Massively Multilingual Neural Machine Translation

Roee Aharoni
NLP Lab, Bar Ilan University

Joint work with Melvin Johnson, Orhan Firat
Google Translate

DL Course

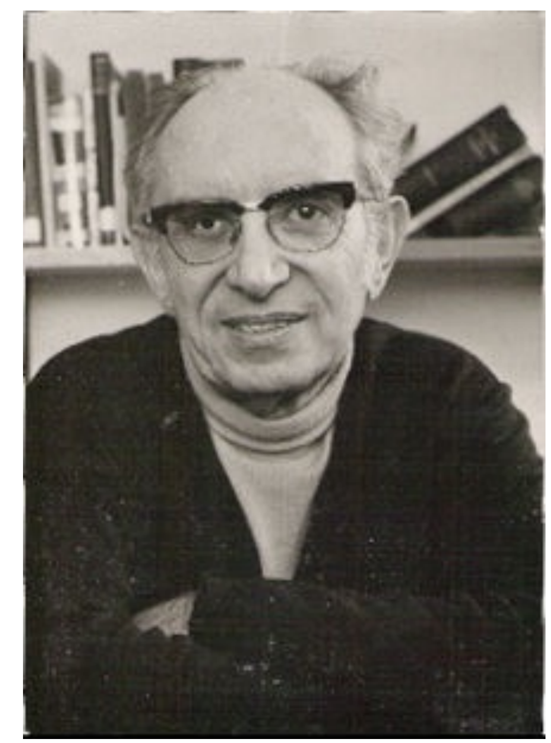
Google Translate

B I U
N L P

Some (Israeli) History

Some (Israeli) History

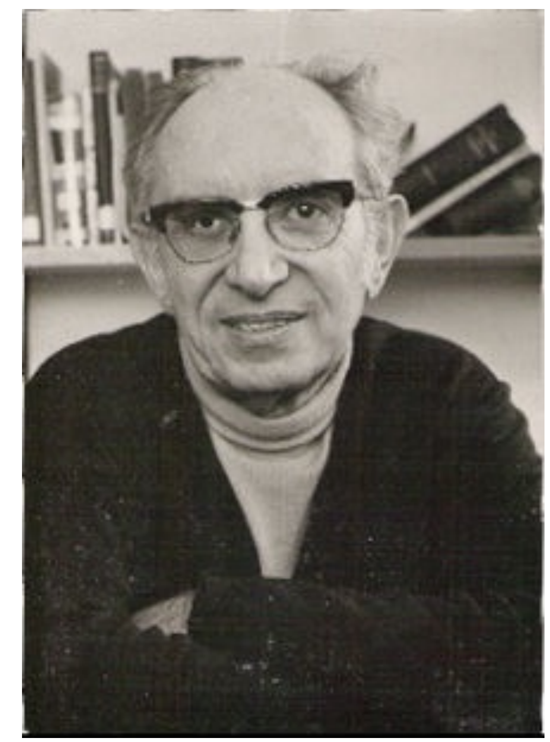
- “Since thinking in terms of machines might perhaps be difficult for the reader, let him imagine an **utterly moronic student** without the slightest knowledge of either the source-language or the target-language…” Bar Hillel, 1953



Some (Israeli) History

- “Since thinking in terms of machines might perhaps be difficult for the reader, let him imagine an **utterly moronic student** without the slightest knowledge of either the source-language or the target-language…” Bar Hillel, 1953

- **Yehoshua Bar Hillel** from the Hebrew University/MIT was the first academic to work full-time on Machine Translation. He organized the first “International Conference on Machine Translation” in 1952. (He also fought with the Haganah, losing an eye)

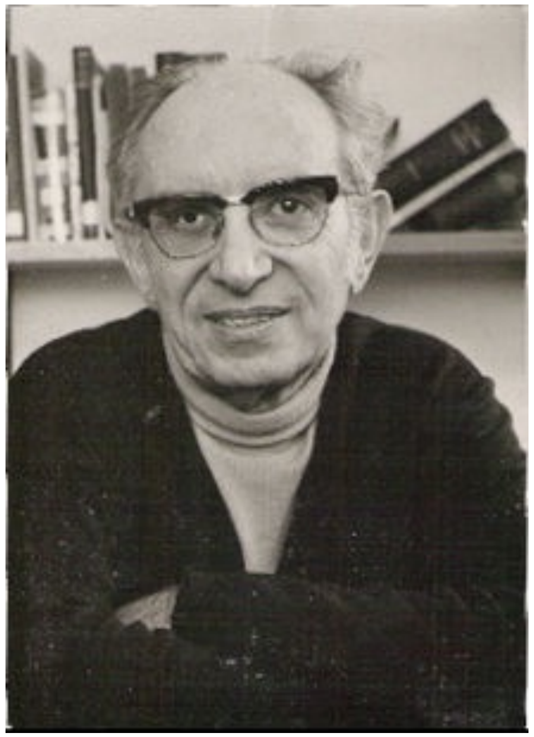


Some (Israeli) History

- “Since thinking in terms of machines might perhaps be difficult for the reader, let him imagine an **utterly moronic student** without the slightest knowledge of either the source-language or the target-language…” Bar Hillel, 1953

- **Yehoshua Bar Hillel** from the Hebrew University/MIT was the first academic to work full-time on Machine Translation. He organized the first “International Conference on Machine Translation” in 1952. (He also fought with the Haganah, losing an eye)

- Wer’e trying to make computers translate for more than 70 years!



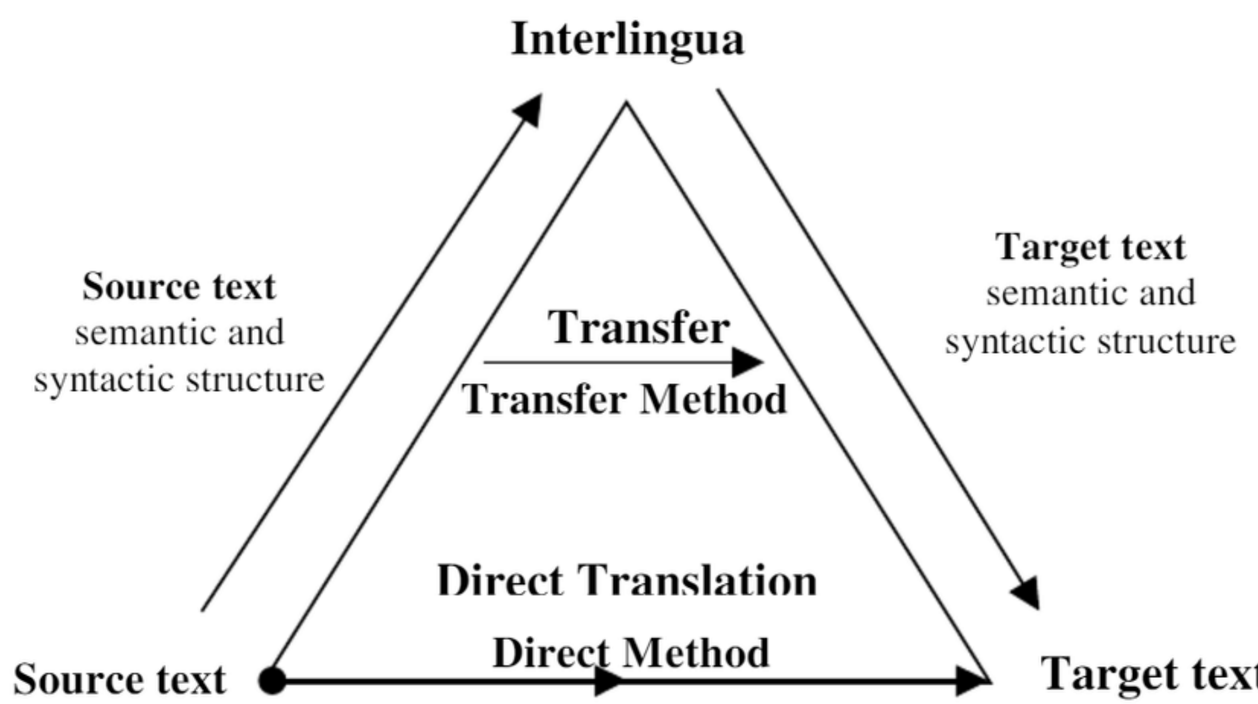
Some (More) History

Some (More) History

- **Bernard Vauquois** has been one of the pioneers of machine translation from 1960 until his death in 1985.

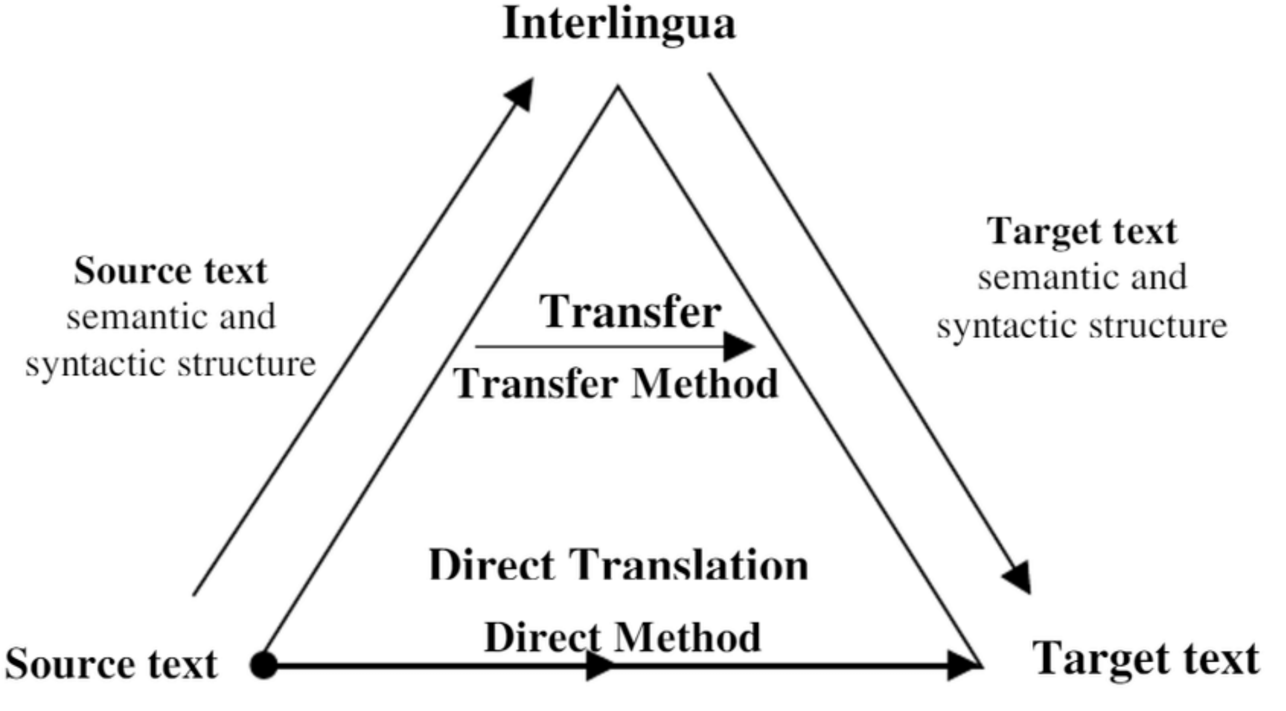
Some (More) History

- **Bernard Vauquois** has been one of the pioneers of machine translation from 1960 until his death in 1985.
- He is known for the **"Vauquois Triangle"** which described possible pipelines for (rule-based) MT



