la maison de Léa <end> .
Encoder
Attention
.
Decoder
. <start> Léa

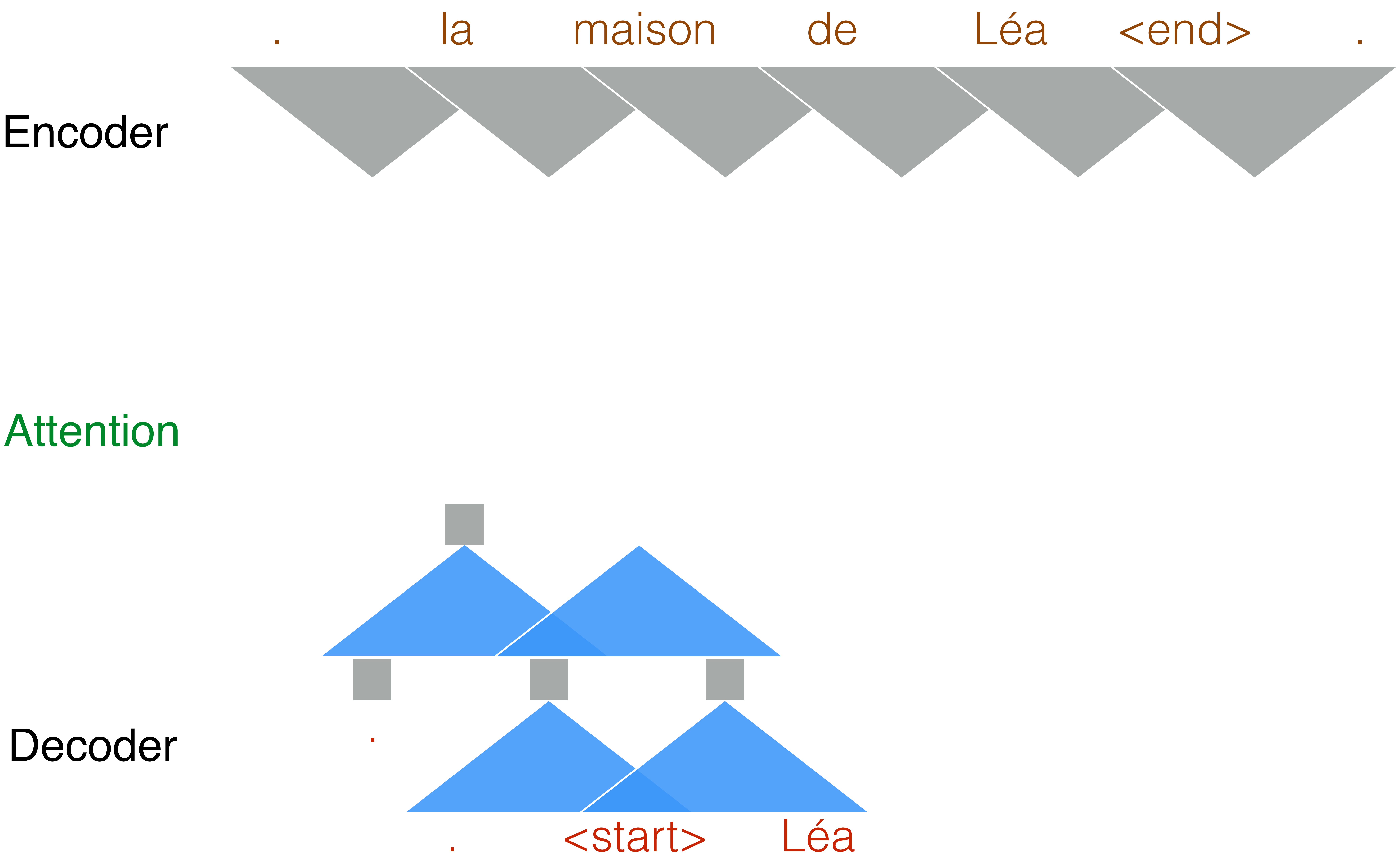


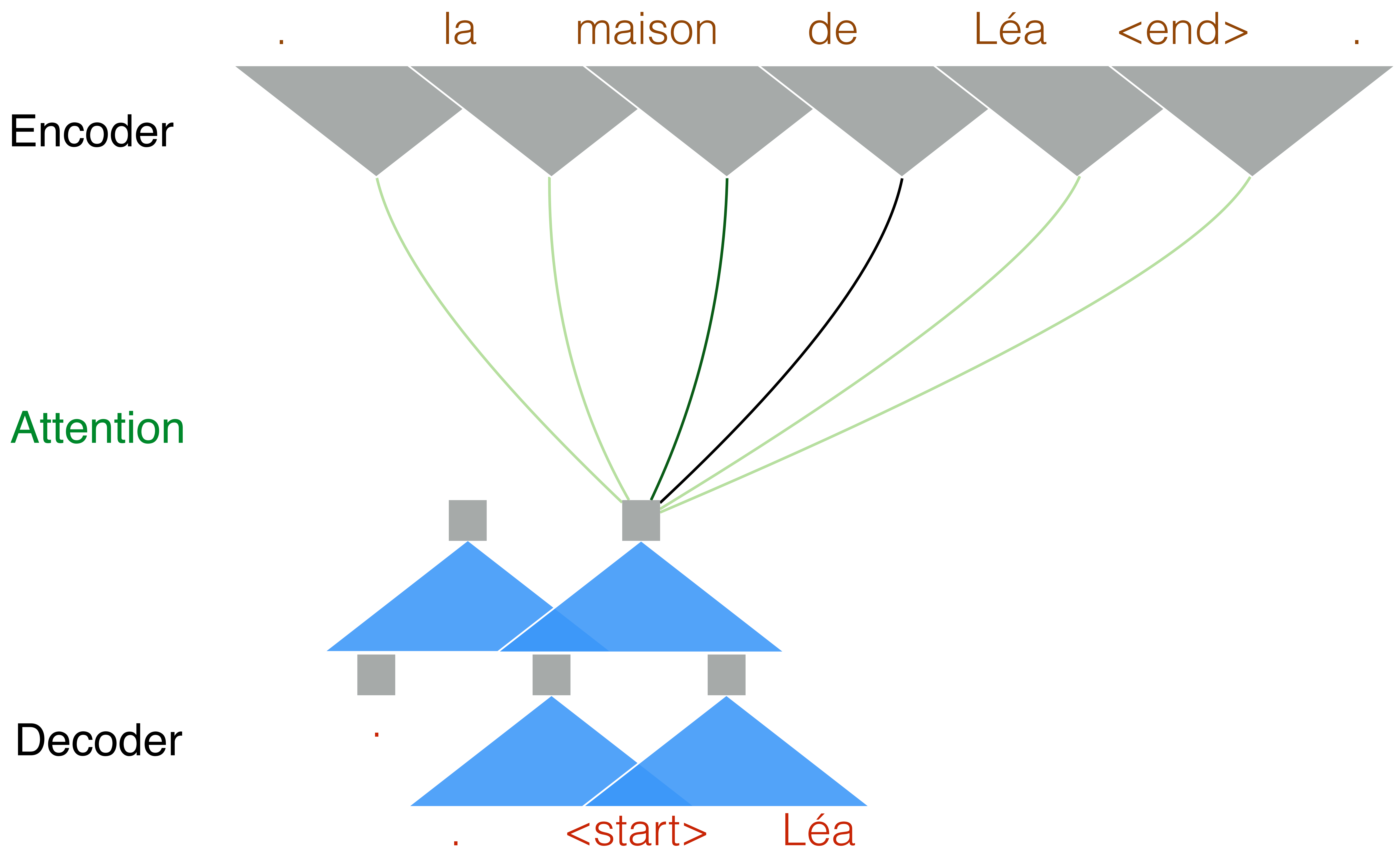
. la maison de Léa <end> .
Encoder
Attention
Decoder
.
. <start> Léa



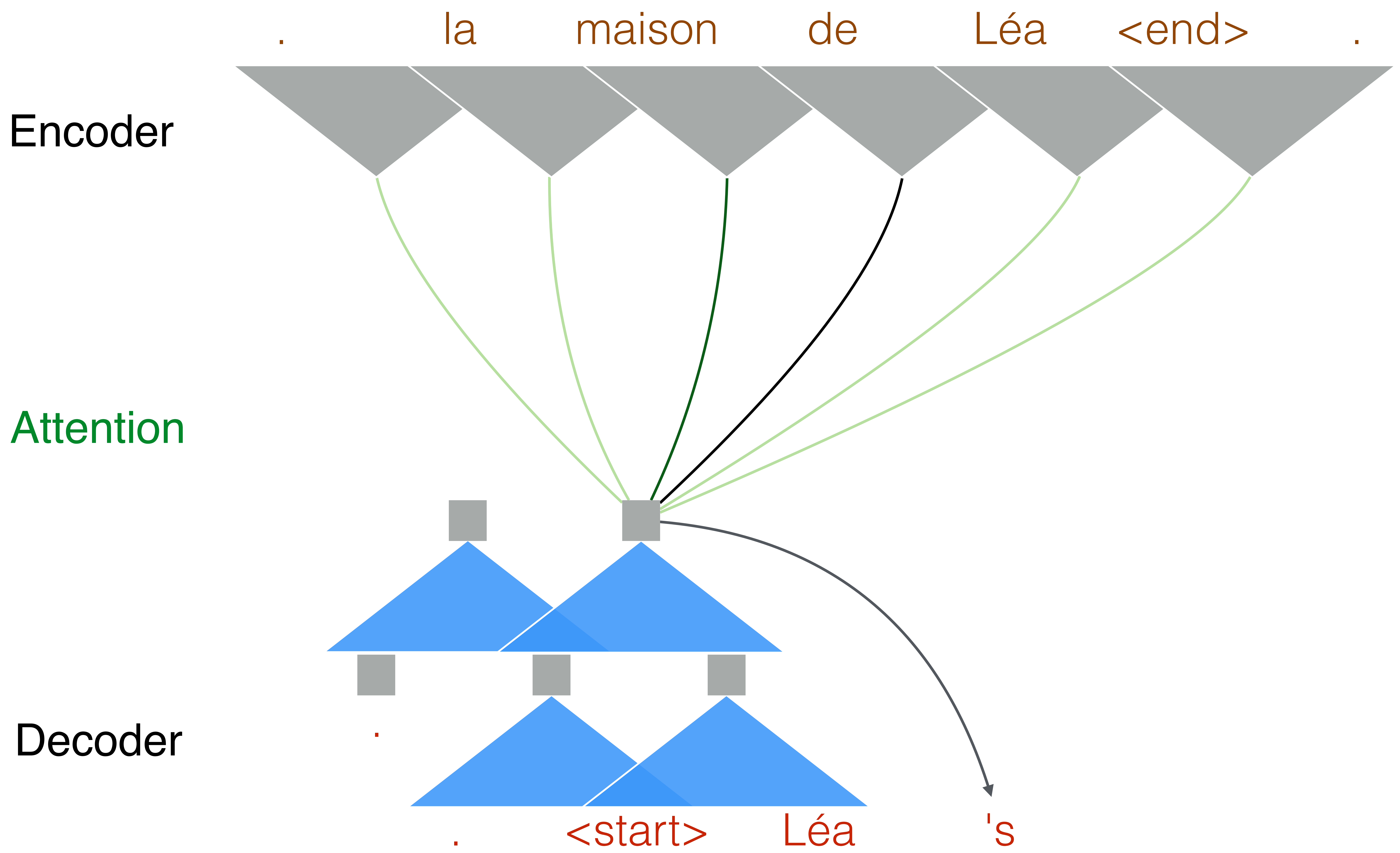
. la maison de Léa <end> .
Encoder
Attention
Decoder
.
. <start> Léa


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

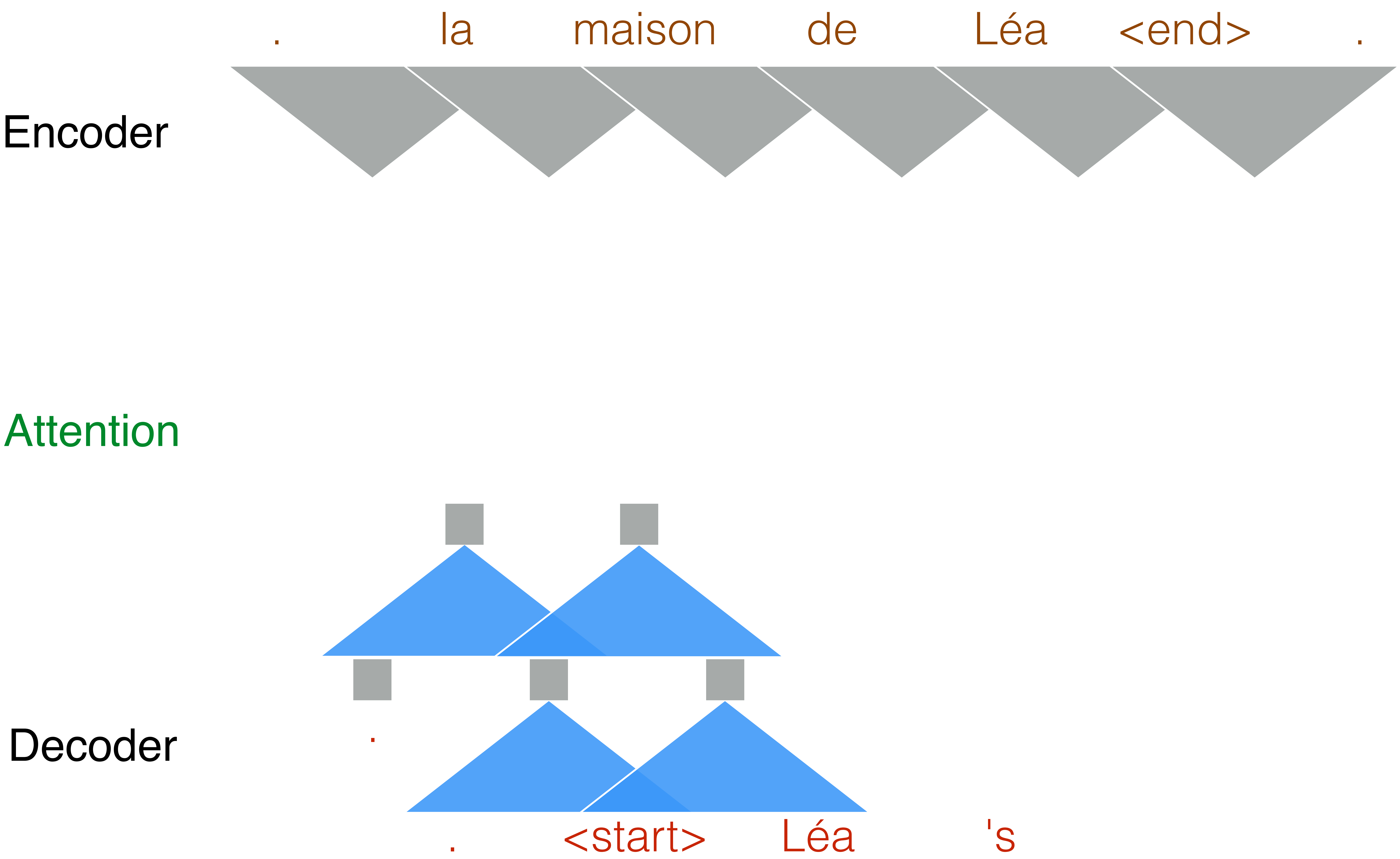

. la maison de Léa <end> .
Encoder
Attention
Decoder
.
. <start> Léa



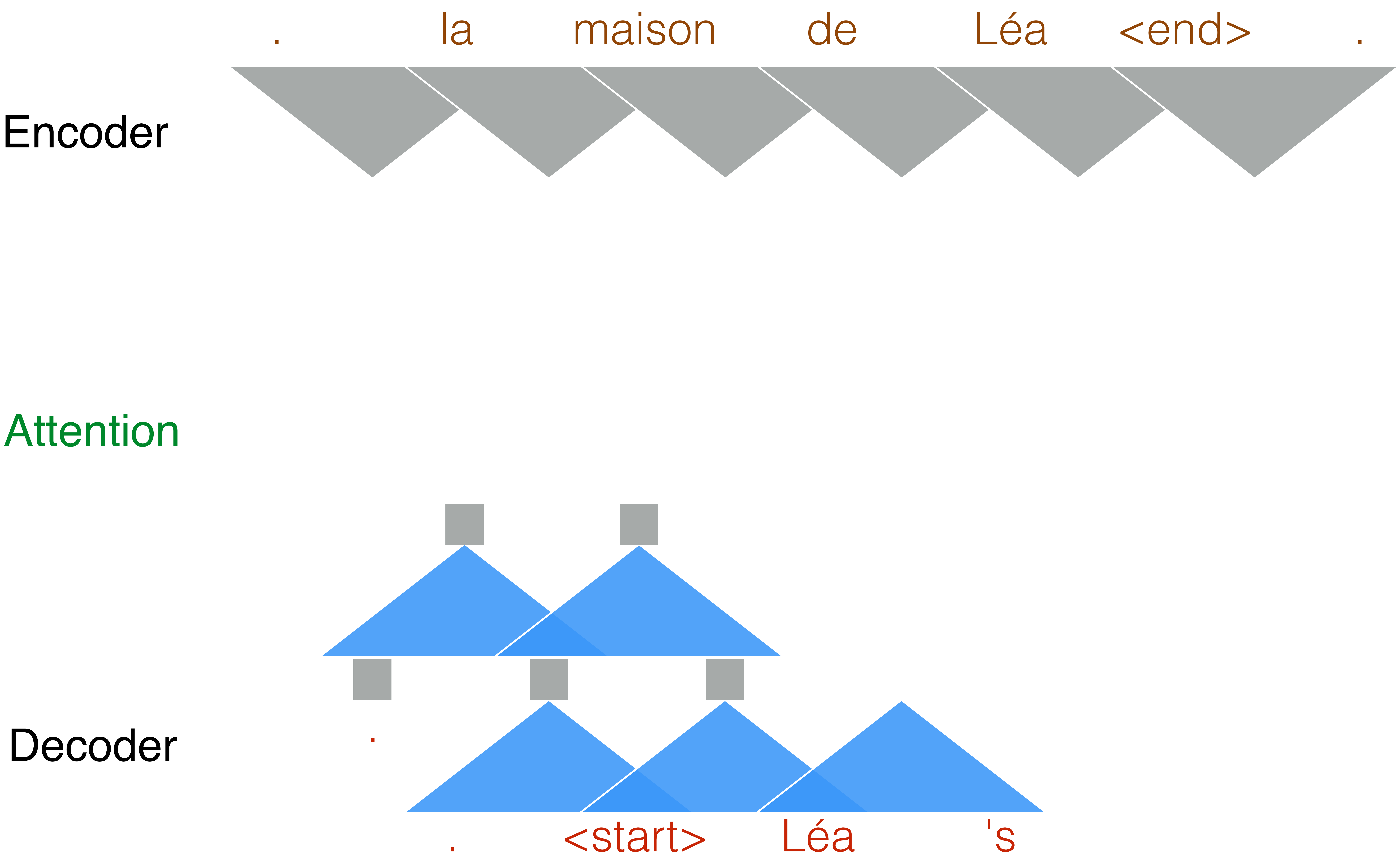
.
la
maison
de
Léa
<end>
.
Encoder
Attention
Decoder
.
.
<start>
Léa