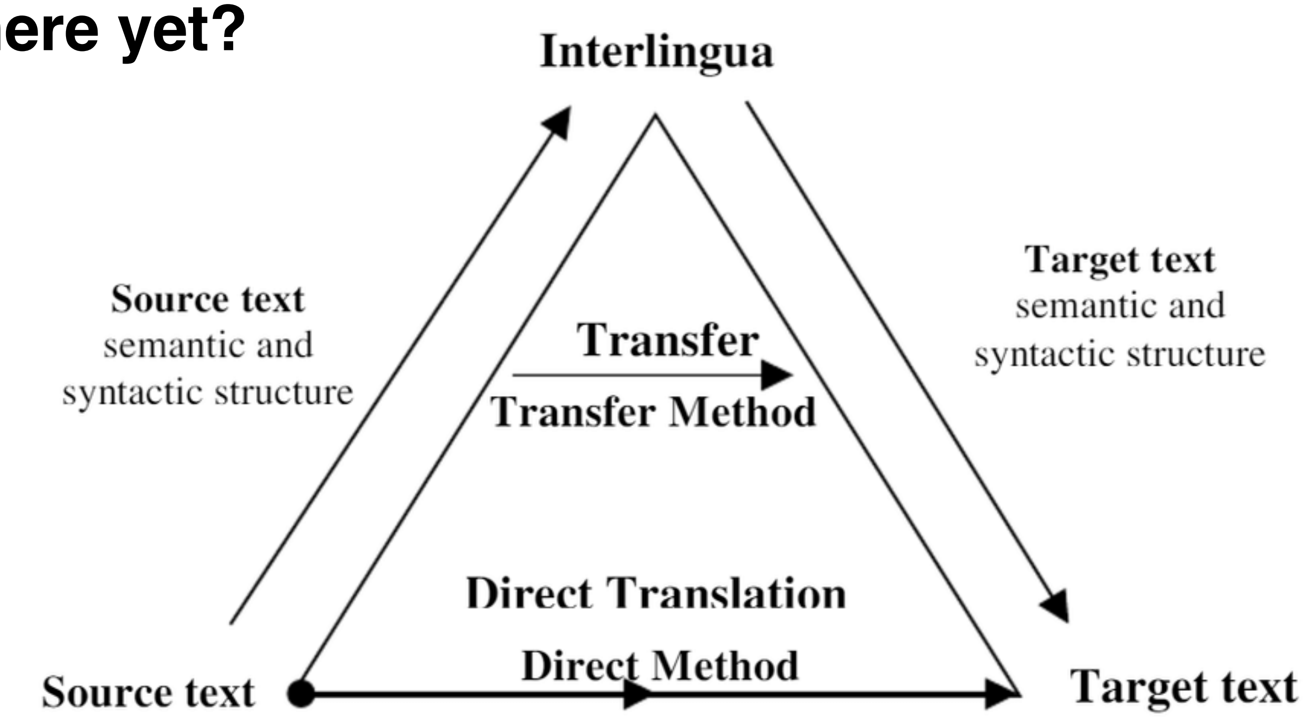
Some (More) History

- **Bernard Vauquois** has been one of the pioneers of machine translation from 1960 until his death in 1985.
- He is known for the **"Vauquois Triangle"** which described possible pipelines for (rule-based) MT
- Proposed an "Interlingua" stage which can be shared between multiple languages



Some (More) History

- **Bernard Vauquois** has been one of the pioneers of machine translation from 1960 until his death in 1985.
- He is known for the **"Vauquois Triangle"** which described possible pipelines for (rule-based) MT
- Proposed an "Interlingua" stage which can be shared between multiple languages
  - **Are we there yet?**



The (Statistical) Machine Translation Objective

The (Statistical) Machine Translation Objective

- We want to find the best translation **f** given a source sentence **e:**

The (Statistical) Machine Translation Objective

- We want to find the best translation **f** given a source sentence **e:**

$$e = \text{(Economic, growth, has, slowed, down, in, recent, years, .)}$$

$$f = \text{(La, croissance, économique, s'est, ralentie, ces, dernières, années, .)}$$

The (Statistical) Machine Translation Objective

- We want to find the best translation **f** given a source sentence **e:**

$$e = \text{(Economic, growth, has, slowed, down, in, recent, years, .)}$$

$$f = \text{(La, croissance, économique, s'est, ralentie, ces, dernières, années, .)}$$

$$f = \underset{f'}{\operatorname{argmax}}\, p(f'|e)$$

The (Statistical) Machine Translation Objective

- We want to find the best translation **f** given a source sentence **e:**

$$e = (\text{Economic, growth, has, slowed, down, in, recent, years, .})$$

$$f = (\text{La, croissance, économique, s'est, ralentie, ces, dernières, années, .})$$

$$f = \underset{f'}{\operatorname{argmax}}\, p(f'|e)$$

- How do we estimate $p(f'|e)$ from data?

The (Statistical) Machine Translation Objective

- We want to find the best translation **f** given a source sentence **e:**

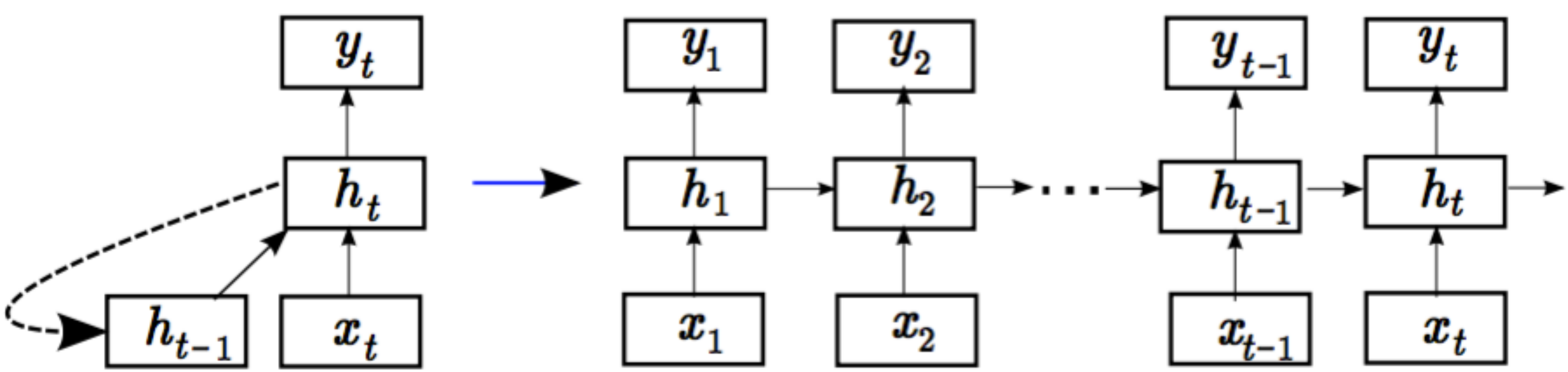
$$e = (\text{Economic, growth, has, slowed, down, in, recent, years, .})$$

$$f = (\text{La, croissance, économique, s'est, ralentie, ces, dernières, années, .})$$

$$f = \underset{f'}{\operatorname{argmax}}\, p(f'|e)$$

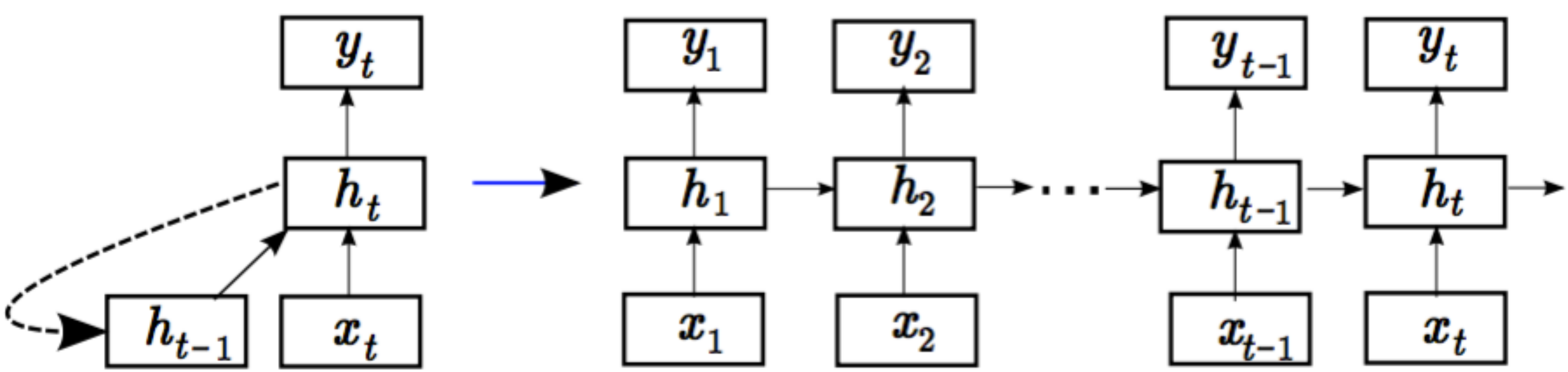
- How do we estimate $p(f'|e)$ from data?
- First, lets recap on…

Recurrent Neural Networks (RNN’s)



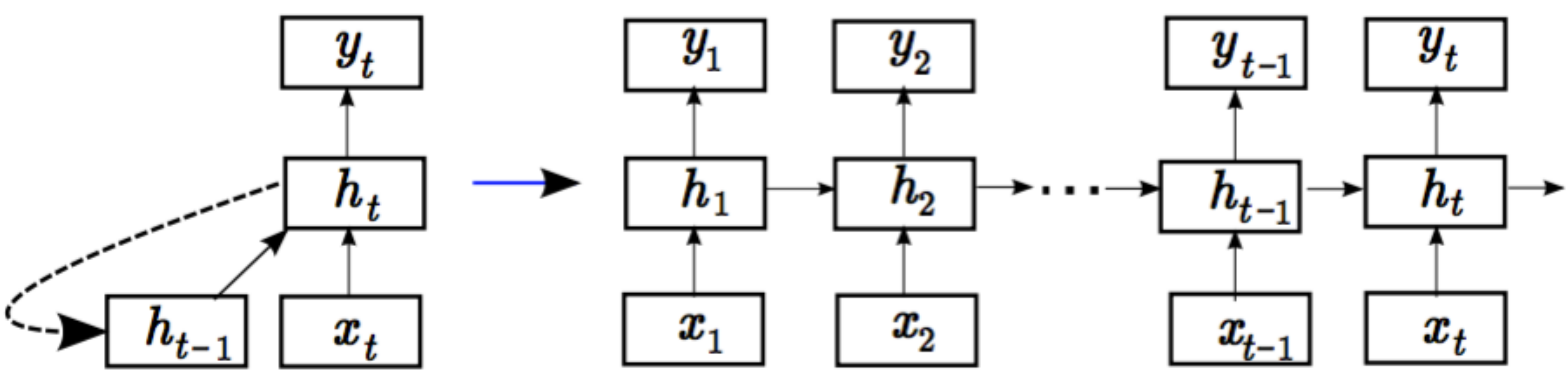
Recurrent Neural Networks (RNN's)

- "Horizontally deep" architecture



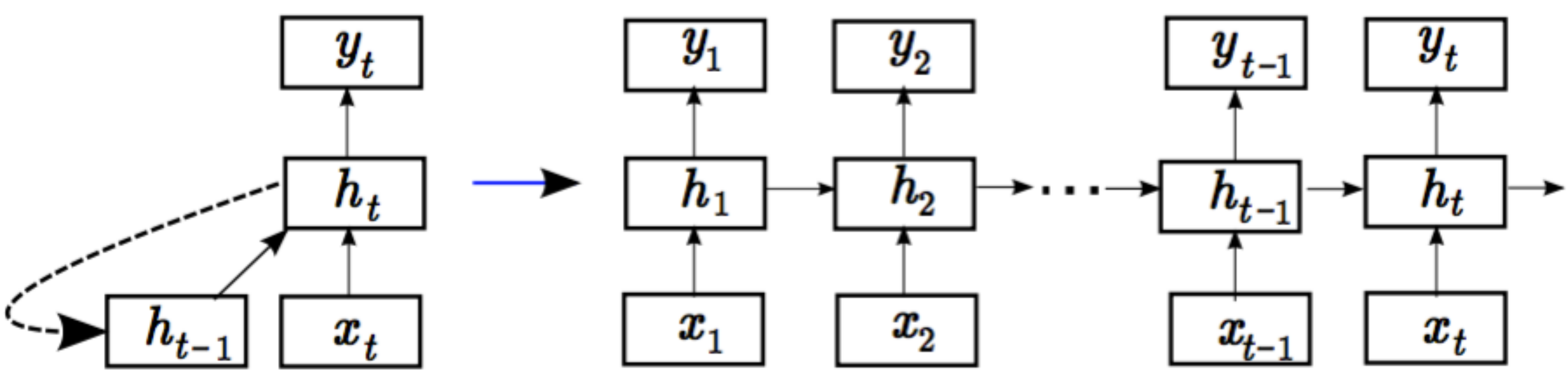
Recurrent Neural Networks (RNN's)

- "Horizontally deep" architecture
- Recurrence equations:



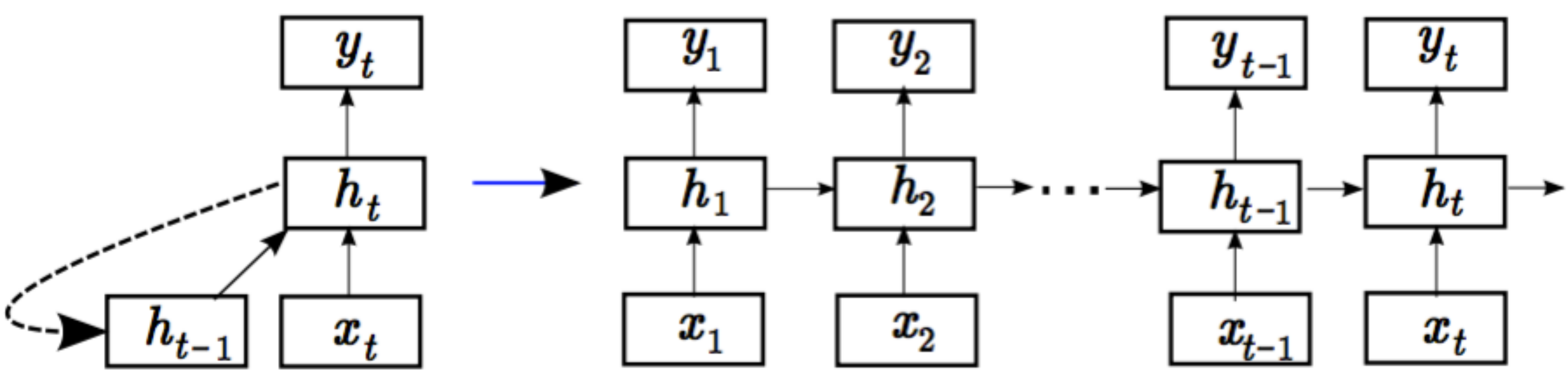
Recurrent Neural Networks (RNN's)

- "Horizontally deep" architecture
- Recurrence equations:
  - Transition function: $h_t = H(h_{t-1}, x_t) = tanh(Wx_t + Uh_{t-1} + b)$



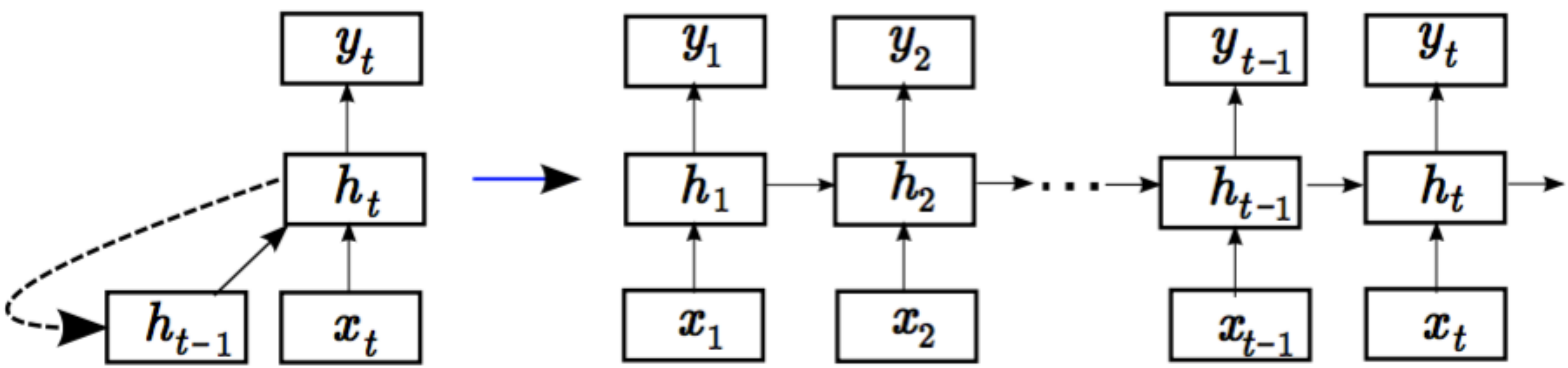
Recurrent Neural Networks (RNN's)

- "Horizontally deep" architecture
- Recurrence equations:
  - Transition function: $h_t = H(h_{t-1}, x_t) = tanh(Wx_t + Uh_{t-1} + b)$
  - Output function: $y_t = Y(h_t)$



Recurrent Neural Networks (RNN's)

- "Horizontally deep" architecture
- Recurrence equations:
  - Transition function: $h_t = H(h_{t-1}, x_t) = tanh(Wx_t + Uh_{t-1} + b)$
  - Output function: $y_t = Y(h_t)$
  - How can we predict a sentence with an RNN?



The Softmax Function & Negative Log Loss

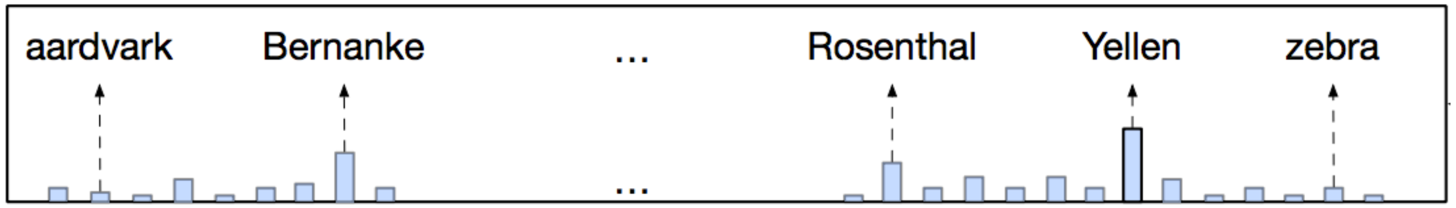
The Softmax Function & Negative Log Loss

- Enables to output a **probability distribution** over **k possible classes** (words, in our case)

The Softmax Function & Negative Log Loss

- Enables to output a **probability distribution** over **k possible classes** (words, in our case)
- y_i (the network output vector in position i) is expected to hold the log-likelihood (probability) for a specific class (in our case, word):

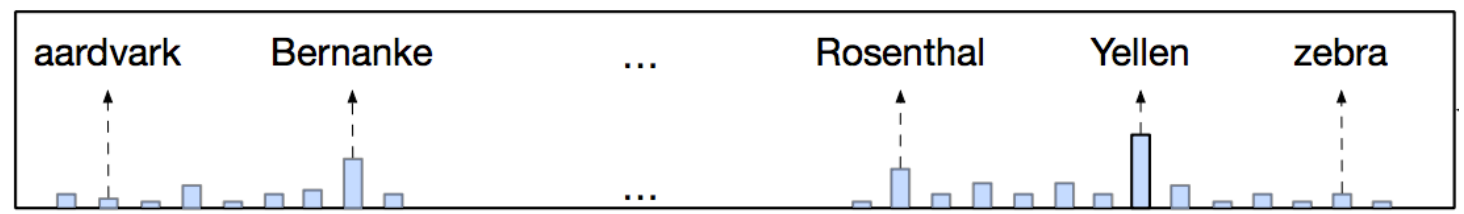
$$p(x = i) = \frac{e^{y_i}}{\sum_{j=1}^{k} e^{y_j}}$$



The Softmax Function & Negative Log Loss

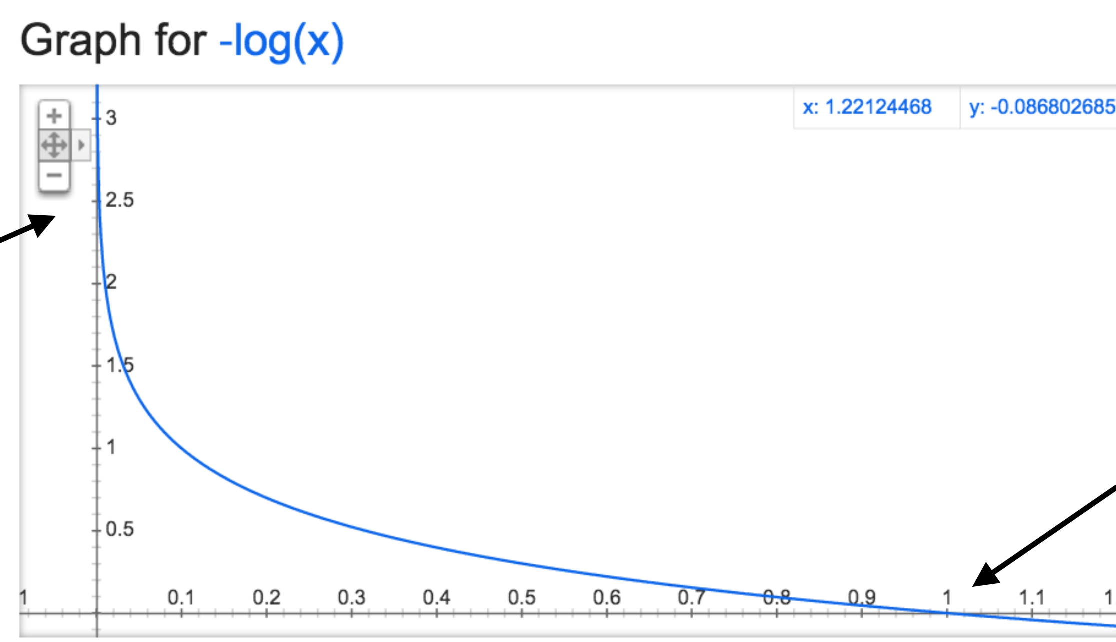
- Enables to output a **probability distribution** over **k possible classes** (words, in our case)
- y_i (the network output vector in position i) is expected to hold the log-likelihood (probability) for a specific class (in our case, word):

$$p(x=i) = \frac{e^{y_i}}{\sum_{j=1}^{k} e^{y_j}}$$



- The network's loss function is usually the sum of **negative log softmax** values for the **correct sequence**

wrong prediction - large loss value



correct prediction - zero loss

Sequence 2 Sequence Learning

Sequence 2 Sequence Learning

- Inspired by RNN language modeling

Sequence 2 Sequence Learning

- Inspired by RNN language modeling
- First (modern) models for NMT presented by Kalchbrenner et. al. 2013, Sutskever et al., 2014, Cho et al., 2014

Sequence 2 Sequence Learning

- Inspired by RNN language modeling
- First (modern) models for NMT presented by Kalchbrenner et. al. 2013, Sutskever et al., 2014, Cho et al., 2014
- 2 RNN's, one for "reading" the input and one for "writing" the output (a.k.a the encoder-decoder architecture)