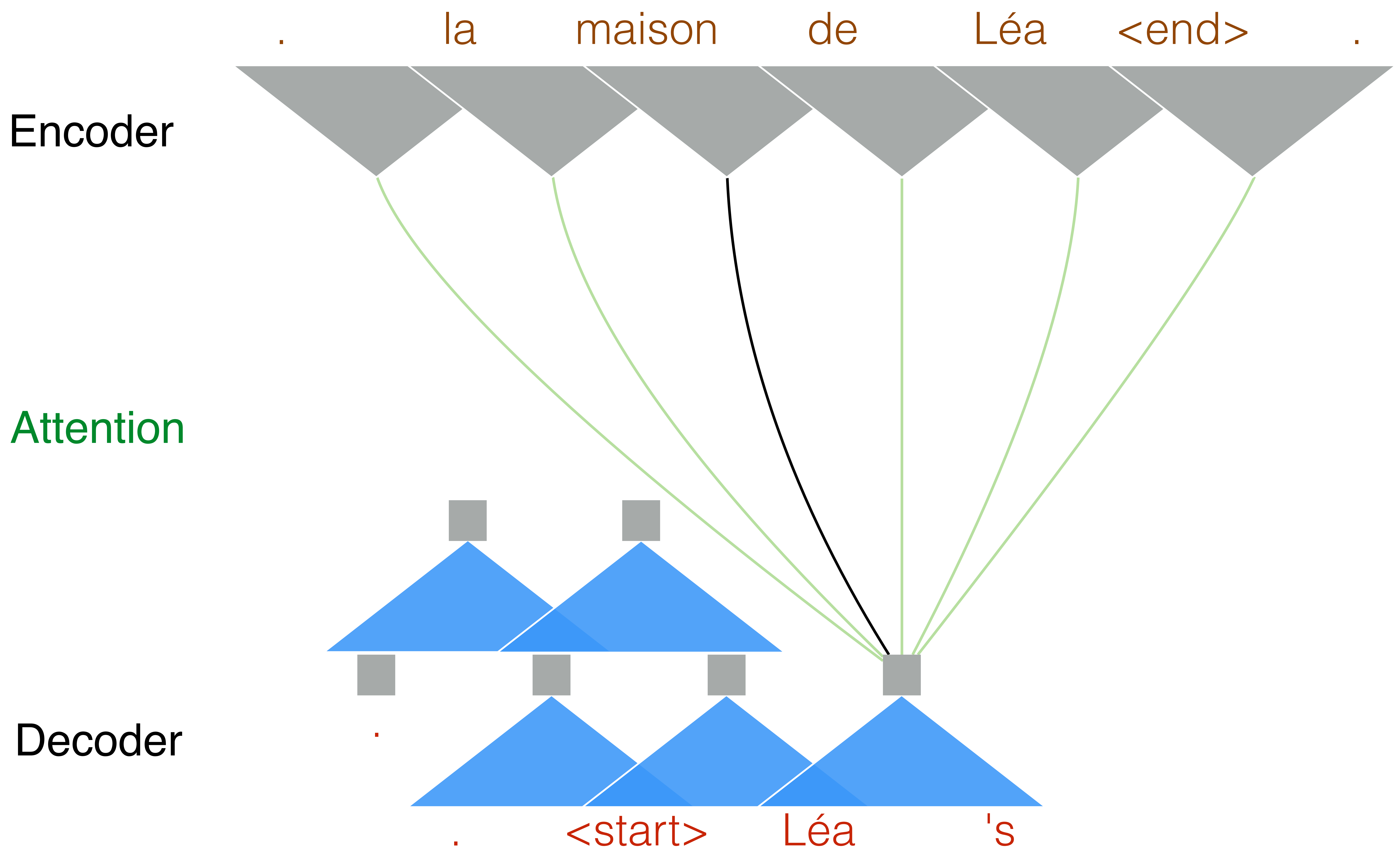


. la maison de Léa <end> .
Encoder
Attention
Decoder
.
. <start> Léa 's

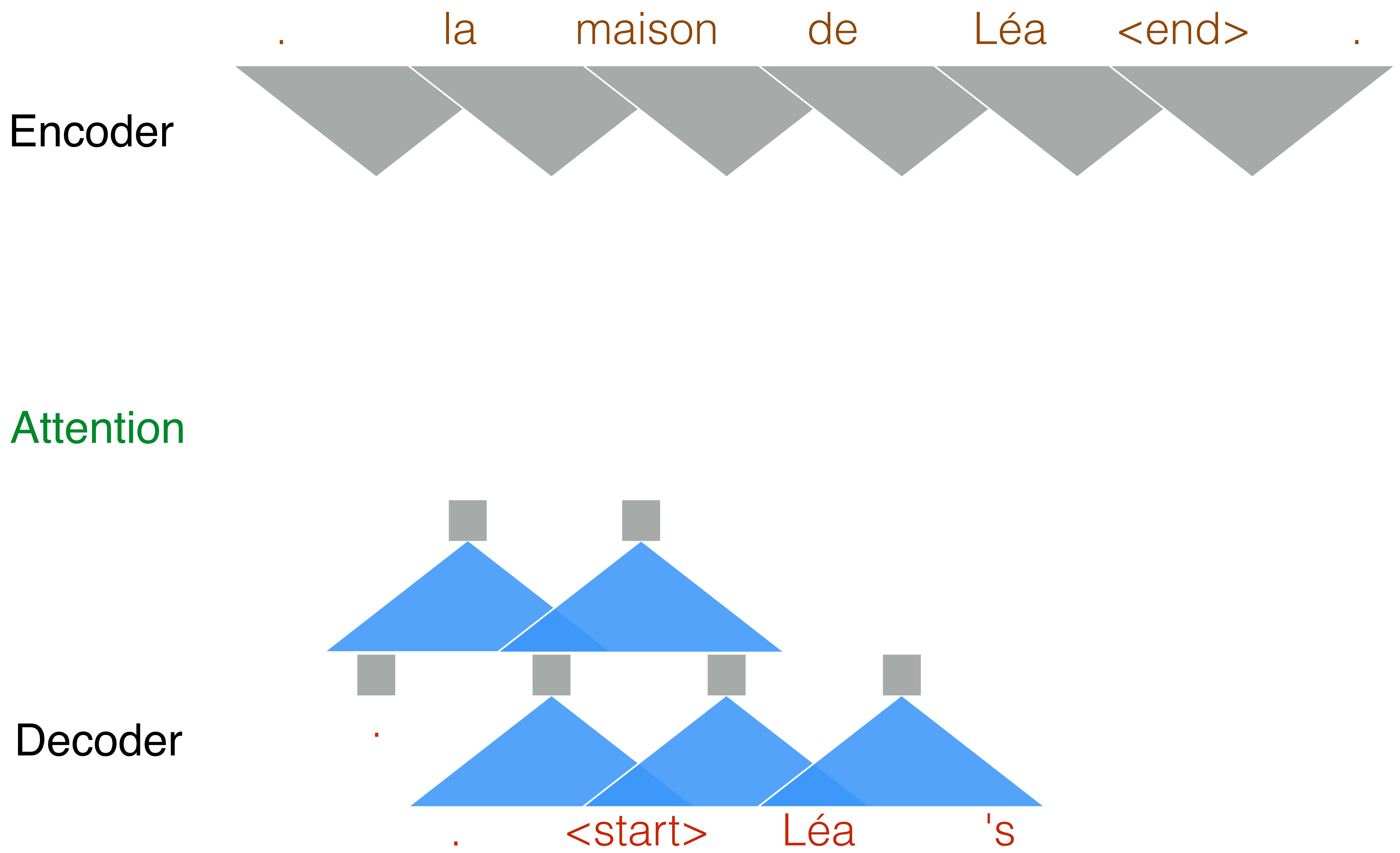


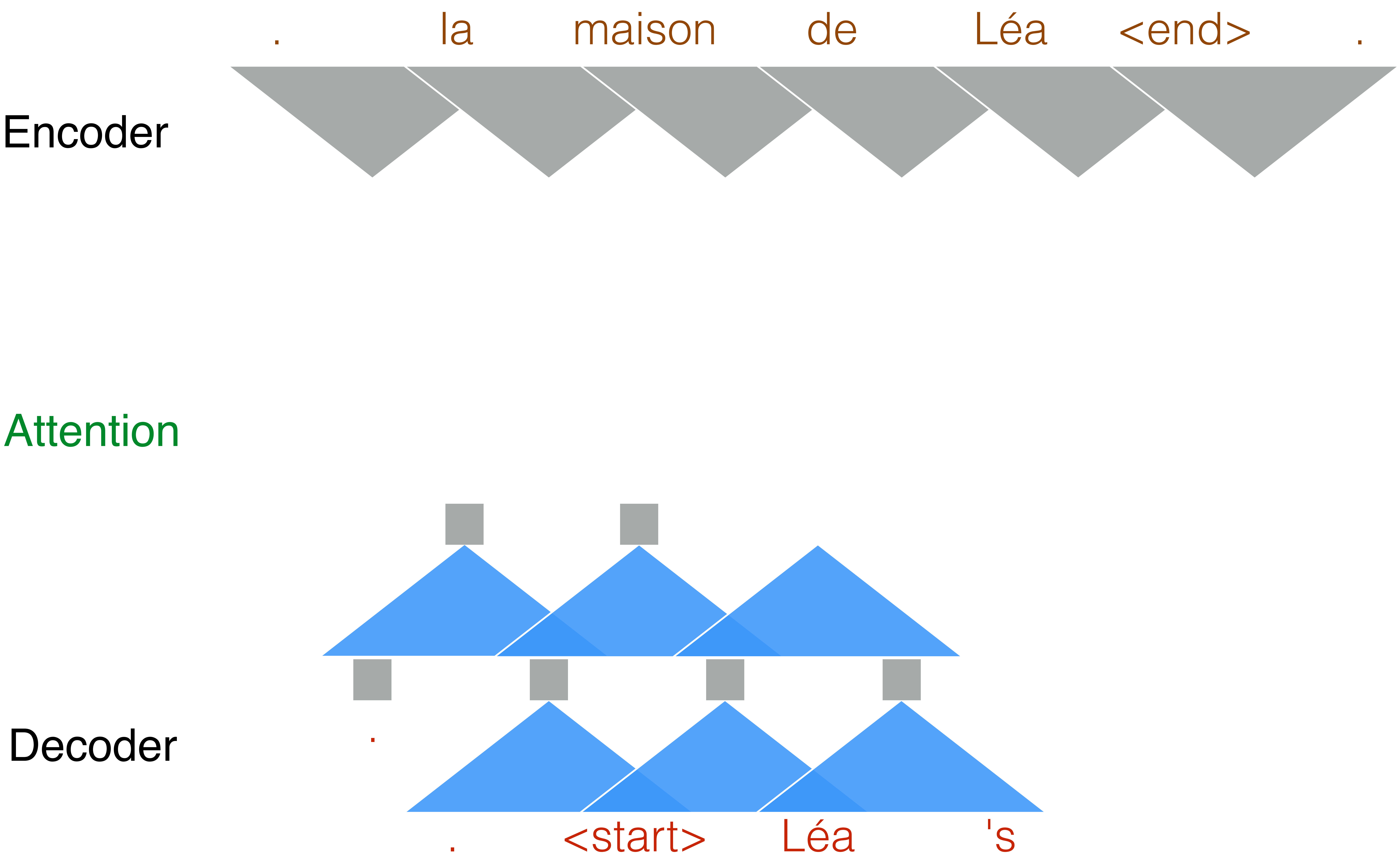
. la maison de Léa <end> .
Encoder
Attention
.
Decoder
. <start> Léa 's

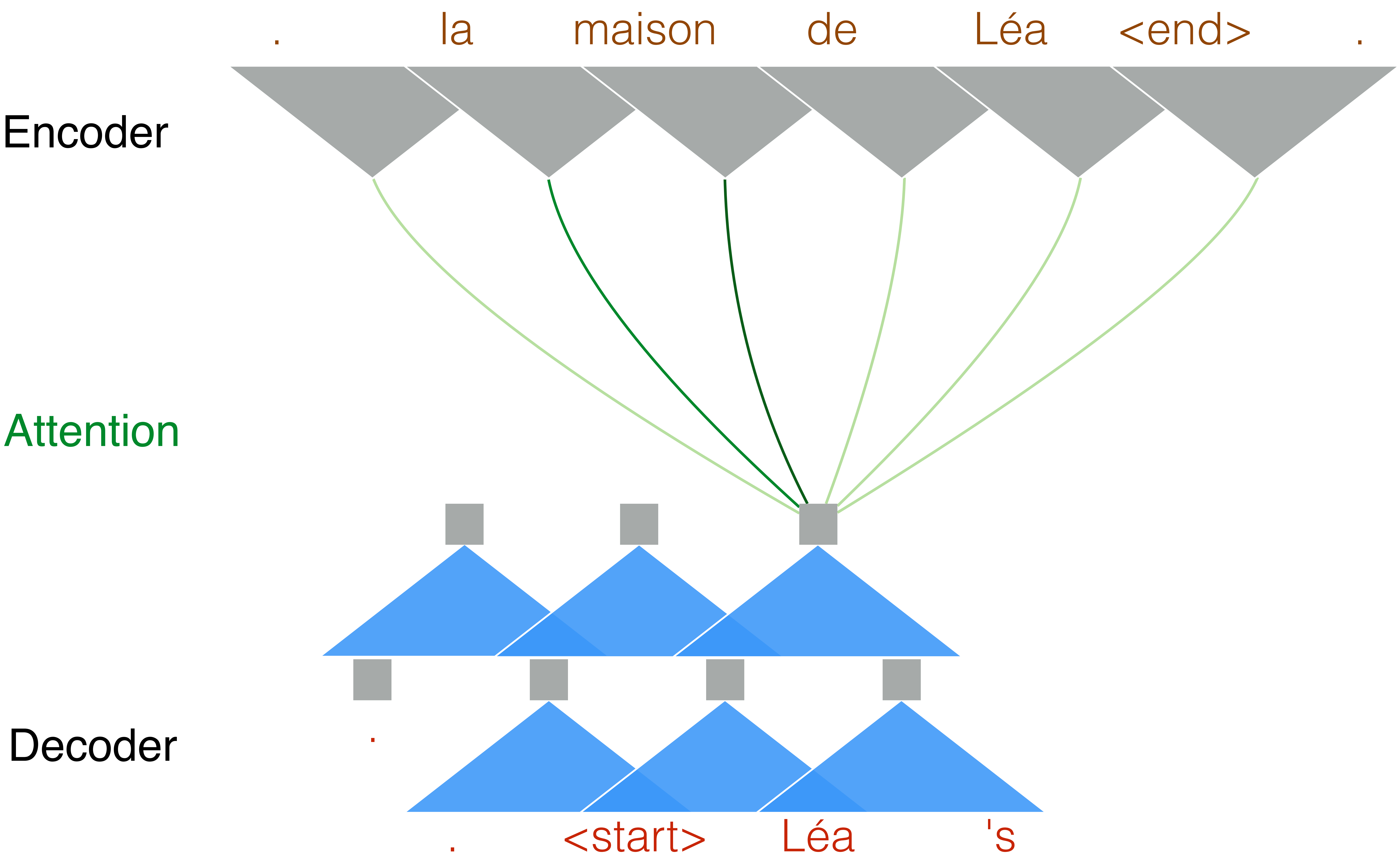

. la maison de Léa <end> .
Encoder
Attention
Decoder
.
. <start> Léa 's

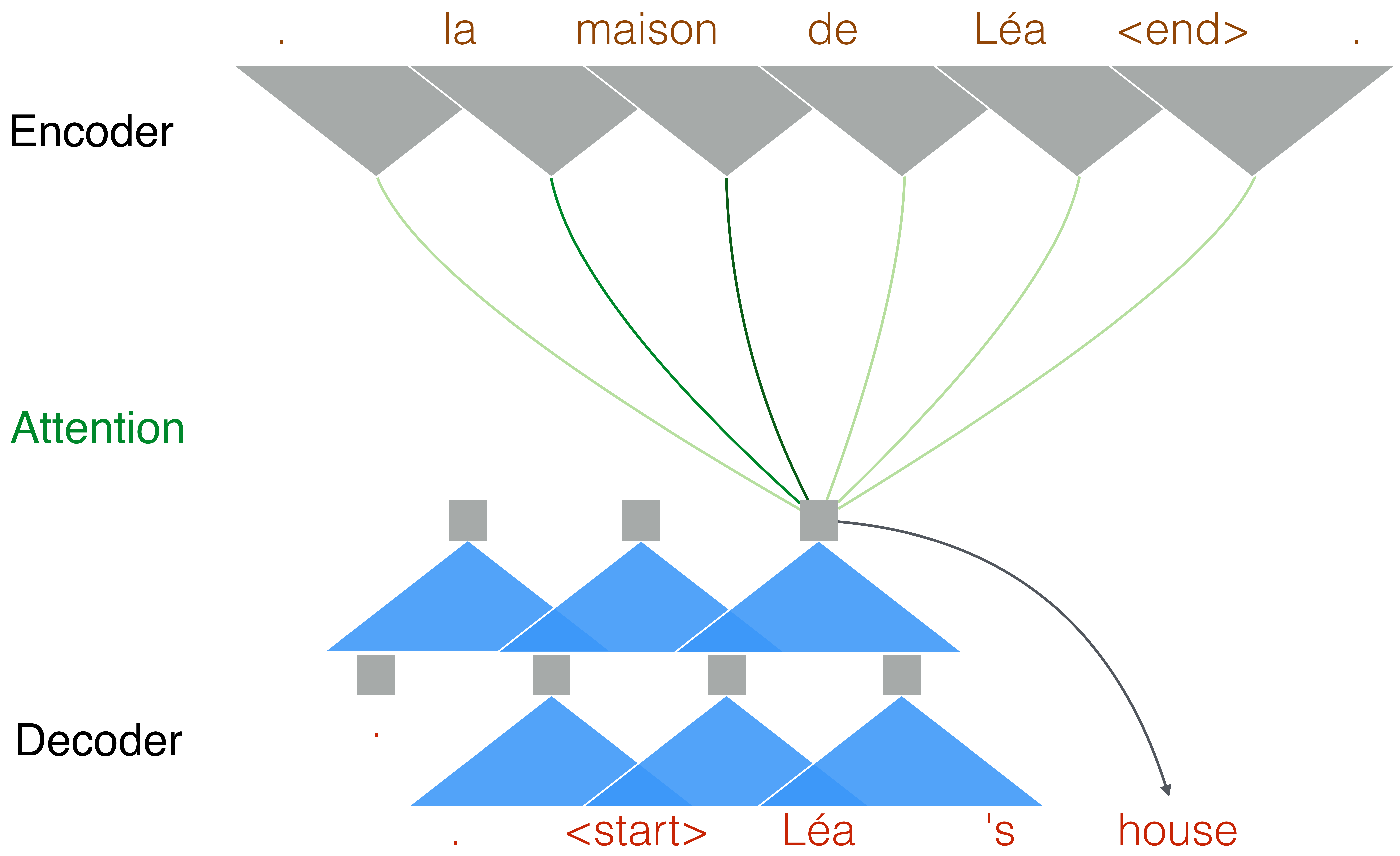


.
la
maison
de
Léa
<end>
.
Encoder
Attention
Decoder
.
.
<start>
Léa
's


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

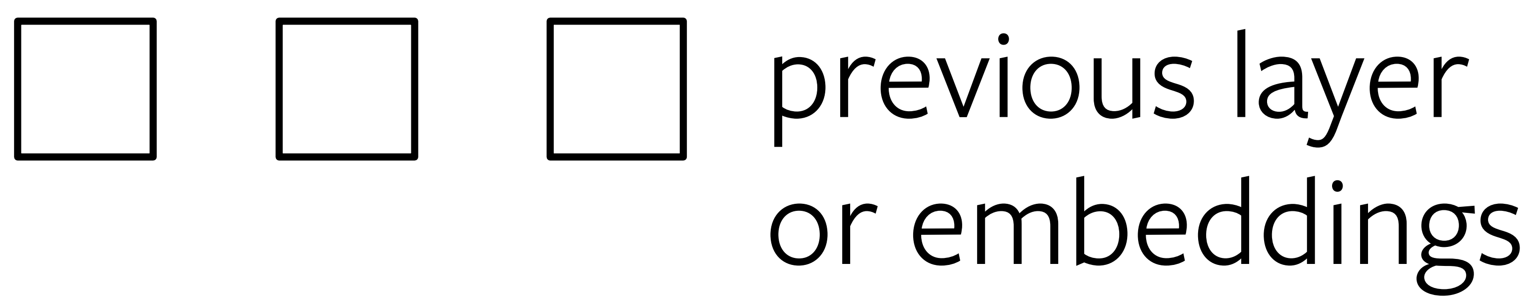

. la maison de Léa <end> .
Encoder
Attention
.
Decoder
. <start> Léa 's


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


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

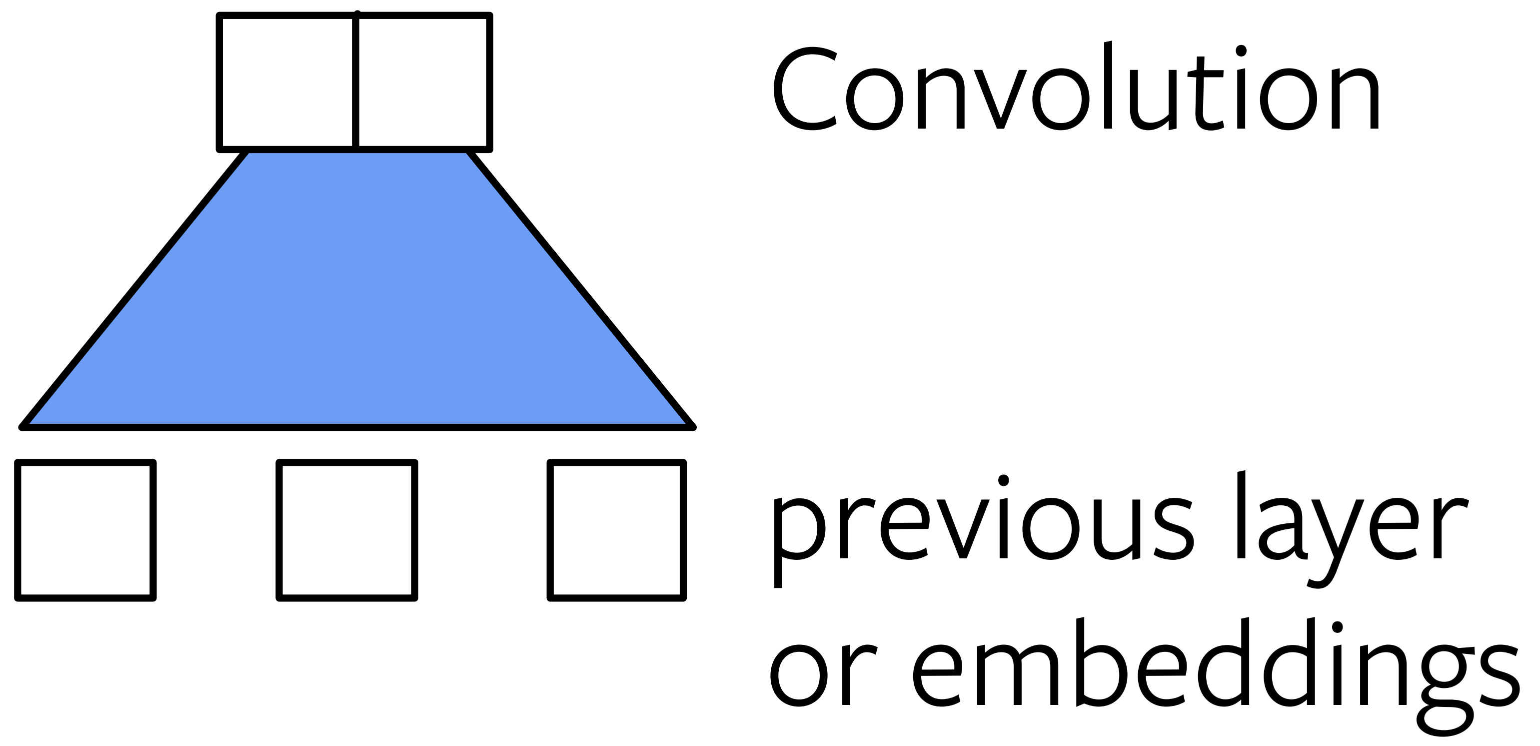
Convolutional S2S: Encoder

- Similar to Dauphin et al. ′17
- 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



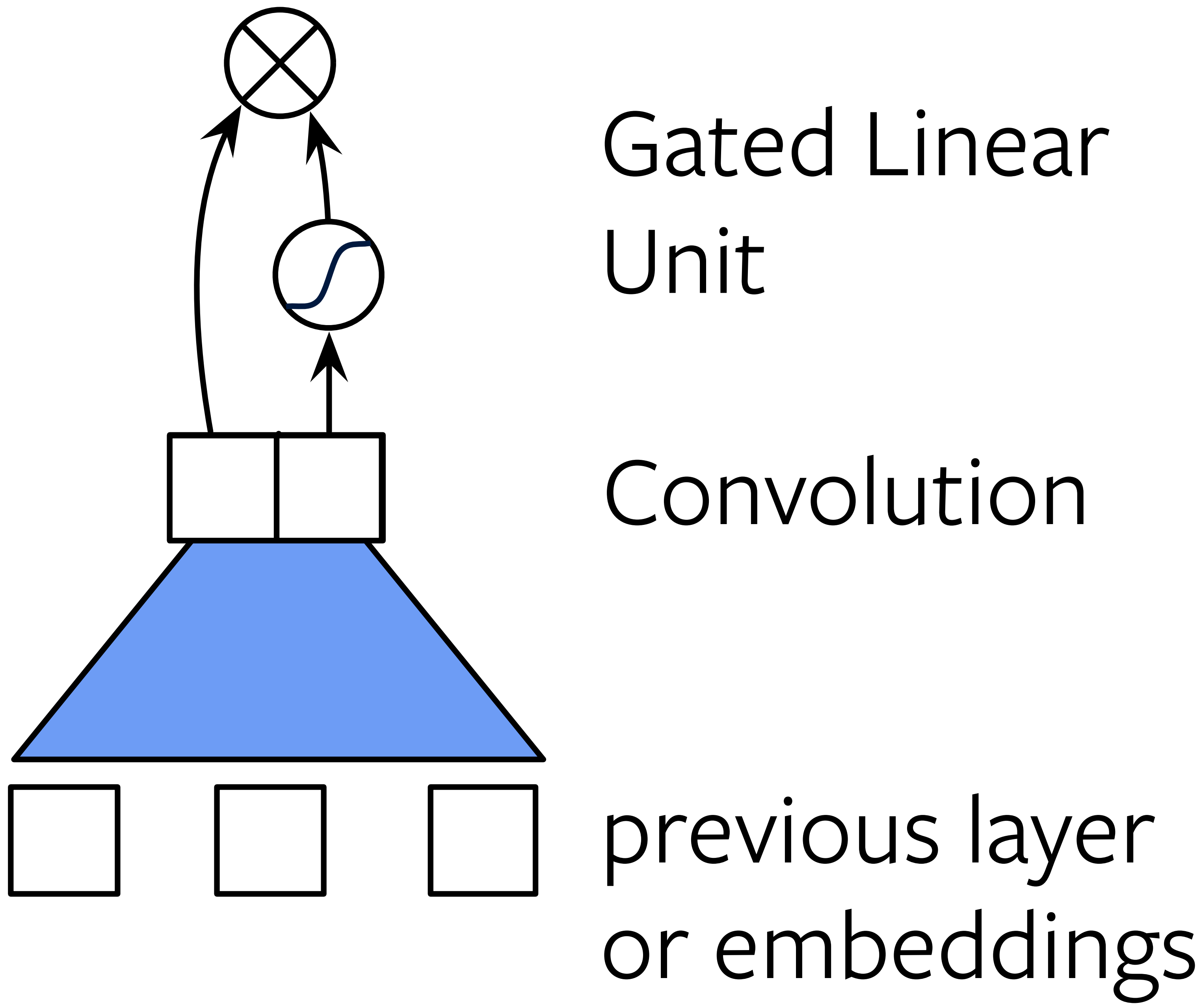
Convolutional S2S: Encoder

- Similar to Dauphin et al. '17
- 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



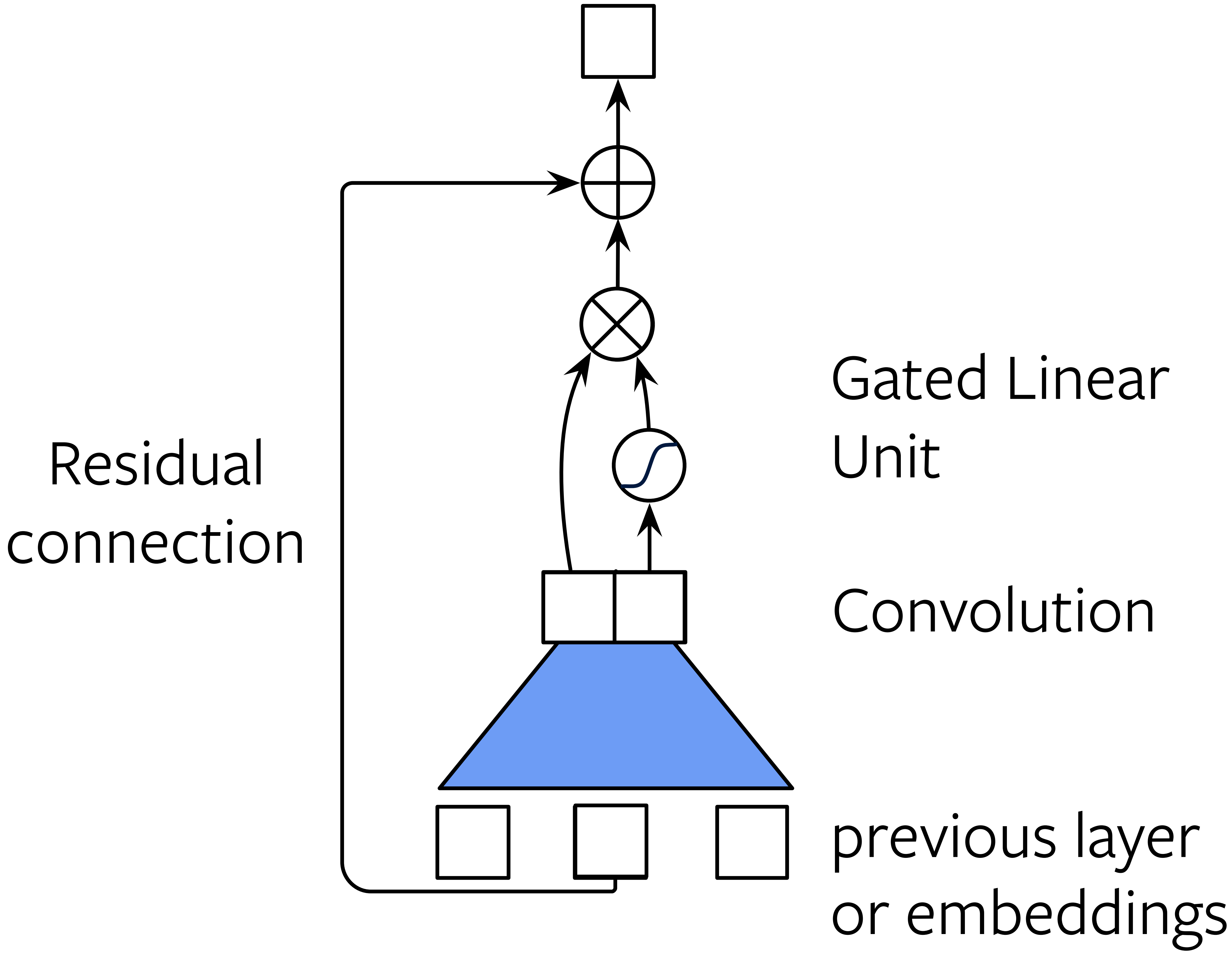
Convolutional S2S: Encoder

- Similar to Dauphin et al. '17
- 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