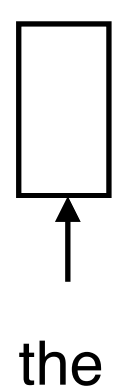
Sequence 2 Sequence Learning

- Inspired by RNN language modeling
- First (modern) models for NMT presented by Kalchbrenner et. al. 2013, Sutskever et al., 2014, Cho et al., 2014
- 2 RNN’s, one for “reading” the input and one for “writing” the output (a.k.a the encoder-decoder architecture)

Encoder

Sequence 2 Sequence Learning

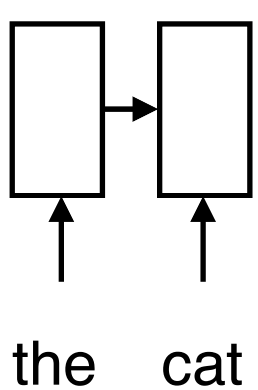
- Inspired by RNN language modeling
- First (modern) models for NMT presented by Kalchbrenner et. al. 2013, Sutskever et al., 2014, Cho et al., 2014
- 2 RNN's, one for "reading" the input and one for "writing" the output (a.k.a the encoder-decoder architecture)



Encoder

Sequence 2 Sequence Learning

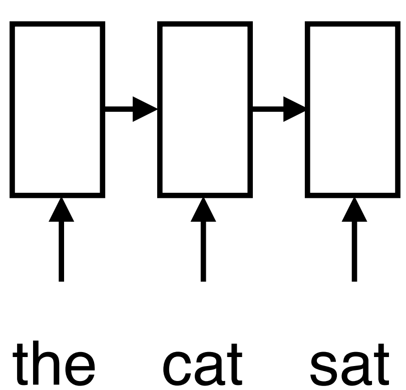
- Inspired by RNN language modeling
- First (modern) models for NMT presented by Kalchbrenner et. al. 2013, Sutskever et al., 2014, Cho et al., 2014
- 2 RNN’s, one for “reading” the input and one for “writing” the output (a.k.a the encoder-decoder architecture)



Encoder

Sequence 2 Sequence Learning

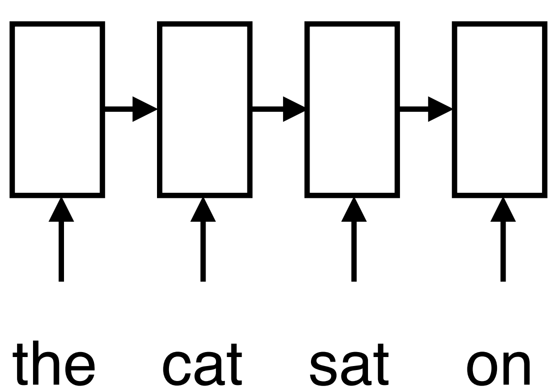
- Inspired by RNN language modeling
- First (modern) models for NMT presented by Kalchbrenner et. al. 2013, Sutskever et al., 2014, Cho et al., 2014
- 2 RNN's, one for "reading" the input and one for "writing" the output (a.k.a the encoder-decoder architecture)



Encoder

Sequence 2 Sequence Learning

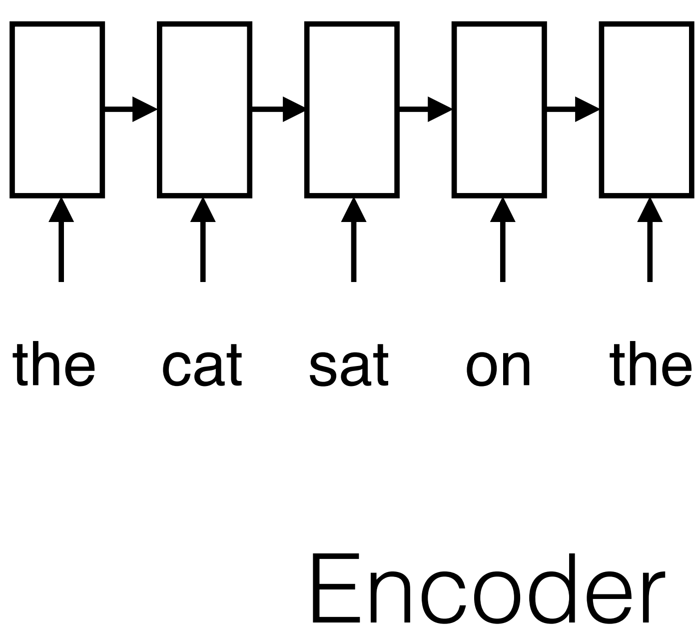
- Inspired by RNN language modeling
- First (modern) models for NMT presented by Kalchbrenner et. al. 2013, Sutskever et al., 2014, Cho et al., 2014
- 2 RNN's, one for "reading" the input and one for "writing" the output (a.k.a the encoder-decoder architecture)



Encoder

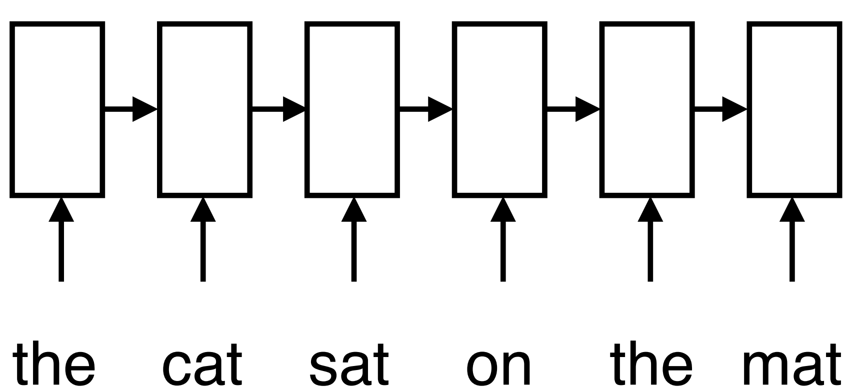
Sequence 2 Sequence Learning

- Inspired by RNN language modeling
- First (modern) models for NMT presented by Kalchbrenner et. al. 2013, Sutskever et al., 2014, Cho et al., 2014
- 2 RNN’s, one for “reading” the input and one for “writing” the output (a.k.a the encoder-decoder architecture)



Sequence 2 Sequence Learning

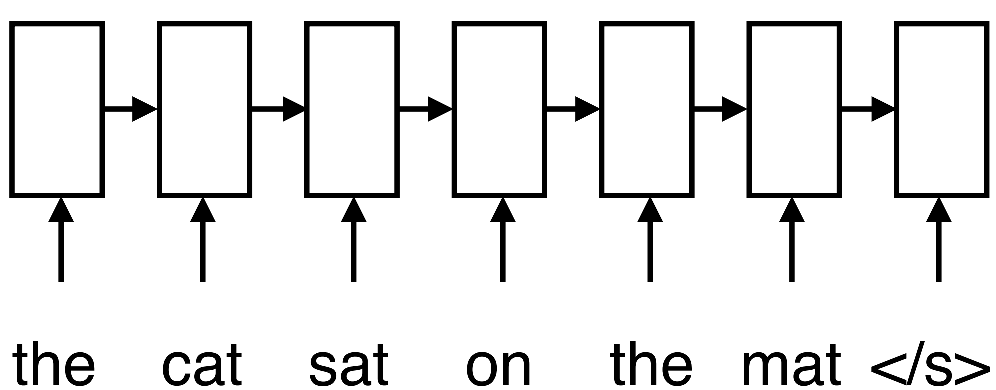
- Inspired by RNN language modeling
- First (modern) models for NMT presented by Kalchbrenner et. al. 2013, Sutskever et al., 2014, Cho et al., 2014
- 2 RNN's, one for "reading" the input and one for "writing" the output (a.k.a the encoder-decoder architecture)



Encoder

Sequence 2 Sequence Learning

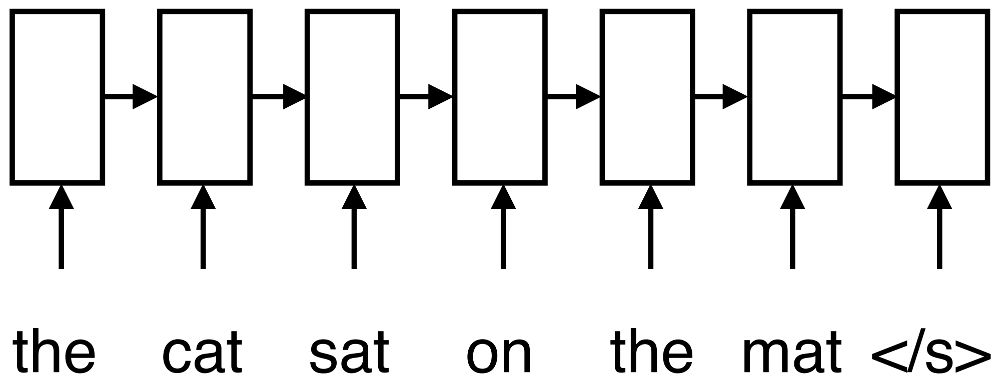
- Inspired by RNN language modeling
- First (modern) models for NMT presented by Kalchbrenner et. al. 2013, Sutskever et al., 2014, Cho et al., 2014
- 2 RNN’s, one for “reading” the input and one for “writing” the output (a.k.a the encoder-decoder architecture)



Encoder

Sequence 2 Sequence Learning

- Inspired by RNN language modeling
- First (modern) models for NMT presented by Kalchbrenner et. al. 2013, Sutskever et al., 2014, Cho et al., 2014
- 2 RNN's, one for "reading" the input and one for "writing" the output (a.k.a the encoder-decoder architecture)