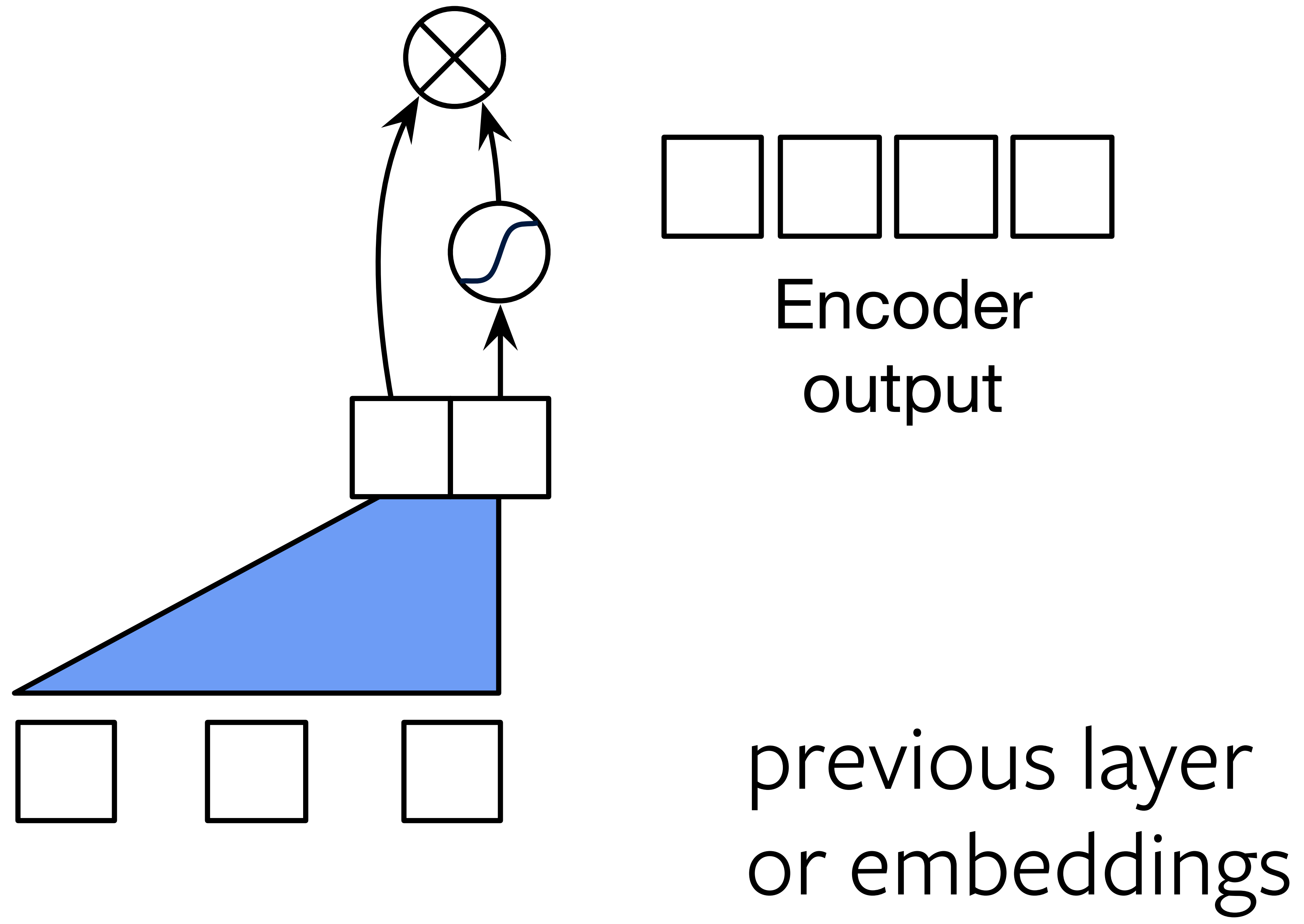
Convolutional S2S: Encoder

- Similar to Dauphin et al. ’17
- 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



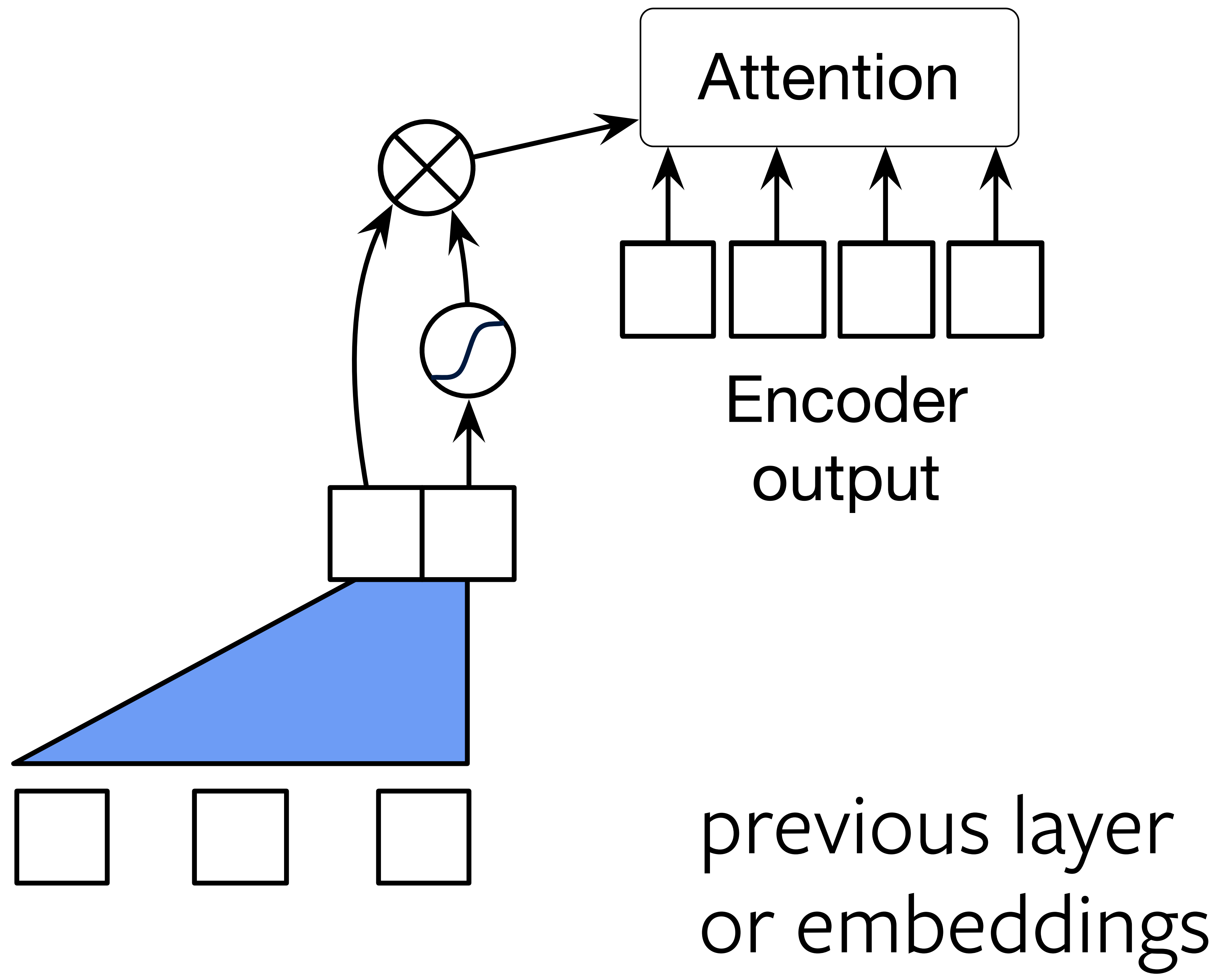
Convolutional S2S: Decoder

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



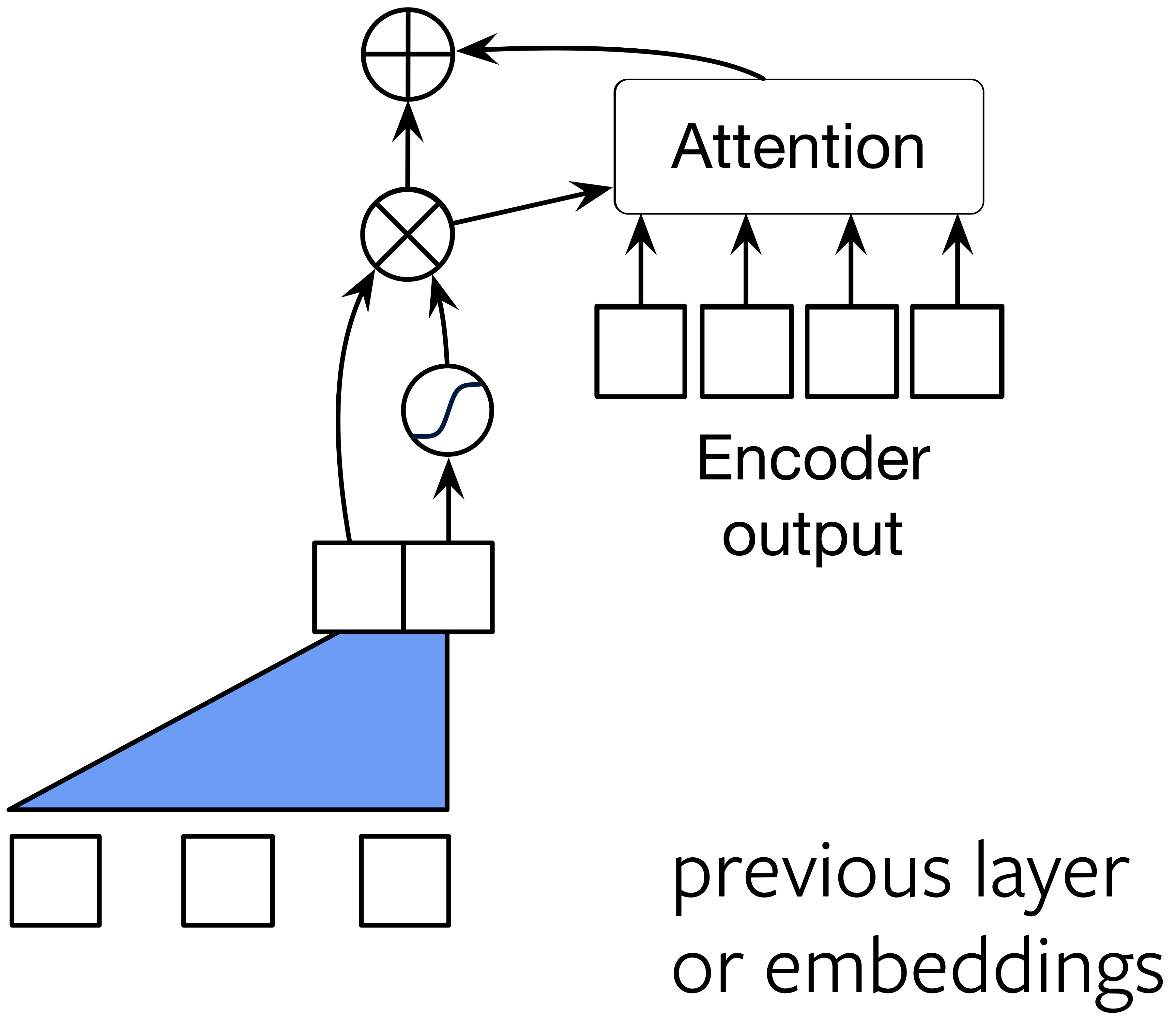
Convolutional S2S: Decoder

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



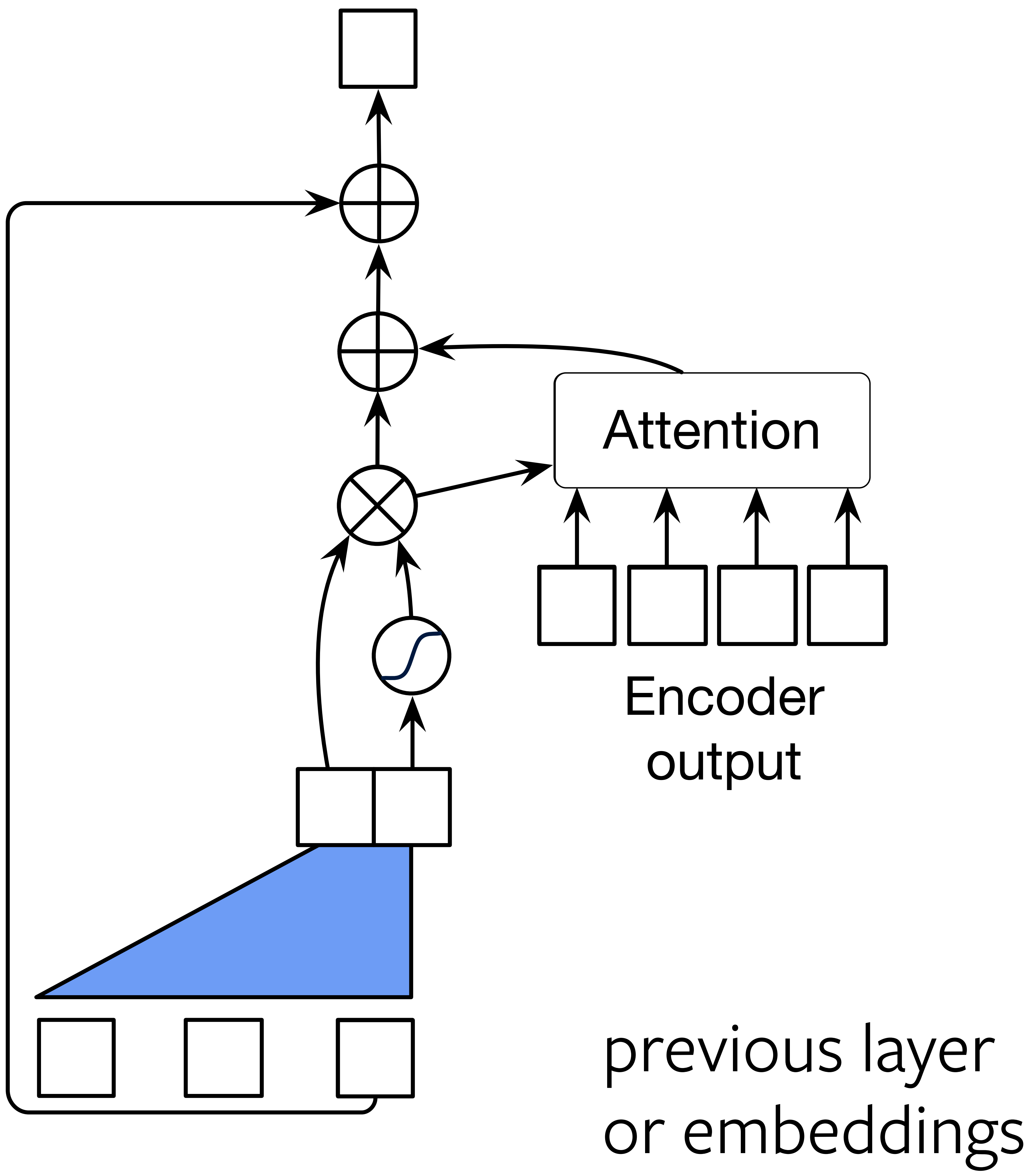
Convolutional S2S: Decoder

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



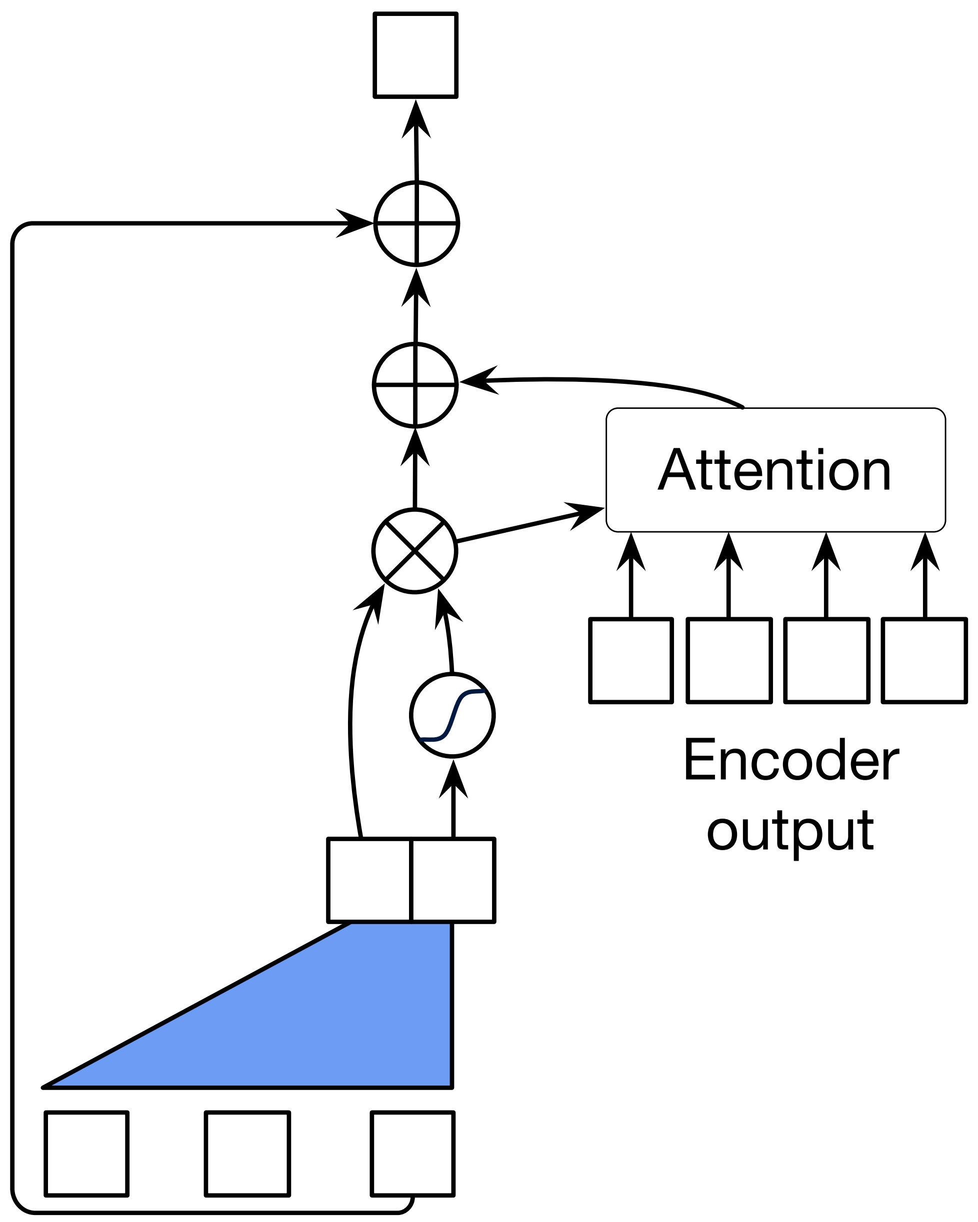
Convolutional S2S: Decoder

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



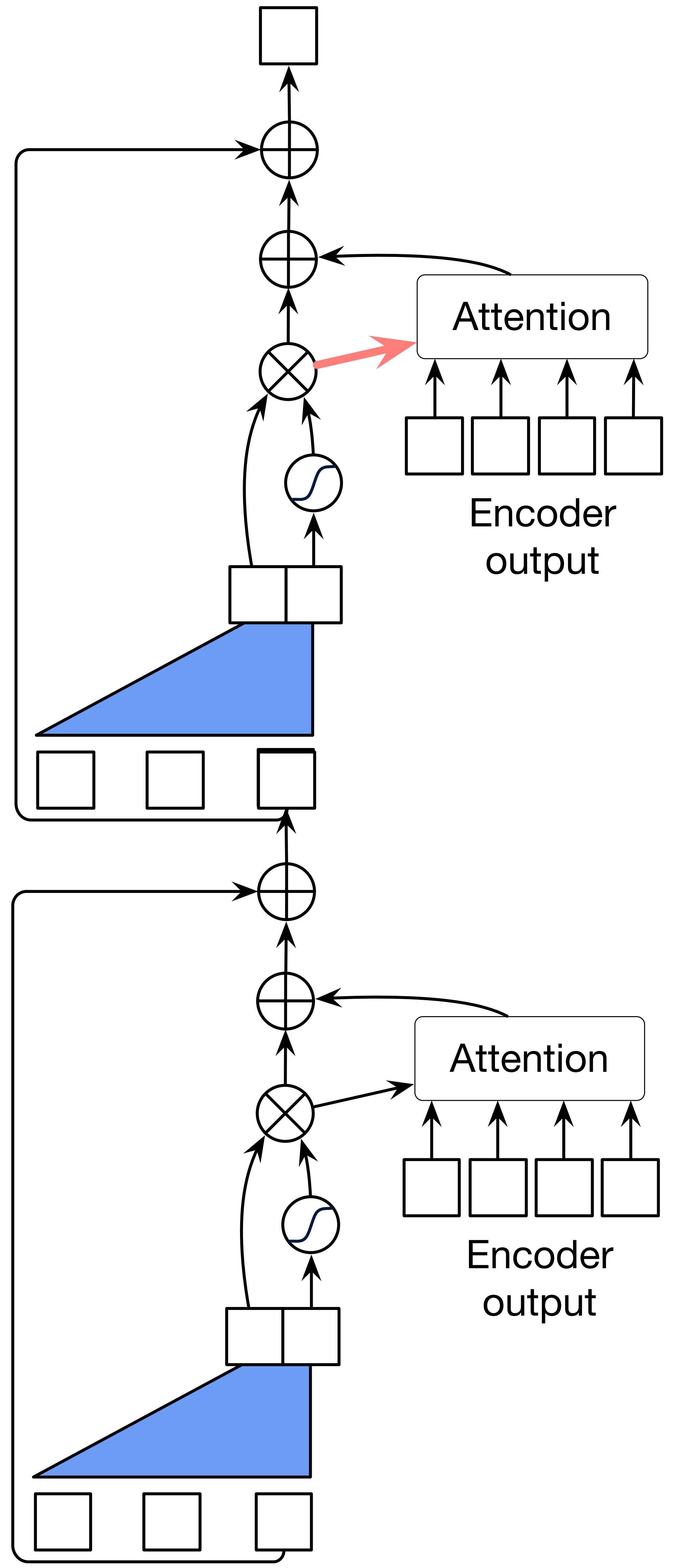
Convolutional S2S: Multi-hop Attention

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



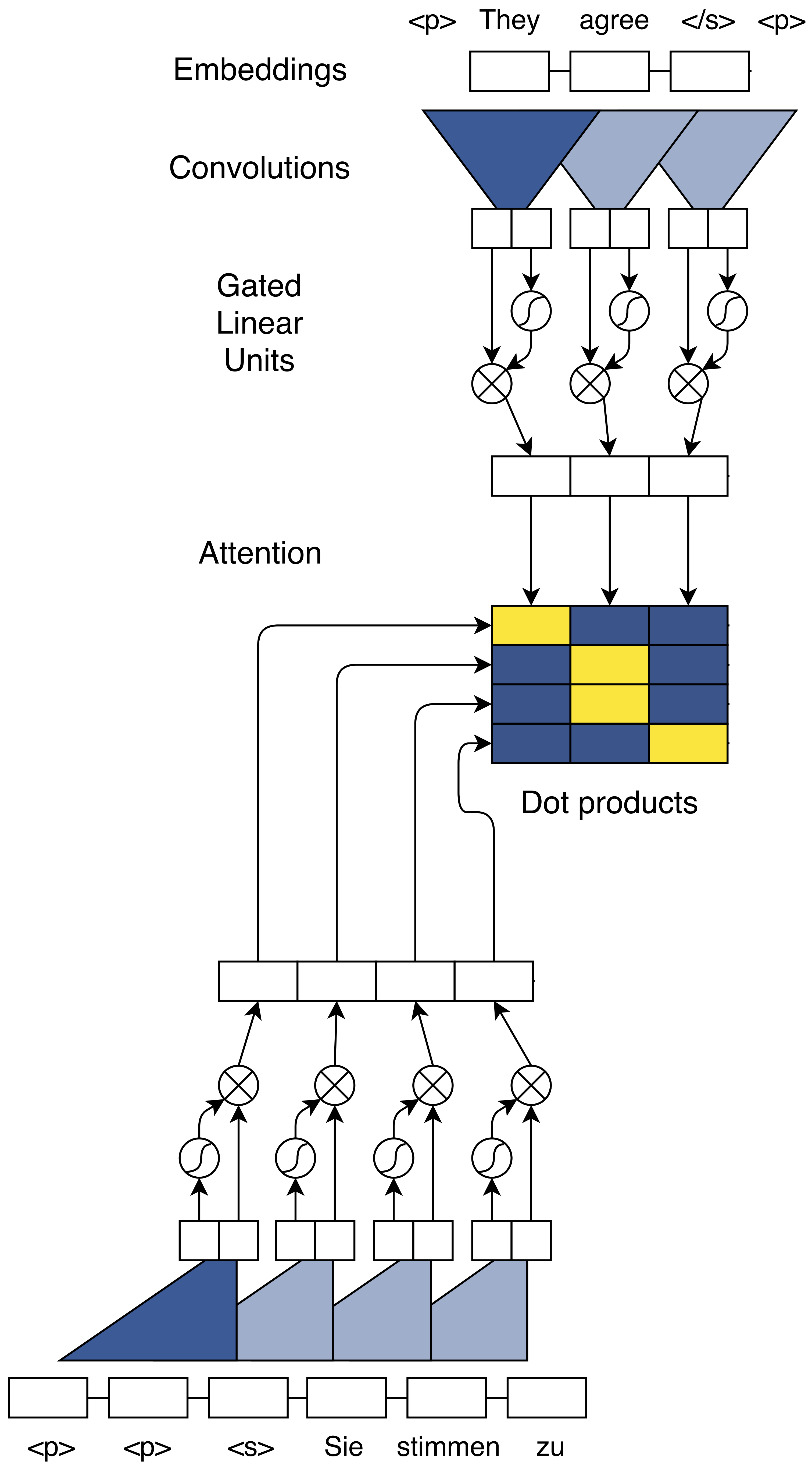
Convolutional S2S: Multi-hop Attention

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



Convolutional S2S

- High training efficiency due to parallel computation in decoder
- Cross Entropy objective
- Very similar to ResNet models (Nesterov etc., He et al. '15)



Experimental Methodology

- Translation & Summarization
- Large-scale tasks:
  - WMT'14 English-German (4.5M sentence pairs)
  - WMT'14 English-French (36M sentence pairs)
  - IWSLT'14 German-English (< 0.2M sentence pairs)

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

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

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

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

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)

Text Summarization

Compress first sentence of a news article into a headline (Rush et al. 2016)

	Rouge-1	Rouge-2	Rouge-L
RNN Likelihood optimized (Shen et al. 2016)	32.7	15.2	30.6
RNN Rouge-optimized (Shen et al. 2016)	36.5	16.6	33.4
RNN repeated words (Suzuki & Nagata, 2017)	36.3	17.3	33.9
ConvS2S	35.9	17.5	33.3

ConvS2S: 6 layers in encoder/decoder, nhid=256

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

Beam Search for Seq2Seq

Generating from Seq2Seq

At test time, we want to **generate** from $P(y|x)$ with $y \in \{1, \ldots, V\}^*$

- **sampling** is easy, decomposability allows left to right sampling
$$y \sim P(y|x) = \prod_t P(y_t|y_1^{t-1}, x)$$
- **MAP inference** is hard
$$\hat{y} = \text{argmax}_y P(y|x) \text{ with } y \in \{1, \ldots, V\}^*$$
- **Beam** approximates MAP inference