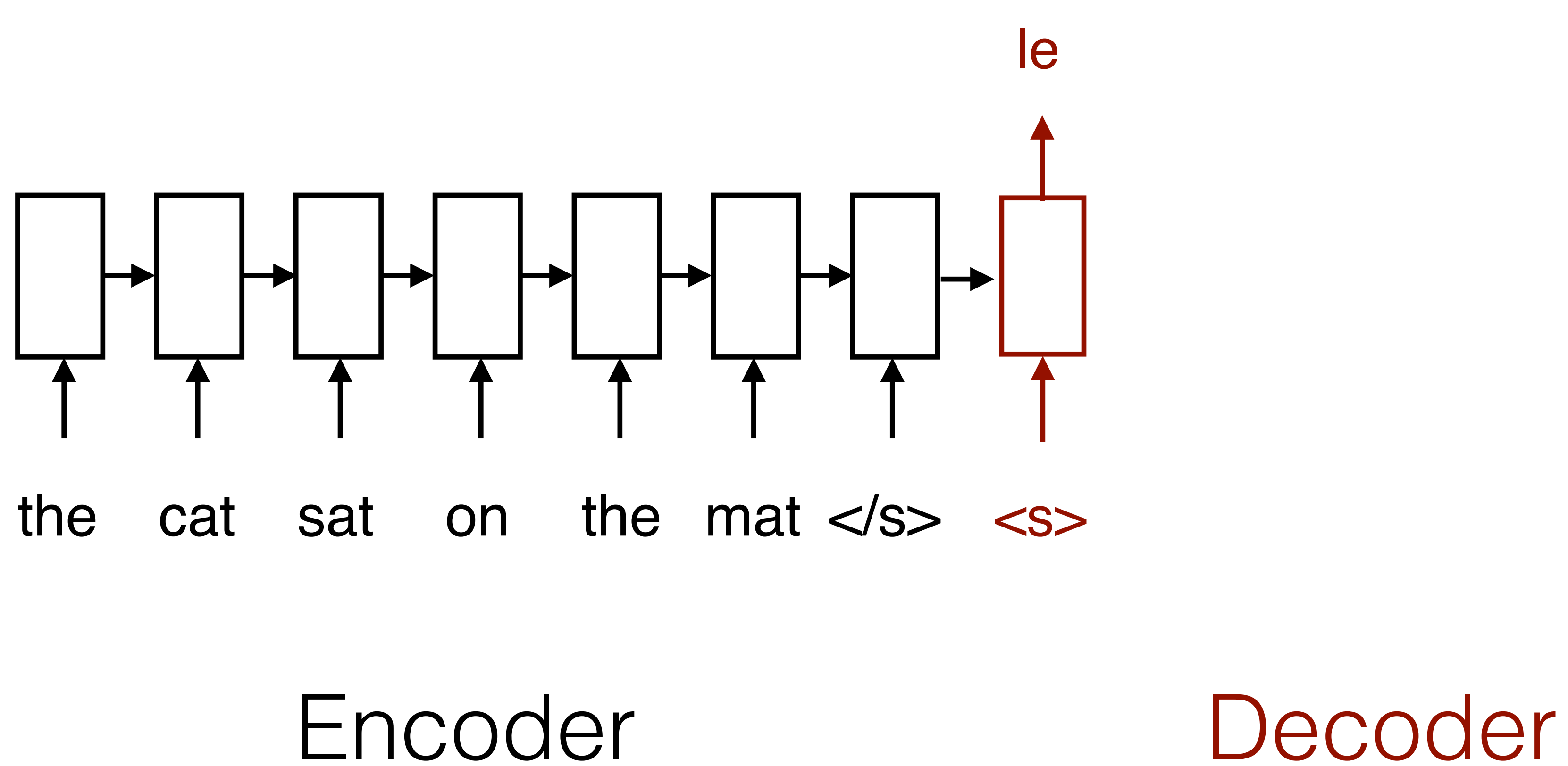


Encoder

Decoder

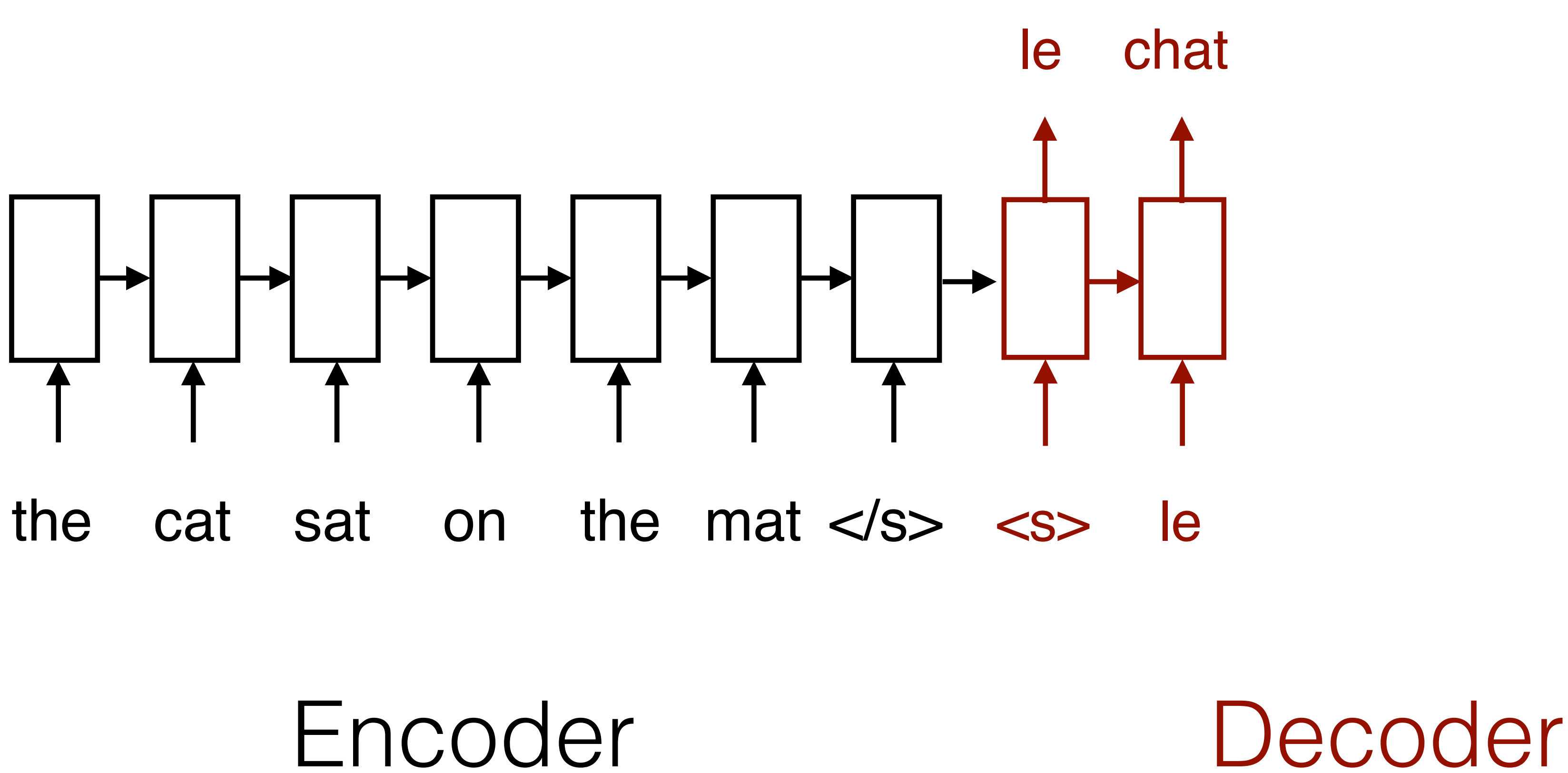
Sequence 2 Sequence Learning

- Inspired by RNN language modeling
- First (modern) models for NMT presented by Kalchbrenner et. al. 2013, Sutskever et al., 2014, Cho et al., 2014
- 2 RNN’s, one for “reading” the input and one for “writing” the output (a.k.a the encoder-decoder architecture)



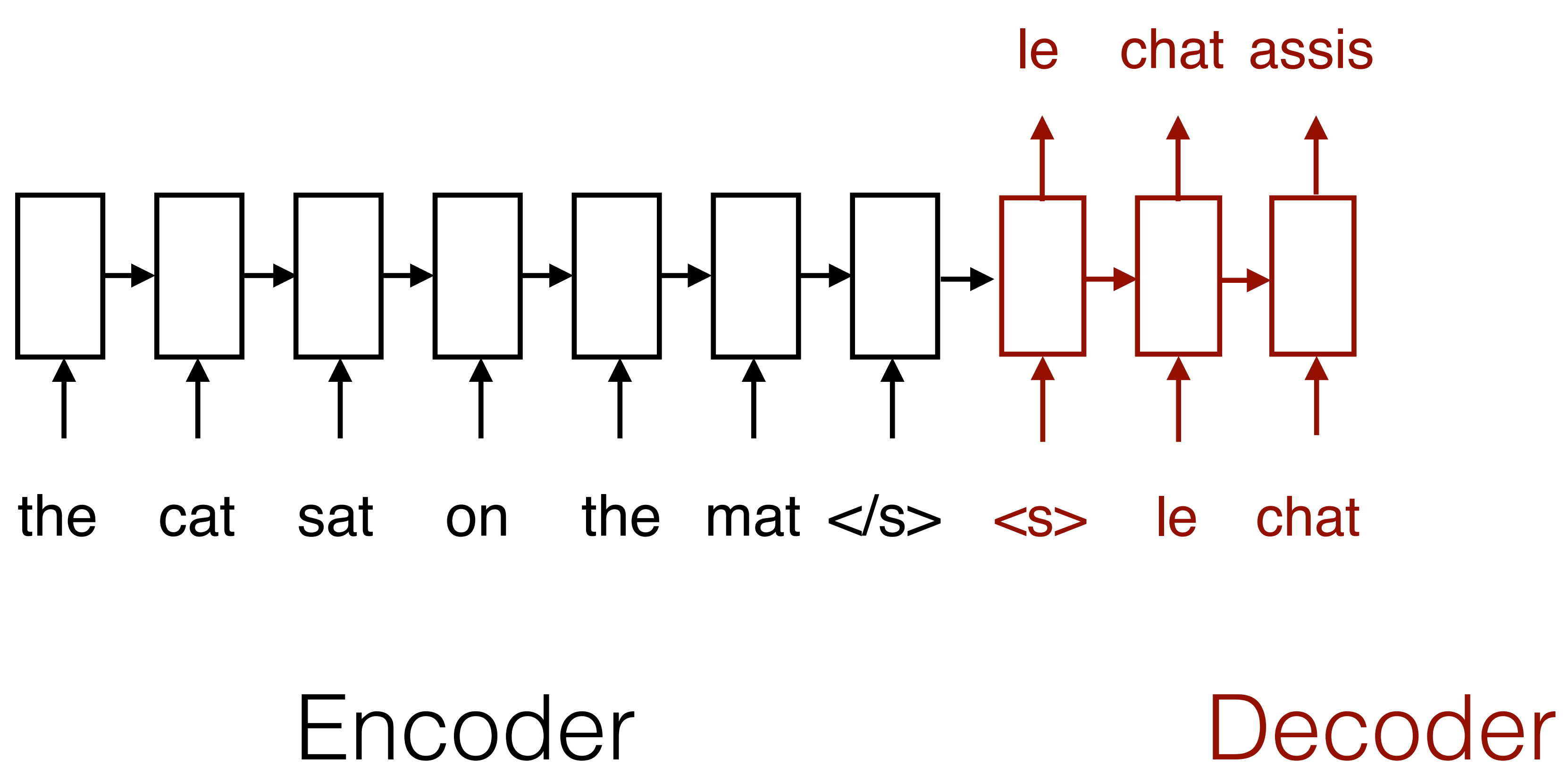
Sequence 2 Sequence Learning

- Inspired by RNN language modeling
- First (modern) models for NMT presented by Kalchbrenner et. al. 2013, Sutskever et al., 2014, Cho et al., 2014
- 2 RNN’s, one for “reading” the input and one for “writing” the output (a.k.a the encoder-decoder architecture)