Beam Search

3 prefixes | **3 * V expansions** | **Top 3 expansions**

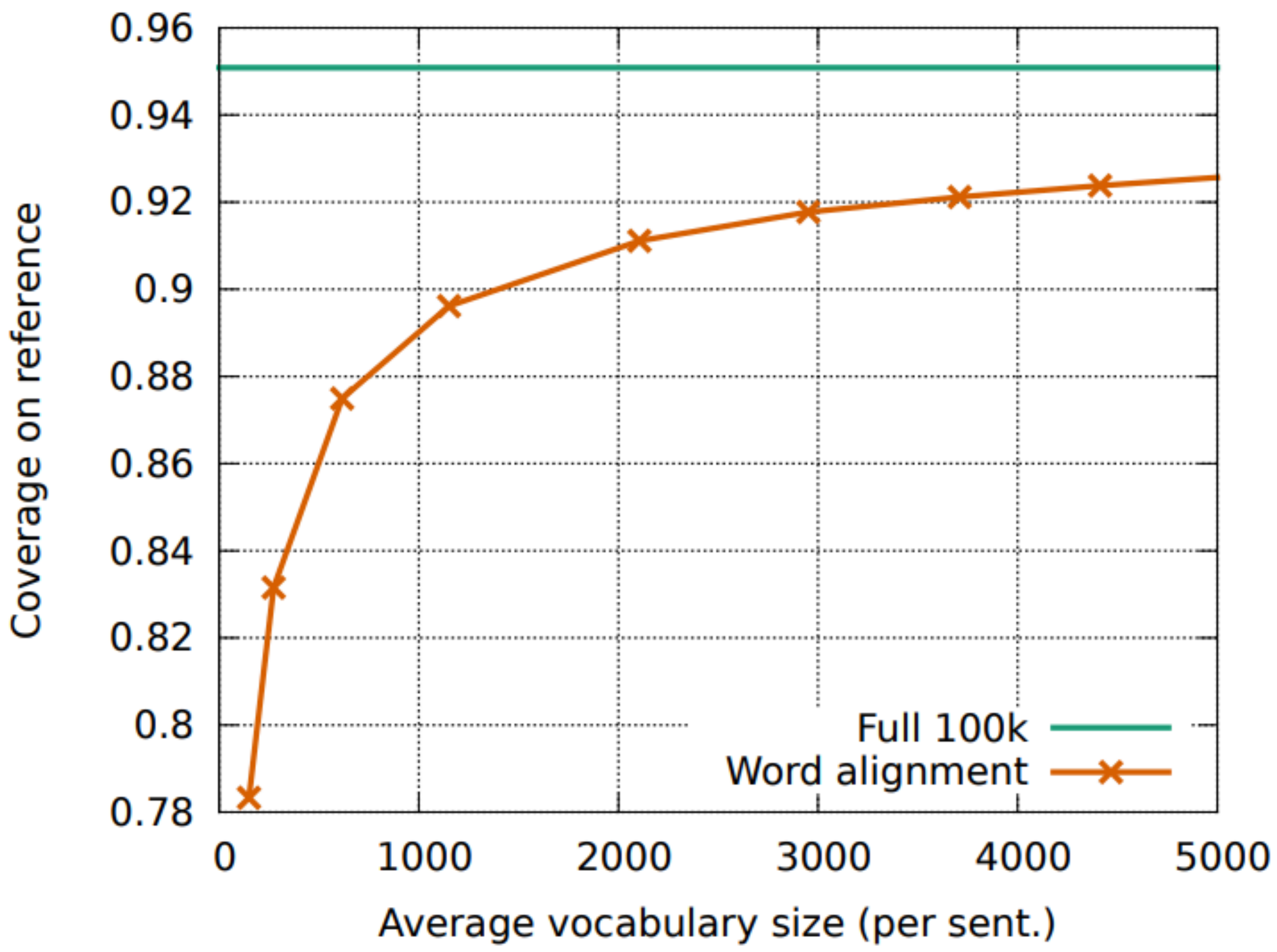
the little cat — jumps, runs

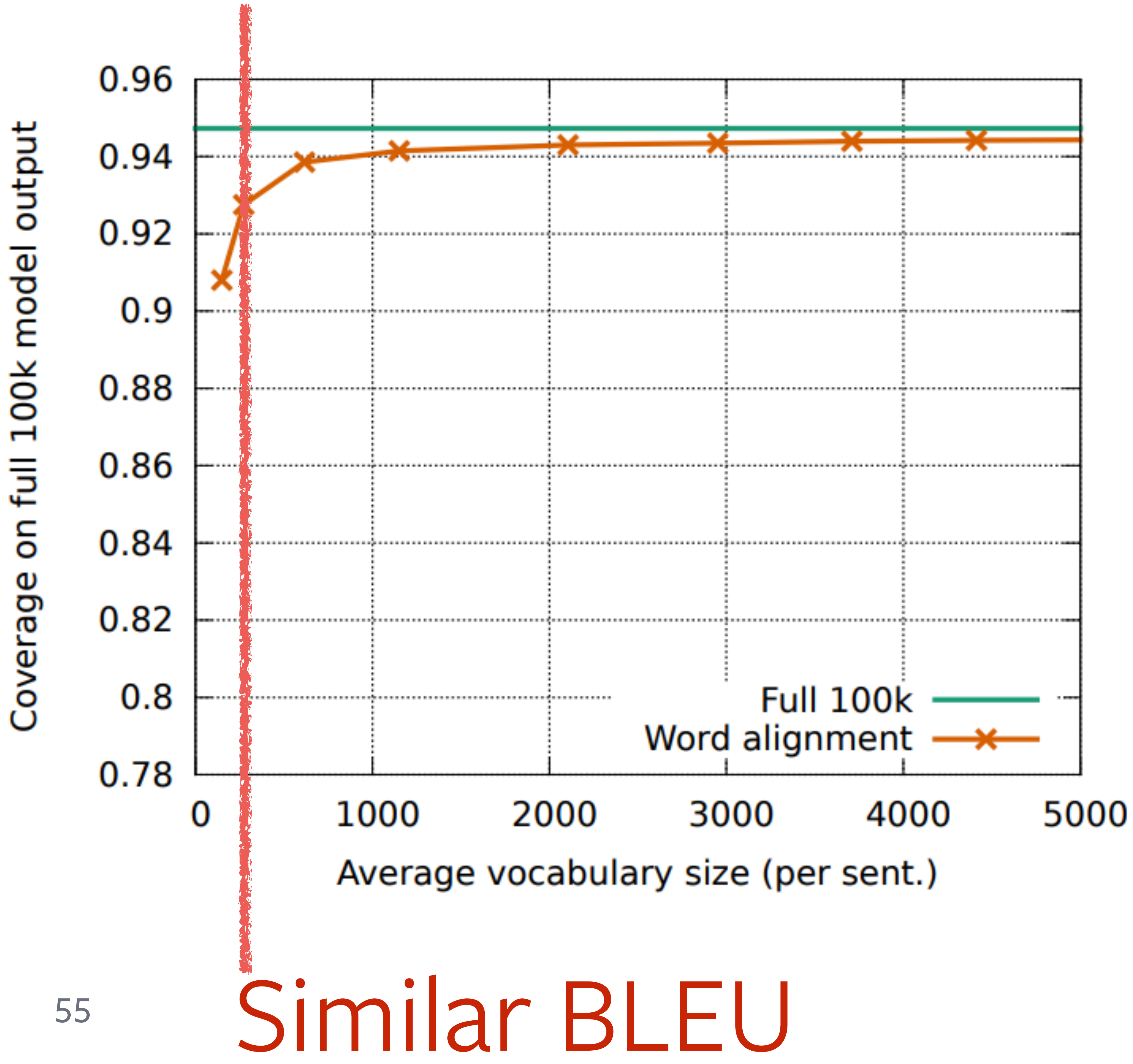
the tiny feline

the small kitten — jumps

On Search Space Size

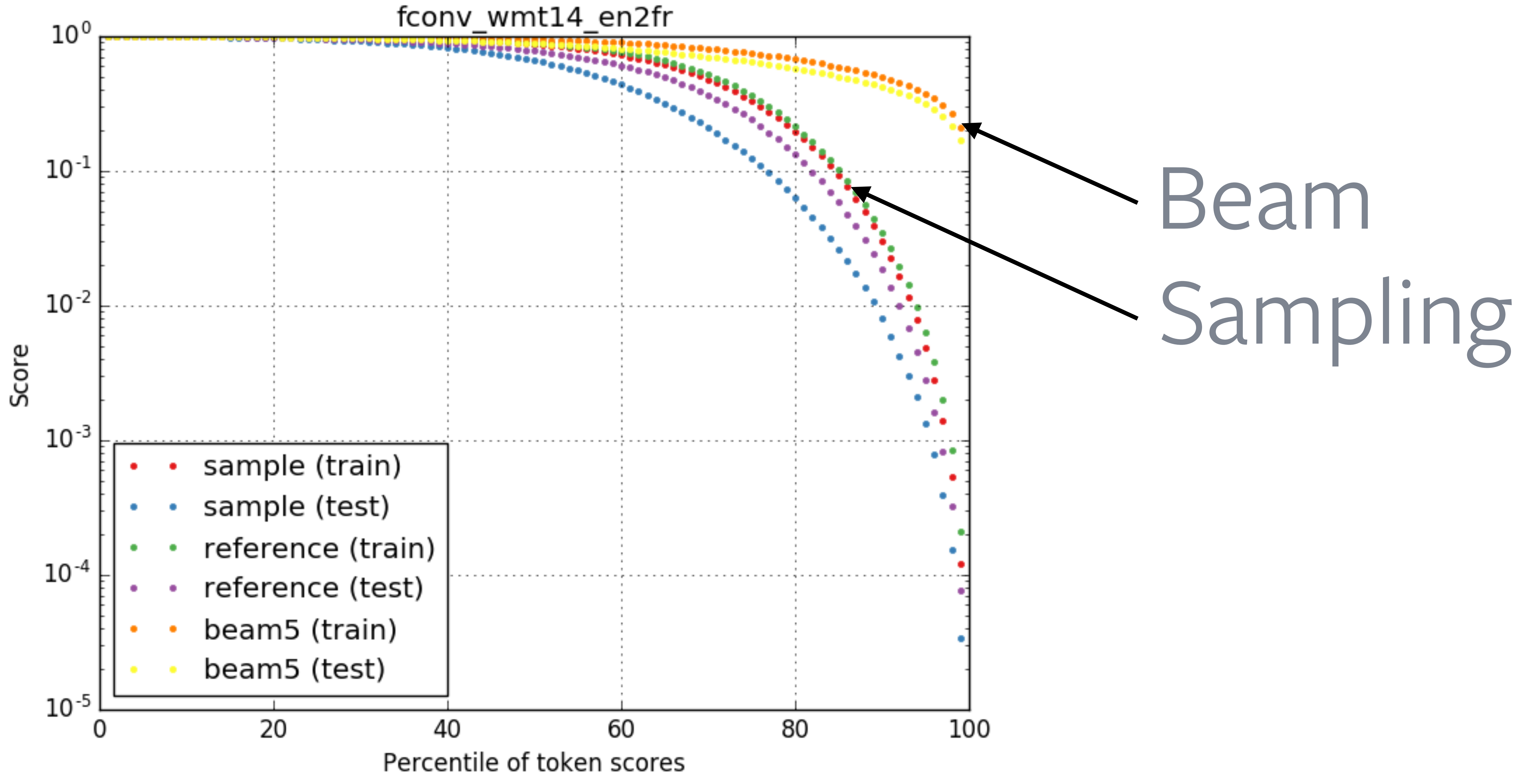
- **Effective vocabulary is rather small**
- e.g. vocabulary selection with alignment method (WMT’14 en-de)





On Search Space Size

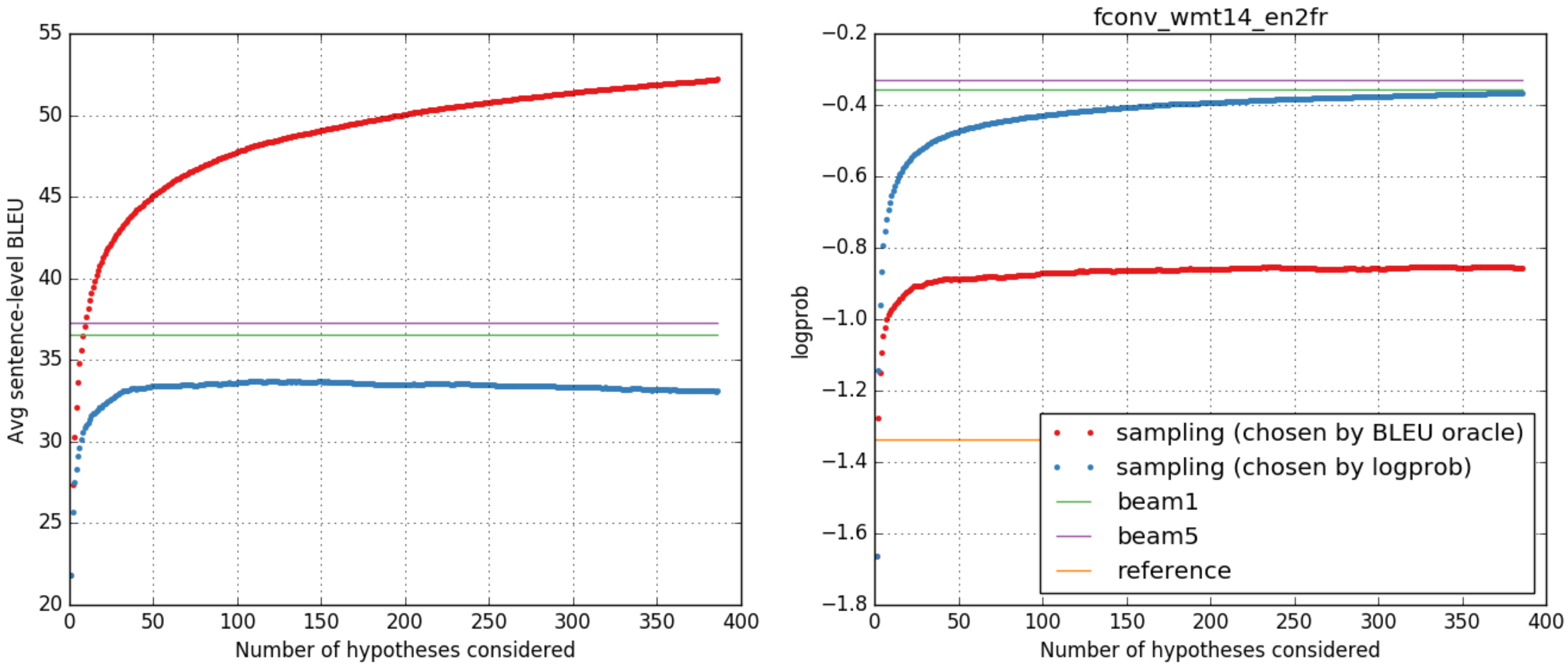
- beam results are prominent



- e.g. 20 time steps above 0.9 means more than $0.9^{20} \simeq \frac{1}{8}$

Beam Search

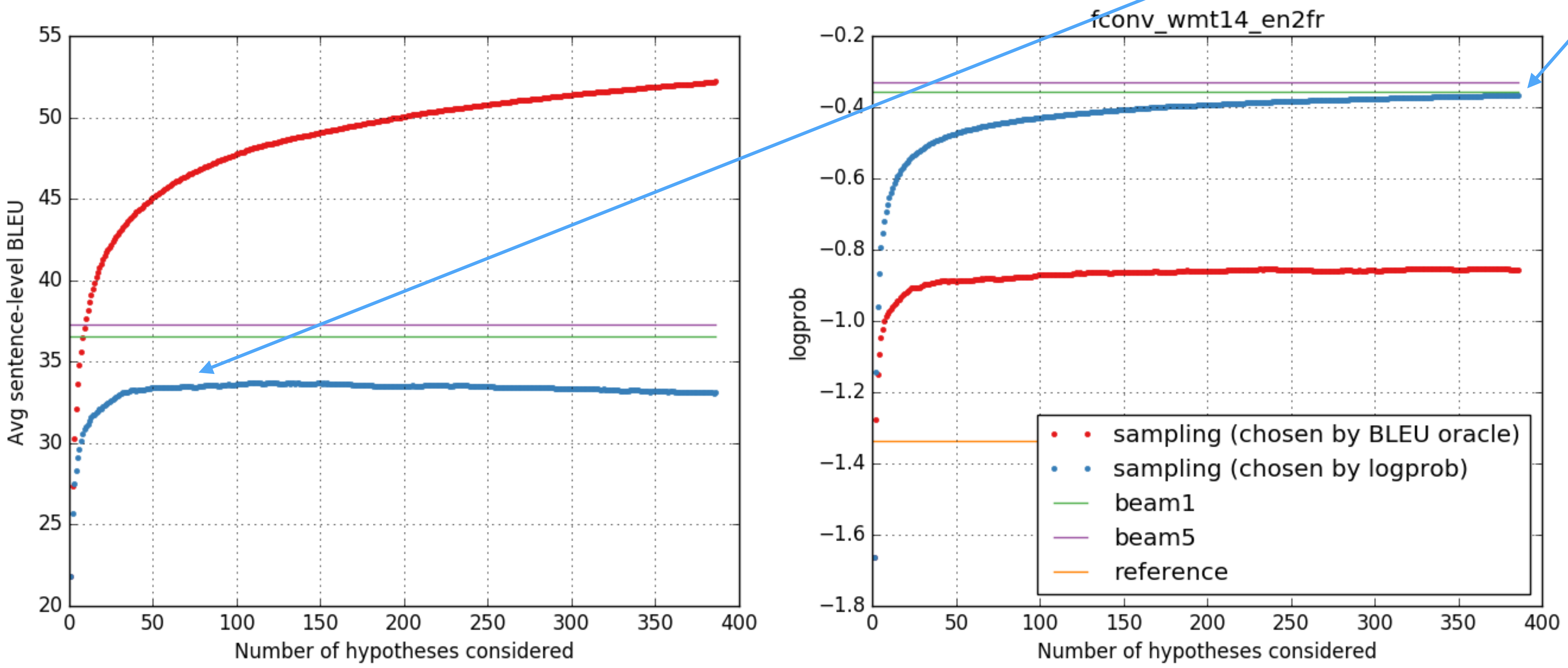
- Q: Does it work? good approximation of MAP?
- Q: Is it a good idea? Do we care about MAP?



Beam Search

- Q: Does it work? good approximation of MAP?
- Q: Is it a good idea? Do we care about MAP?

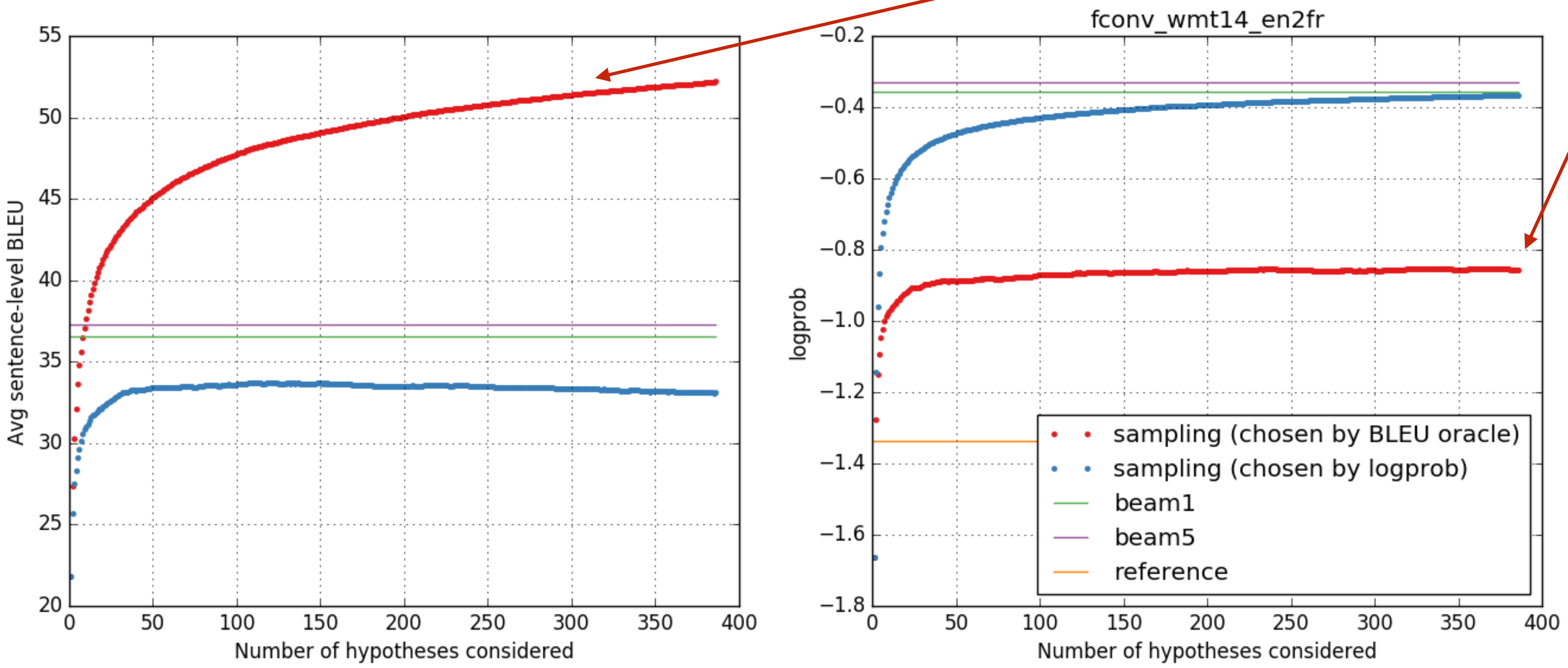
Sample k times & choose by logrpob



Beam Search

- Q: Does it work? good approximation of MAP?
- Q: Is it a good idea? Do we care about MAP?

Sample k times & choose by BLEU



Not a Search Problem?

Analyzing Neural MT Search and Model Performance

arXiv Aug'17

Jan Niehues, Eunah Cho, Thanh-Le Ha, Alex Waibel
Institute for Anthropomatics and Robotics
Karlsruhe Institute of Technology, Germany
{jan.niehues,eunah.cho,thanh-le.ha,alex.waibel}@kit.edu

n-best \ Model	Single	Ensemble	Oracle
Single	31.96	32.37	41.81
Ensemble	32.09	32.41	42.31
Union	31.95	32.39	44.55

+0.4

Table 1: Baseline: TED German→English

Better model = better estimation not better search

Sequence-Level Training

- Maximizing token likelihood does not consider BLEU

Sequence Level Training with Recurrent Neural Networks
Marc-Aurelio Ranzato, Sumit Chopra, Michael Auli, Wojciech Zaremba - Nov 2015
[Reinforce, Williams 92][expected BLEU]

Sequence-to-Sequence Learning as Beam-Search Optimization
Sam Wiseman, Alexander M. Rush - June 2016
[learning as search optimization (LaSO), Daume and Marcu 2005][search BLEU]

Google's Neural Machine Translation System
Wu et al - Oct 2016
[close to Ranzato'15 + new reward][expected GLEU]

An Actor-Critic Algorithm for Sequence Prediction
Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, J. Pineau, A. Courville, Y. Bengio - July ’16
[actor-critic, Sutton'84][expected BLEU]

Sequence-Level Training

IWSLT'14 de-en

	LL	Seq
Ranzato	20.3	21.9
Bahdanau	21.5	22.6
Wiseman	24.0	26.4
Conv	**30.2**	----

WMT'14 en-fr

	LL	LL-E	Seq	Seq-E
GNMT	39.0	40.3	39.9	41.2
Conv	40.5	**41.6**	----	----
Attn	41.0	----	----	----

WMT'14 en-de

	LL	LL-E	Seq	Seq-E
GNMT	24.7	26.2	24.6	26.3
Conv	25.2	26.4	----	----
Attn	**28.4**	----	----	----

small gains compared to

- architecture search
- ensembling

Sequence-Level Training

Pros

- optimize end loss

Cons

- slower, need joint loss and/or init
- optimize expectation, not MAP (except Wiseman+Rush)
- smaller gains than architecture search or ensembling

Conclusions

Gated Convolutional Architecture

- Gating allows for linear model initially
- Fast at train, inference
- Accurate for LM, MT

Training Seq2Seq for Machine Translation

- Most progress attributed to architecture & ensembling
- Less progress due to BLEU optimization

Future Work

- Domain adaptation
- Leverage monolingual data
  - e.g. study back-translation at scale
- Understand ensemble
- Model diversity, uncertainty

Questions?

Diolch Kiitos Sheun umesc Kasih Mamnoon Todah
Shnorhakalutiun
Shokriya
Gamsahapnida Te ekkür Dekuju/Dekujeme Hvala Ngiyabonga Dziękuję Shokrun Spaas Mul Ačí Xie
Dank Takk Cam Gra or al
Kop Salamat Merci Dankie Dhanyavaad Go Grazie Faleminderit
Dakujem Waad Daw Dhanyavaadaalu Dhanyavad Khopjai Kruthagnathalu
krap ThankYou kun Arigatou
Tack Shukriya Or Dhonnobaad
ederim Asante
Grazzi raibh Gracias Nandree Blagodariya Gomapsupnida Euxaristo Kun Hain Dhan
Fyrir Terima Enkosi Danke dank daa

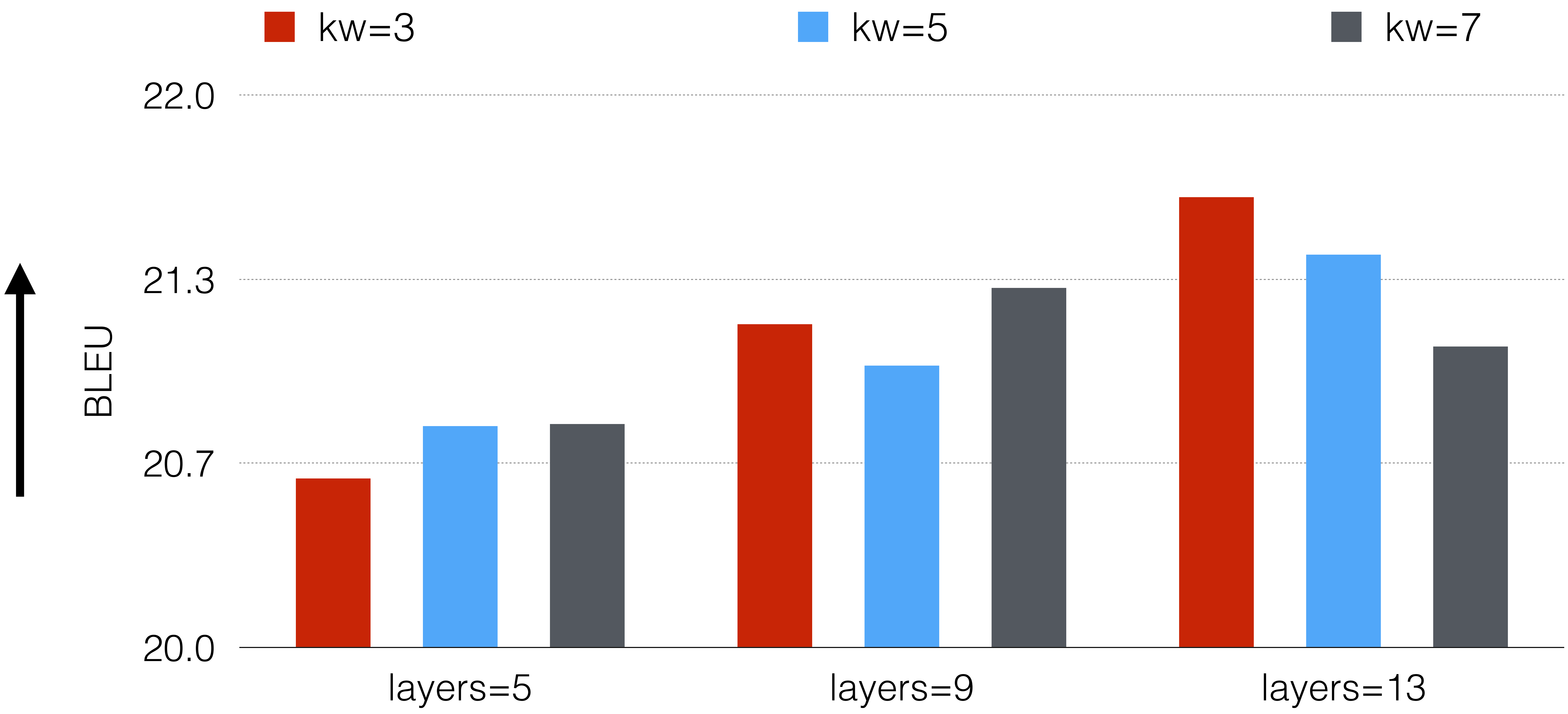
Results Multi-hop Attention

Attn Layers	PPL	BLEU
1,2,3,4,5	6.65	21.63
1,2,3,4	6.70	21.54
1,2,3	6.95	21.36
1,2	6.92	21.47
1,3,5	6.97	21.10
1	7.15	21.26
2	7.09	21.30
3	7.11	21.19
4	7.19	21.31
5	7.66	20.24

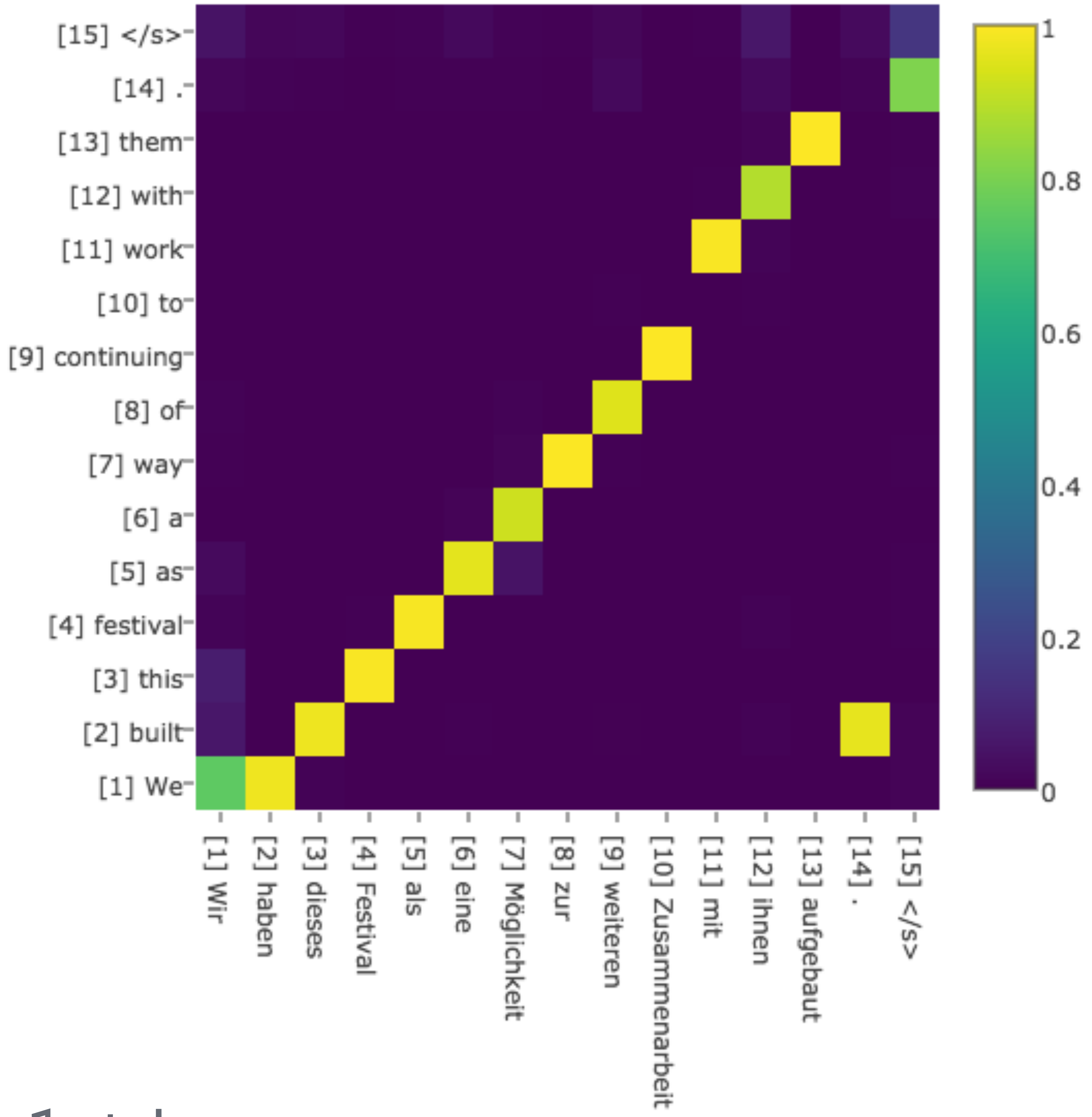
Ensemble En-Fr

WMT'14 English-French	BLEU
Zhou et al. (2016)	40.4
Wu et al. (2016) GNMT	40.35
Wu et al. (2016) GNMT + RL	41.16
ConvS2S	41.44
ConvS2S (10 models)	41.62