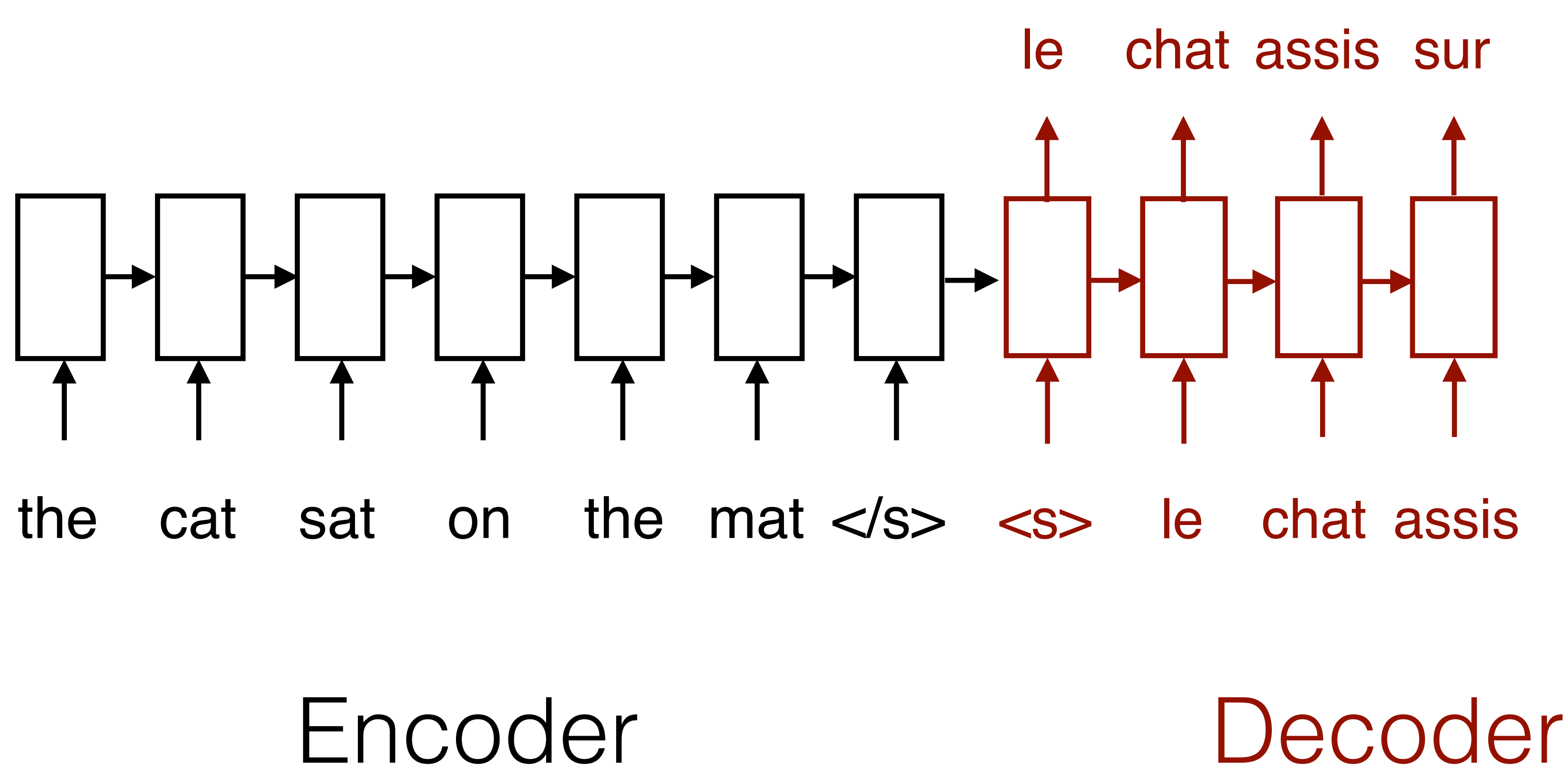
Sequence 2 Sequence Learning

- Inspired by RNN language modeling
- First (modern) models for NMT presented by Kalchbrenner et. al. 2013, Sutskever et al., 2014, Cho et al., 2014
- 2 RNN’s, one for “reading” the input and one for “writing” the output (a.k.a the encoder-decoder architecture)



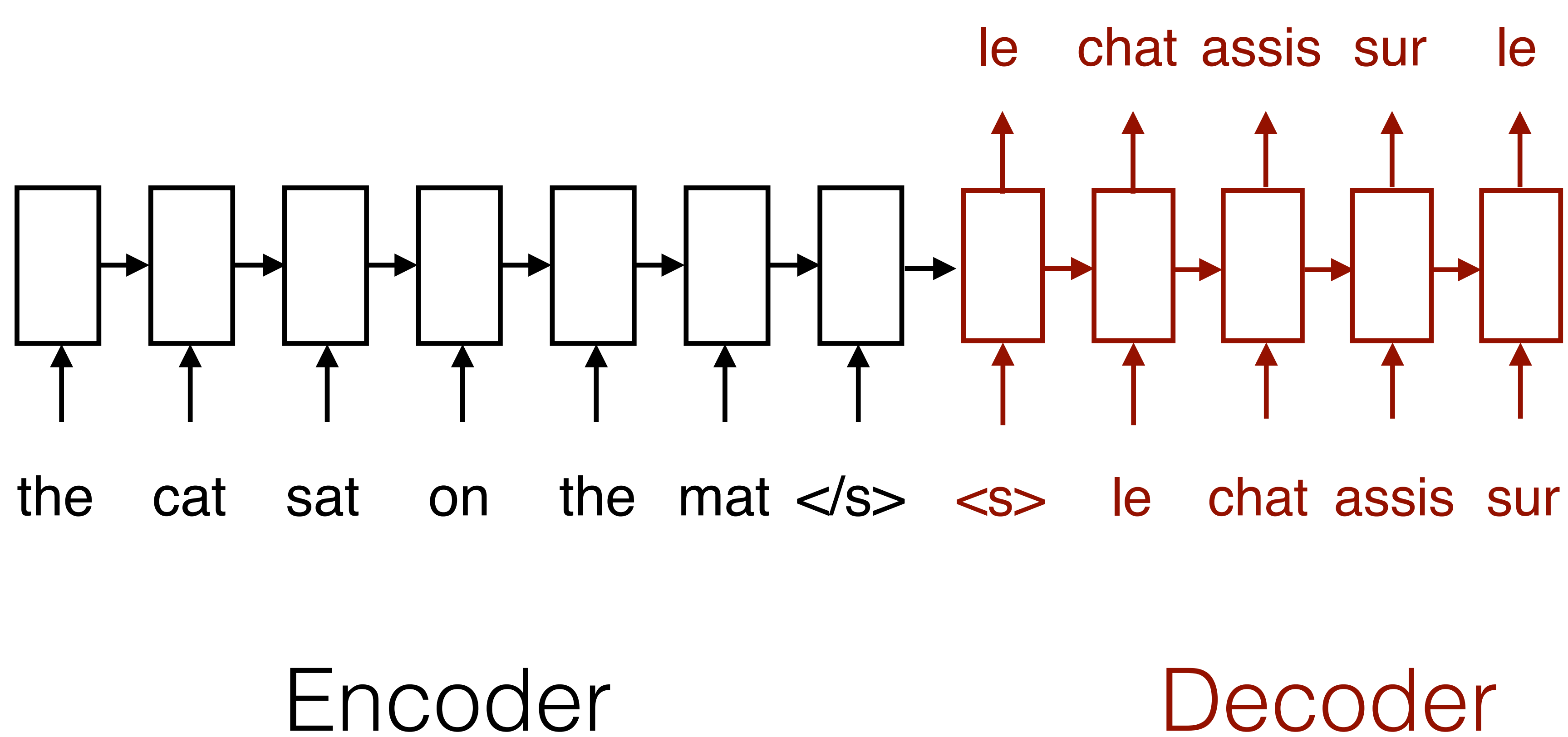
Sequence 2 Sequence Learning

- Inspired by RNN language modeling
- First (modern) models for NMT presented by Kalchbrenner et. al. 2013, Sutskever et al., 2014, Cho et al., 2014
- 2 RNN's, one for "reading" the input and one for "writing" the output (a.k.a the encoder-decoder architecture)



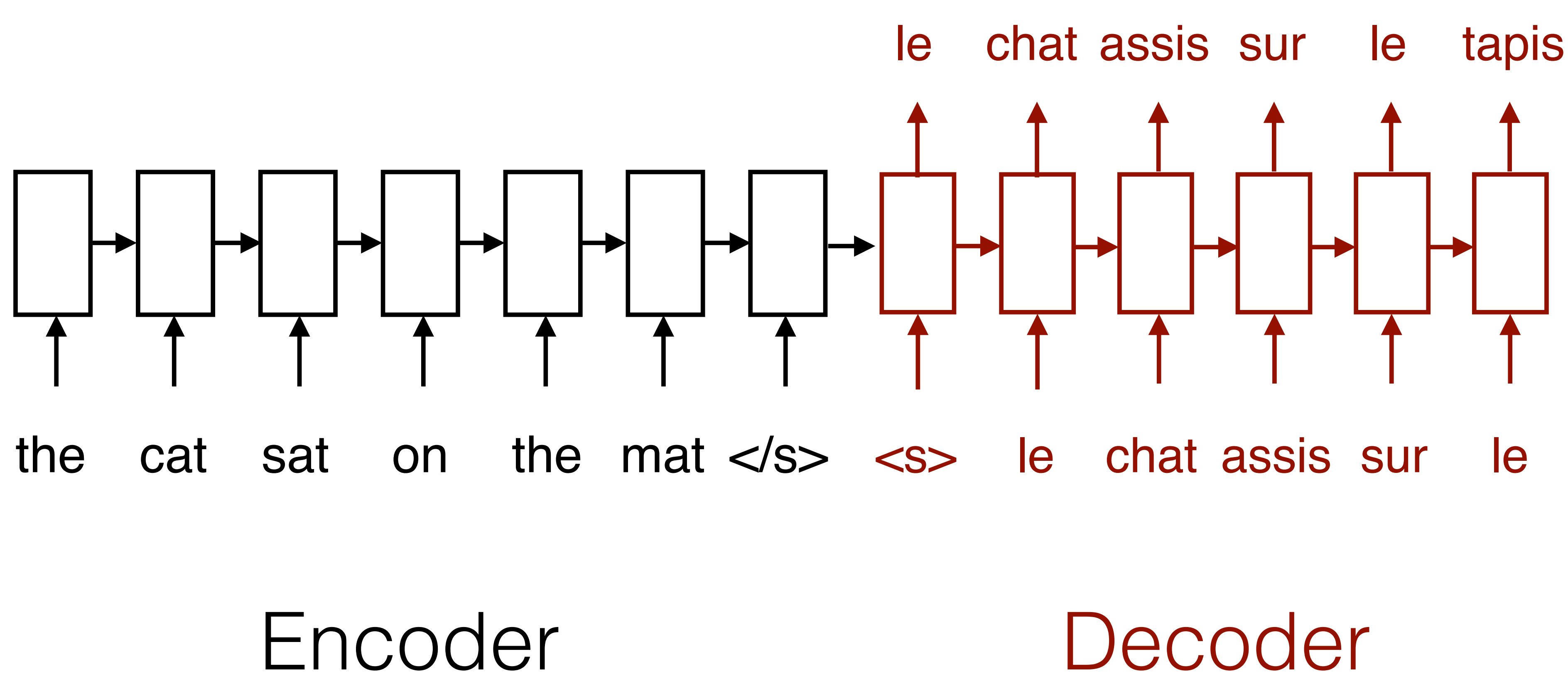
Sequence 2 Sequence Learning

- Inspired by RNN language modeling
- First (modern) models for NMT presented by Kalchbrenner et. al. 2013, Sutskever et al., 2014, Cho et al., 2014
- 2 RNN’s, one for “reading” the input and one for “writing” the output (a.k.a the encoder-decoder architecture)



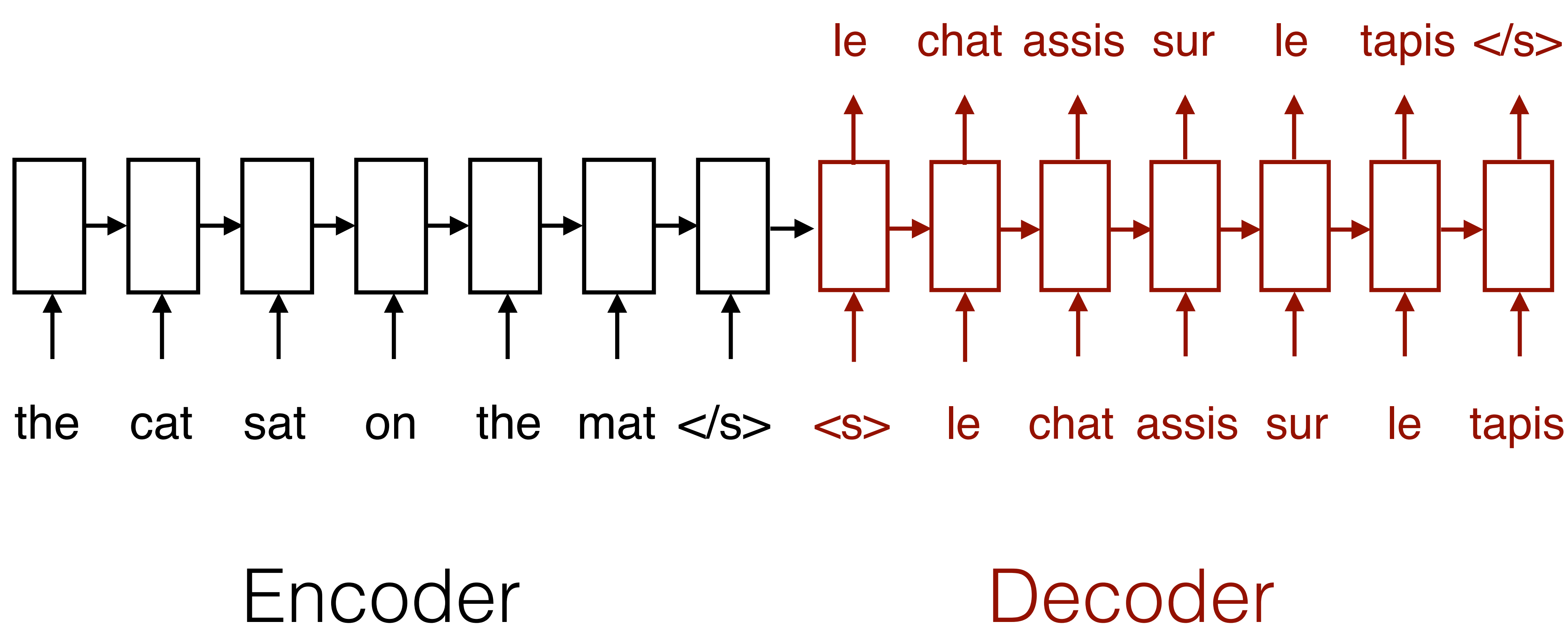
Sequence 2 Sequence Learning

- Inspired by RNN language modeling
- First (modern) models for NMT presented by Kalchbrenner et. al. 2013, Sutskever et al., 2014, Cho et al., 2014
- 2 RNN's, one for "reading" the input and one for "writing" the output (a.k.a the encoder-decoder architecture)



Sequence 2 Sequence Learning

- Inspired by RNN language modeling
- First (modern) models for NMT presented by Kalchbrenner et. al. 2013, Sutskever et al., 2014, Cho et al., 2014
- 2 RNN’s, one for “reading” the input and one for “writing” the output (a.k.a the encoder-decoder architecture)



Sequence 2 Sequence Learning

Sequence 2 Sequence Learning

More formally - model $p(y|x)$ using a single neural network:

Sequence 2 Sequence Learning

More formally - model $p(y|x)$ using a single neural network:

$$y = y_1 ... y_N$$

Sequence 2 Sequence Learning

More formally - model $p(y|x)$ using a single neural network:

$$y = y_1 ... y_N$$

$$p(y|x) = p(y_1|x)p(y_2|y_1, x)p(y_3|y_1, y_2, x)\ldots p(y_N|y_1 ... y_{N-1}, x)$$

Sequence 2 Sequence Learning

More formally - model $p(y|x)$ using a single neural network:

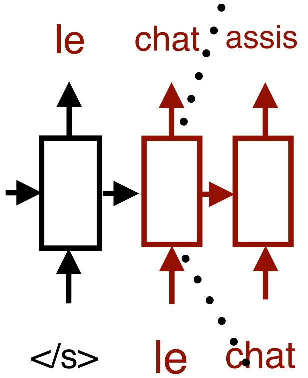
$$y = y_1 ... y_N$$

$$p(y|x) = p(y_1|x)p(y_2|y_1, x)p(y_3|y_1, y_2, x) \dots p(y_N|y_1...y_{N-1}, x)$$

$$p(y_i = word_k|y_{<i}, x) = softmax_k(NN_\Theta(y_{<i}, x))$$

Seq2Seq decoder step - Zoom-In

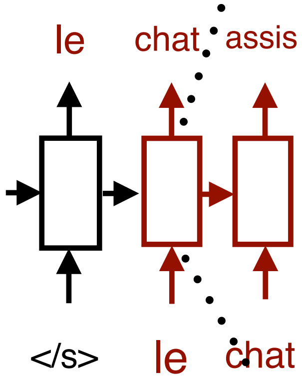
“chat”



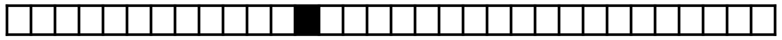
“le”

Seq2Seq decoder step - Zoom-In

“chat”



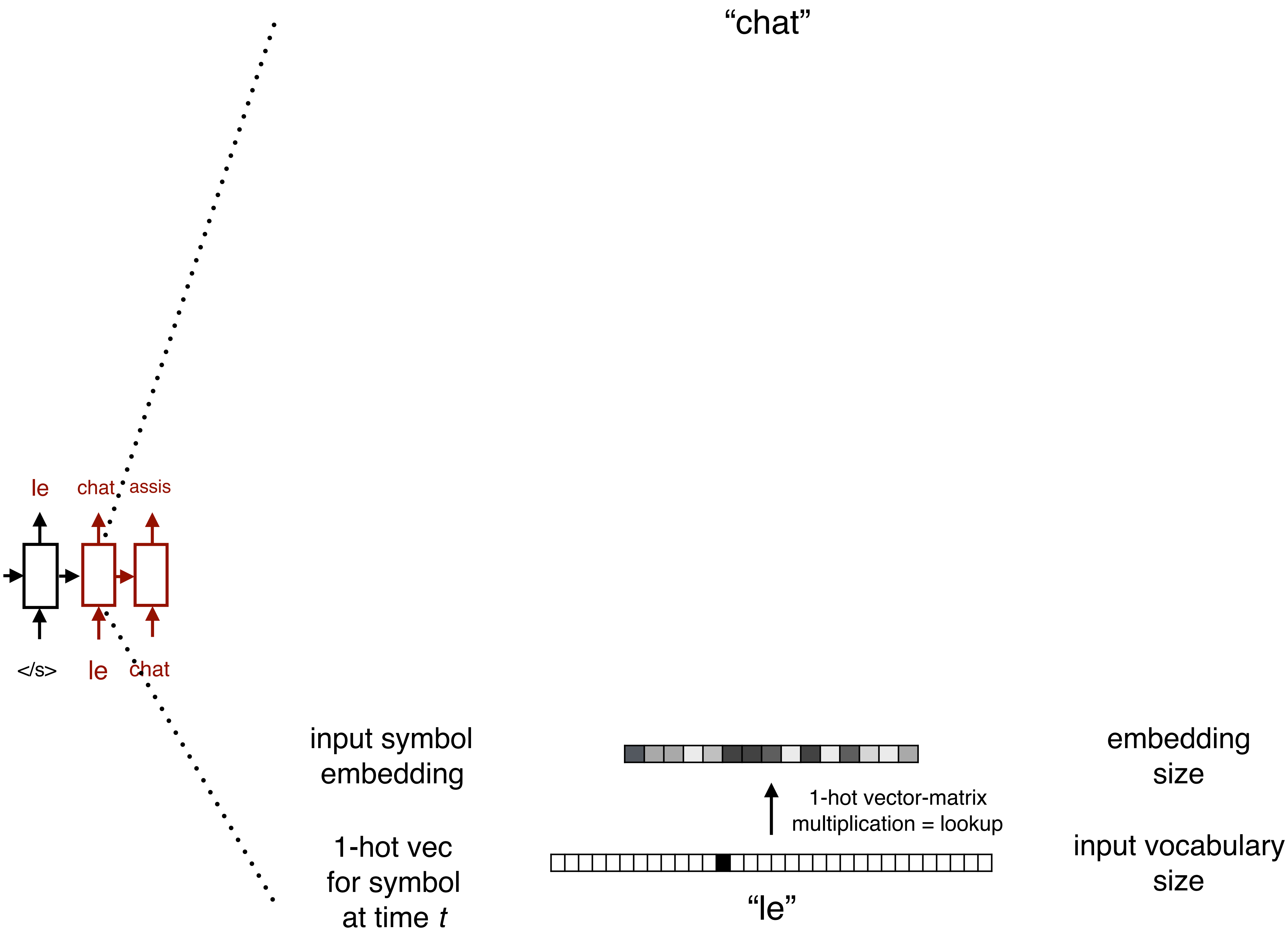
1-hot vec
for symbol
at time *t*



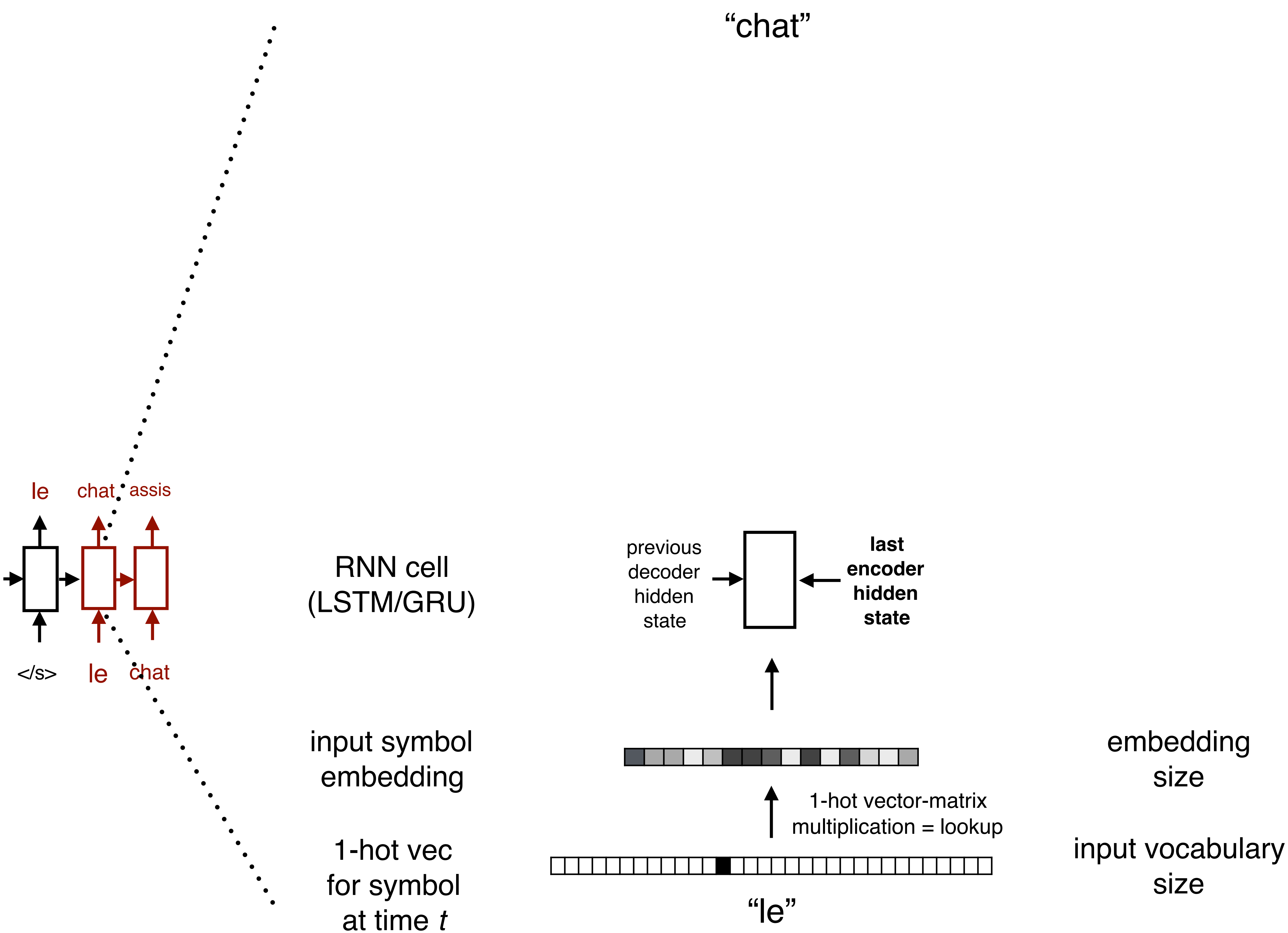
“le”