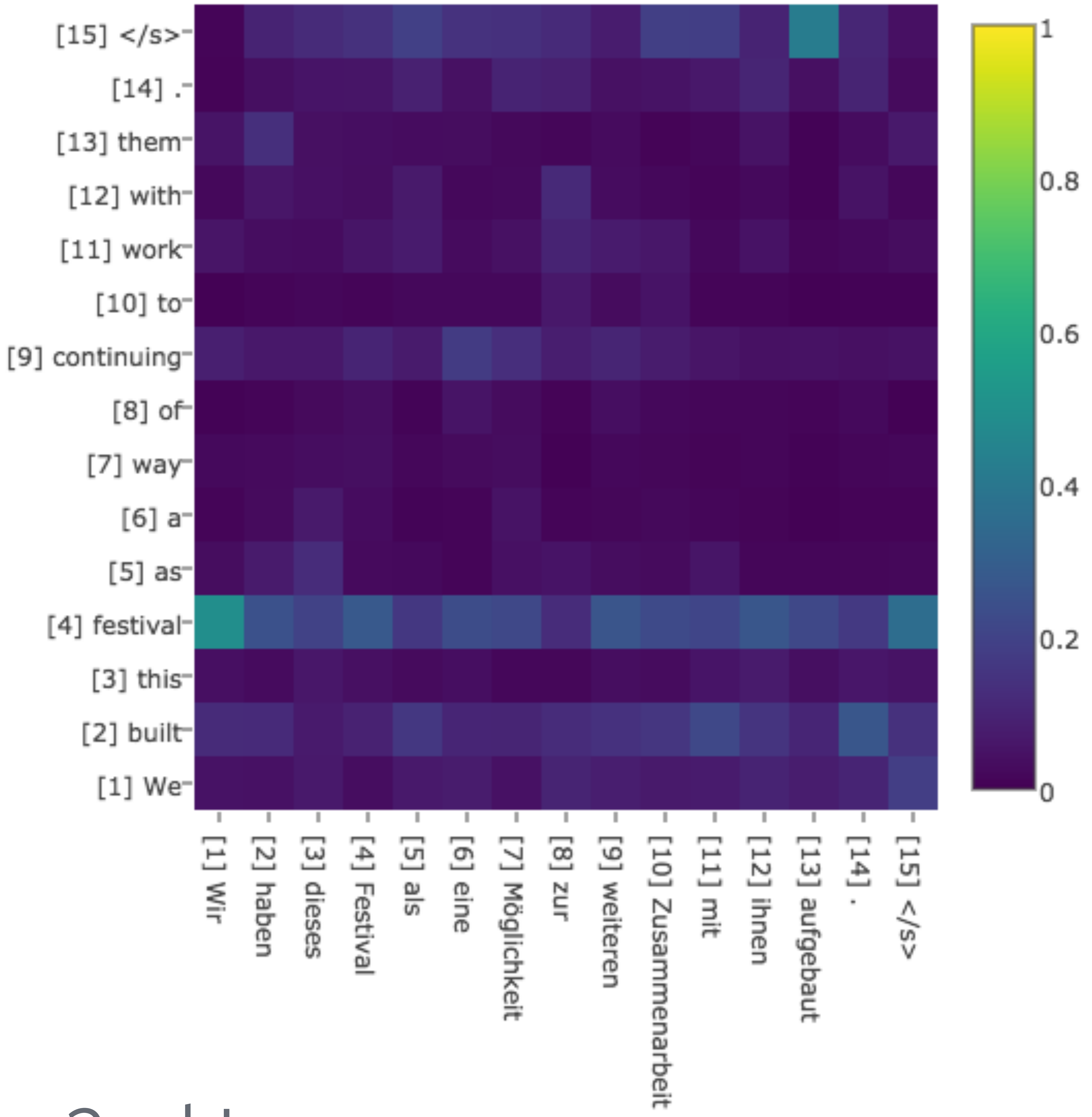
Network Depth



Convolutional S2S: Multi-Hop Attention

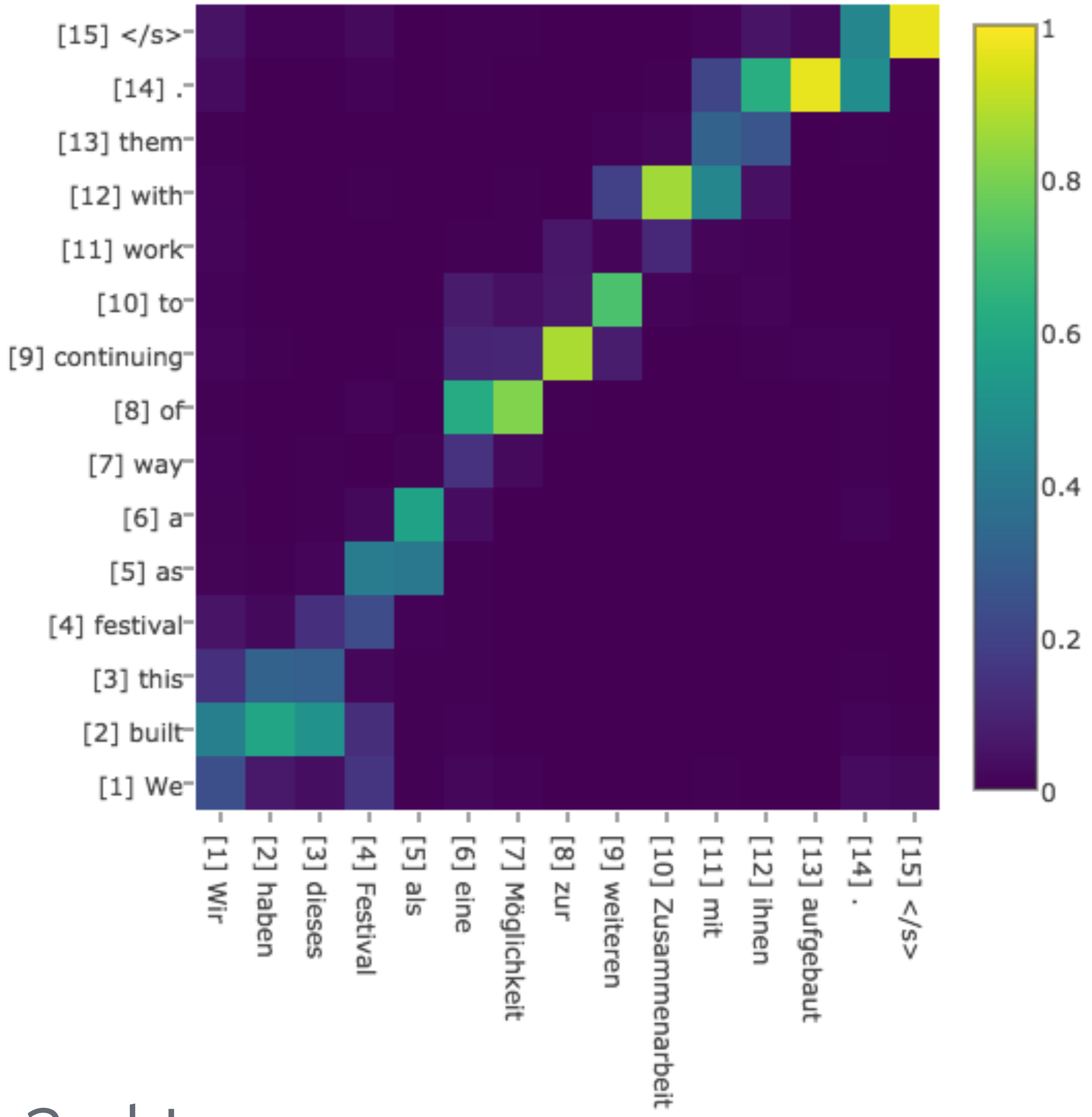


1st Layer

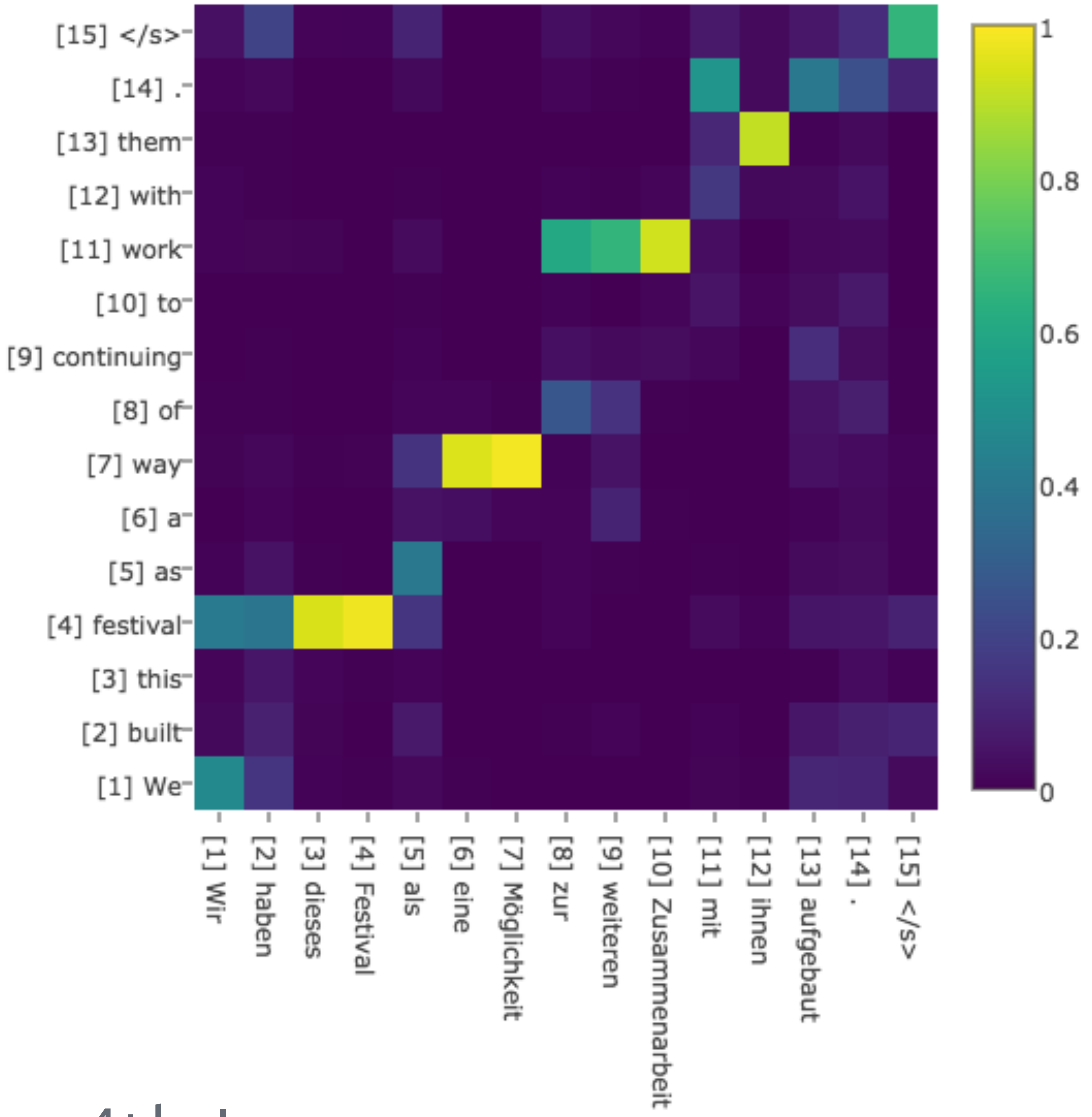


2nd Layer

Convolutional S2S: Multi-Hop Attention

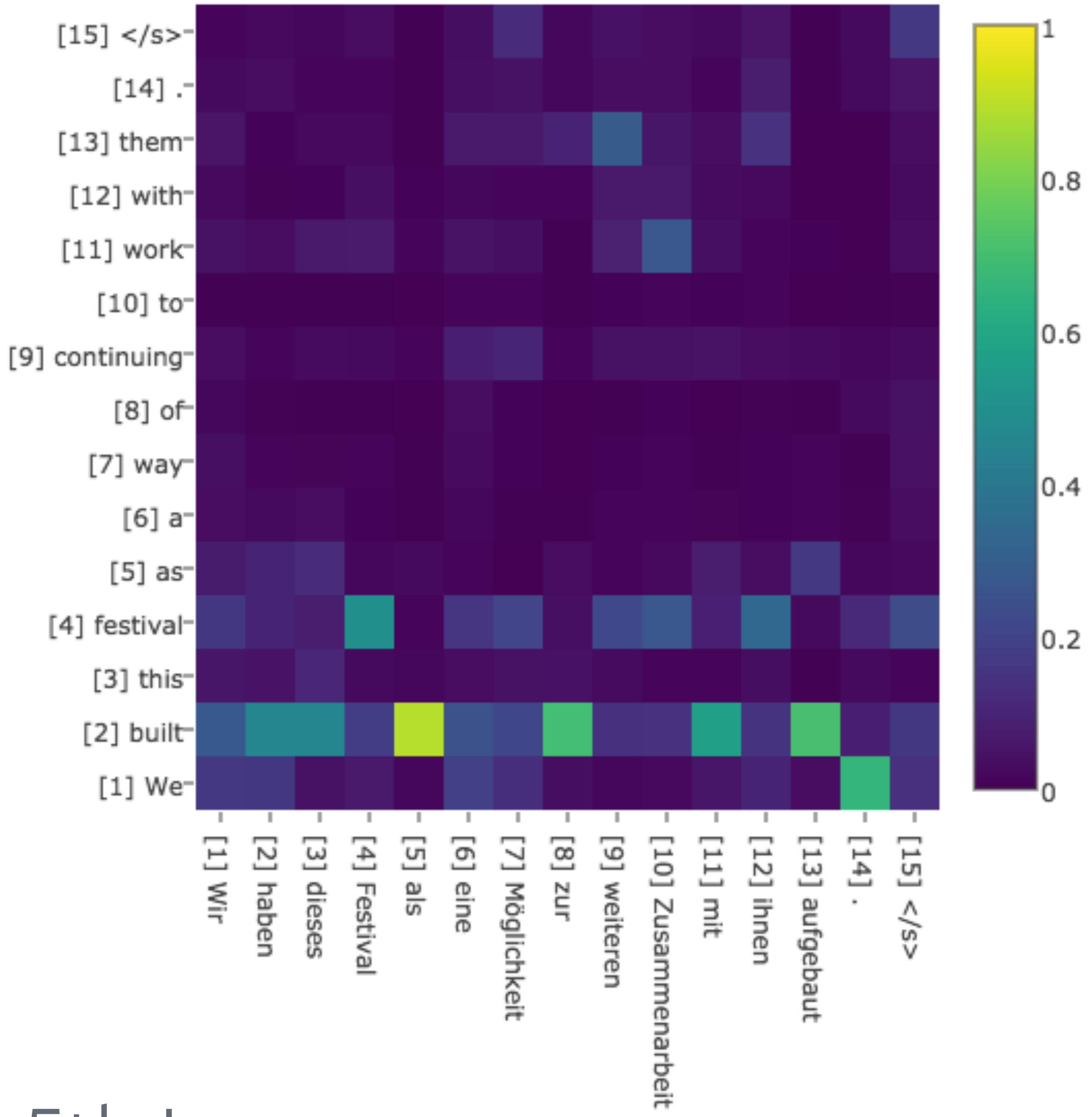


3rd Layer

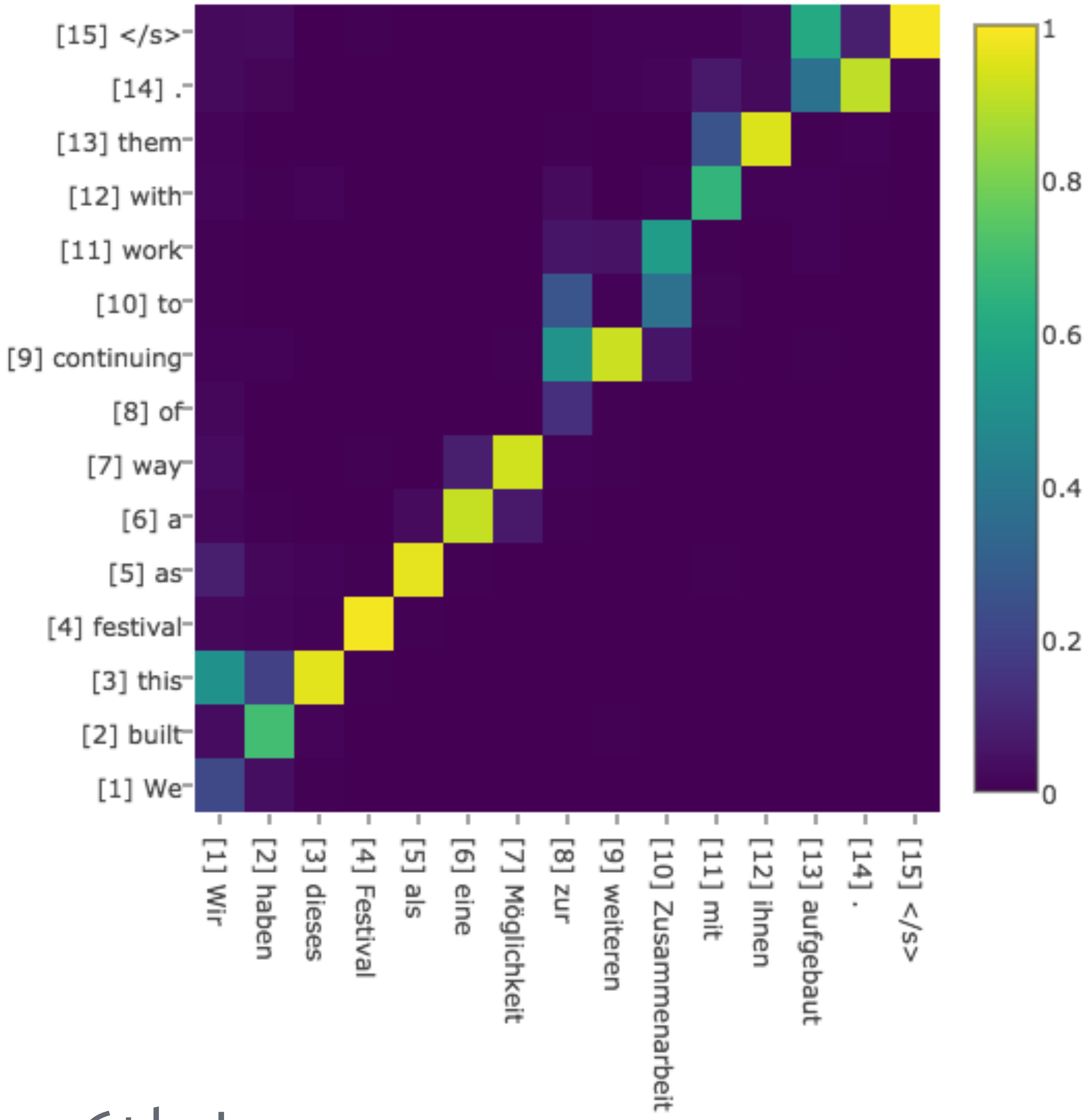


4th Layer

Convolutional S2S: Multi-Hop Attention

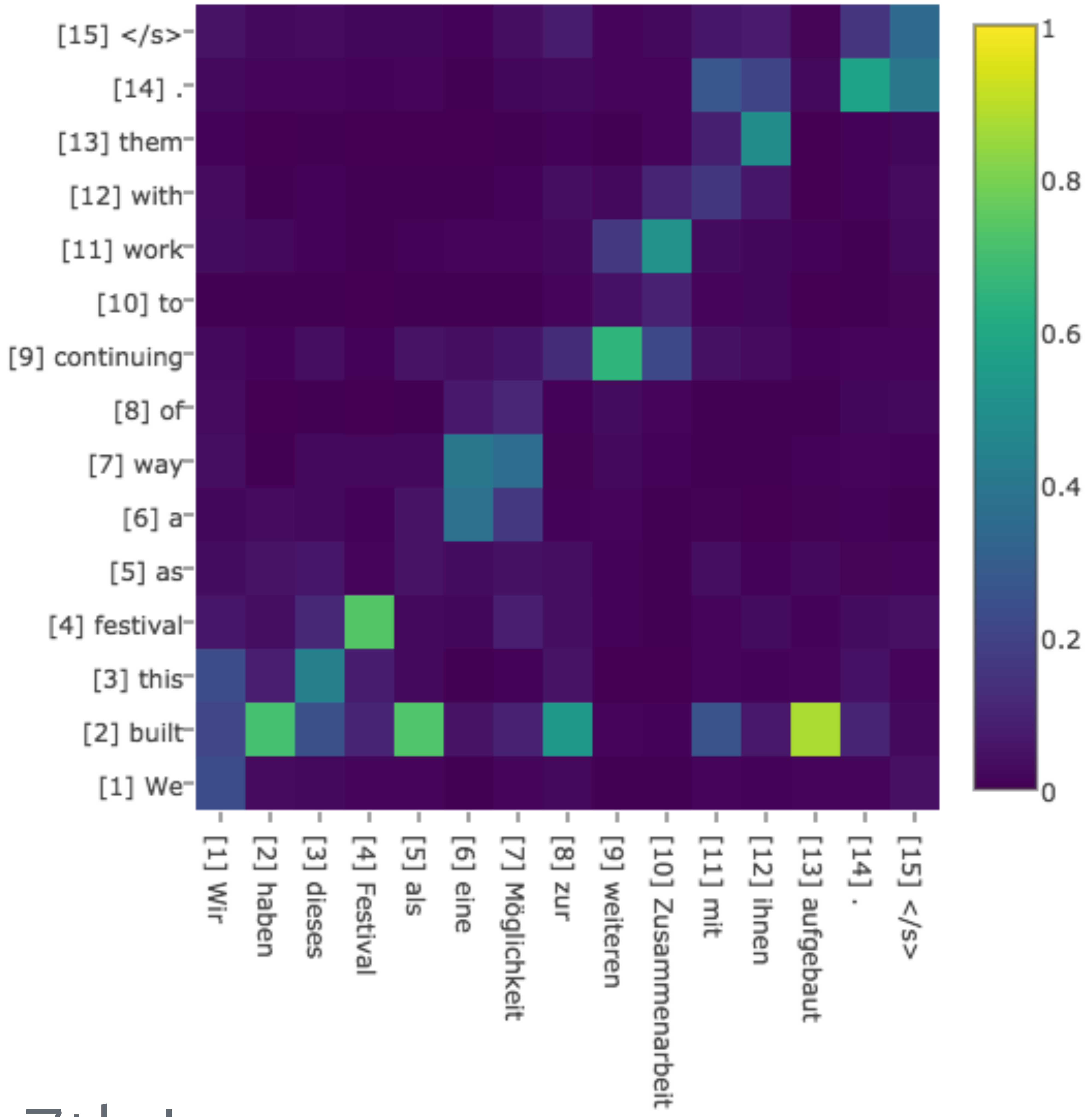


5th Layer

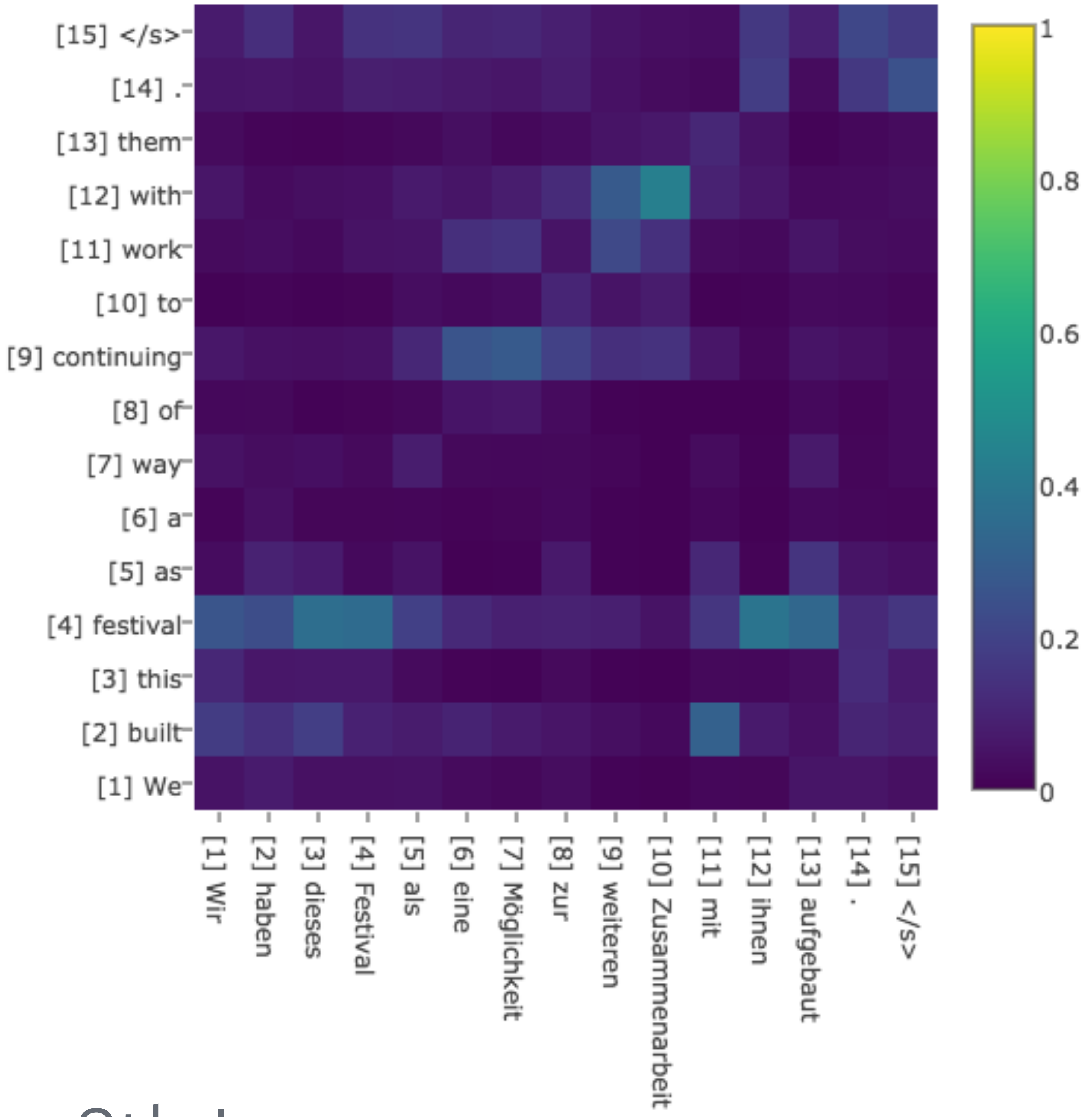


6th Layer

Convolutional S2S: Multi-Hop Attention



7th Layer



8th Layer