input vocabulary
size

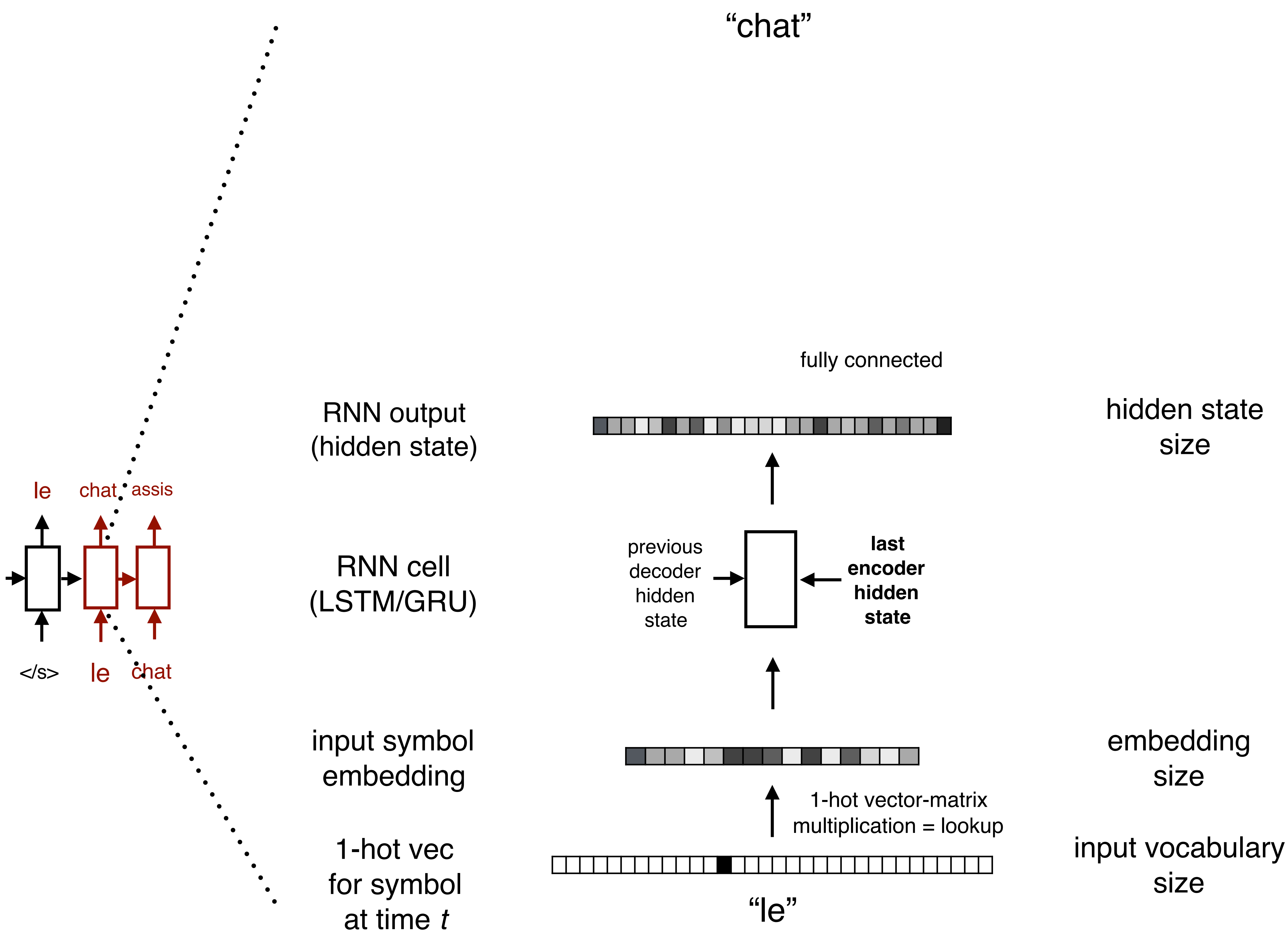
Seq2Seq decoder step - Zoom-In



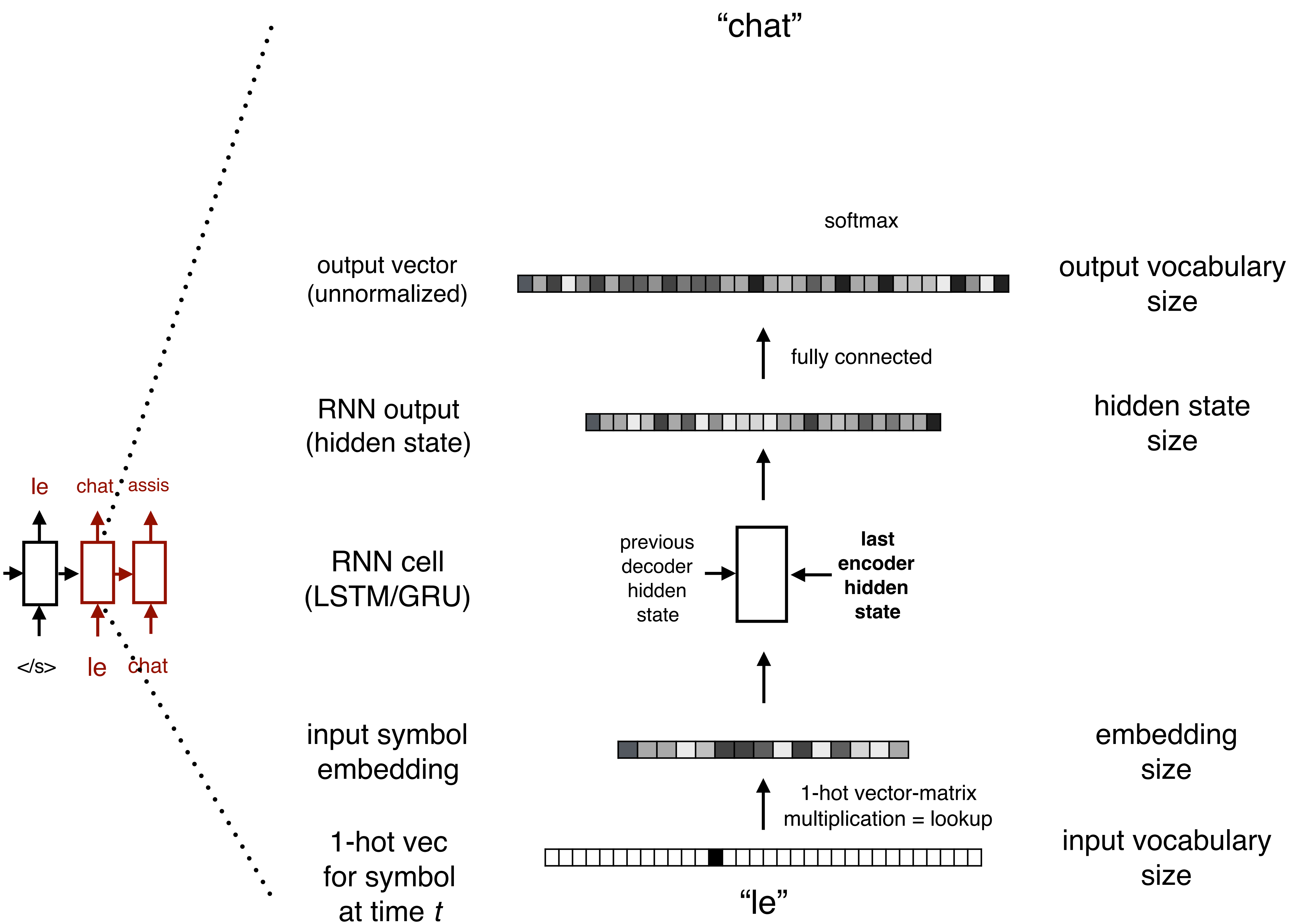
Seq2Seq decoder step - Zoom-In



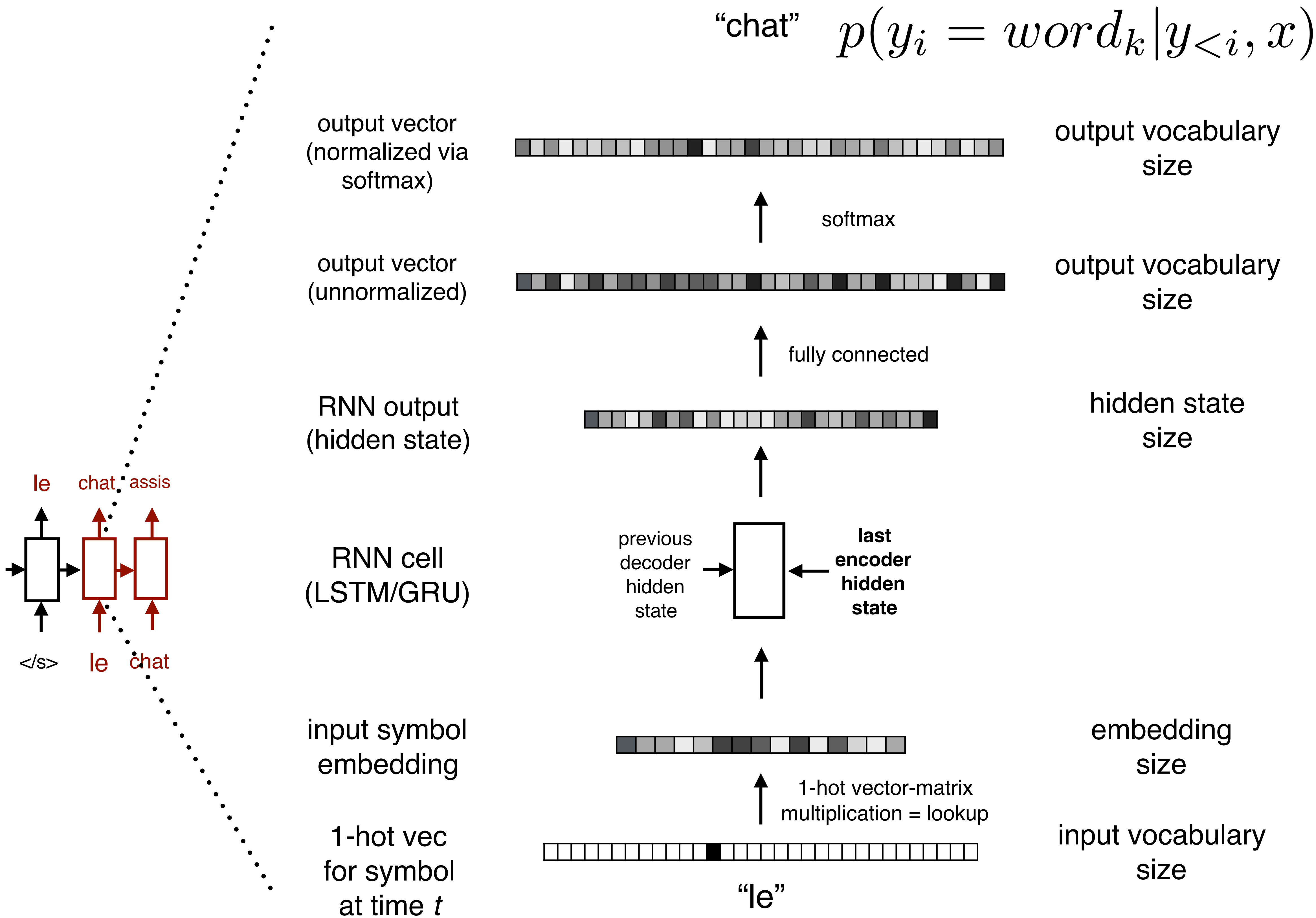
Seq2Seq decoder step - Zoom-In



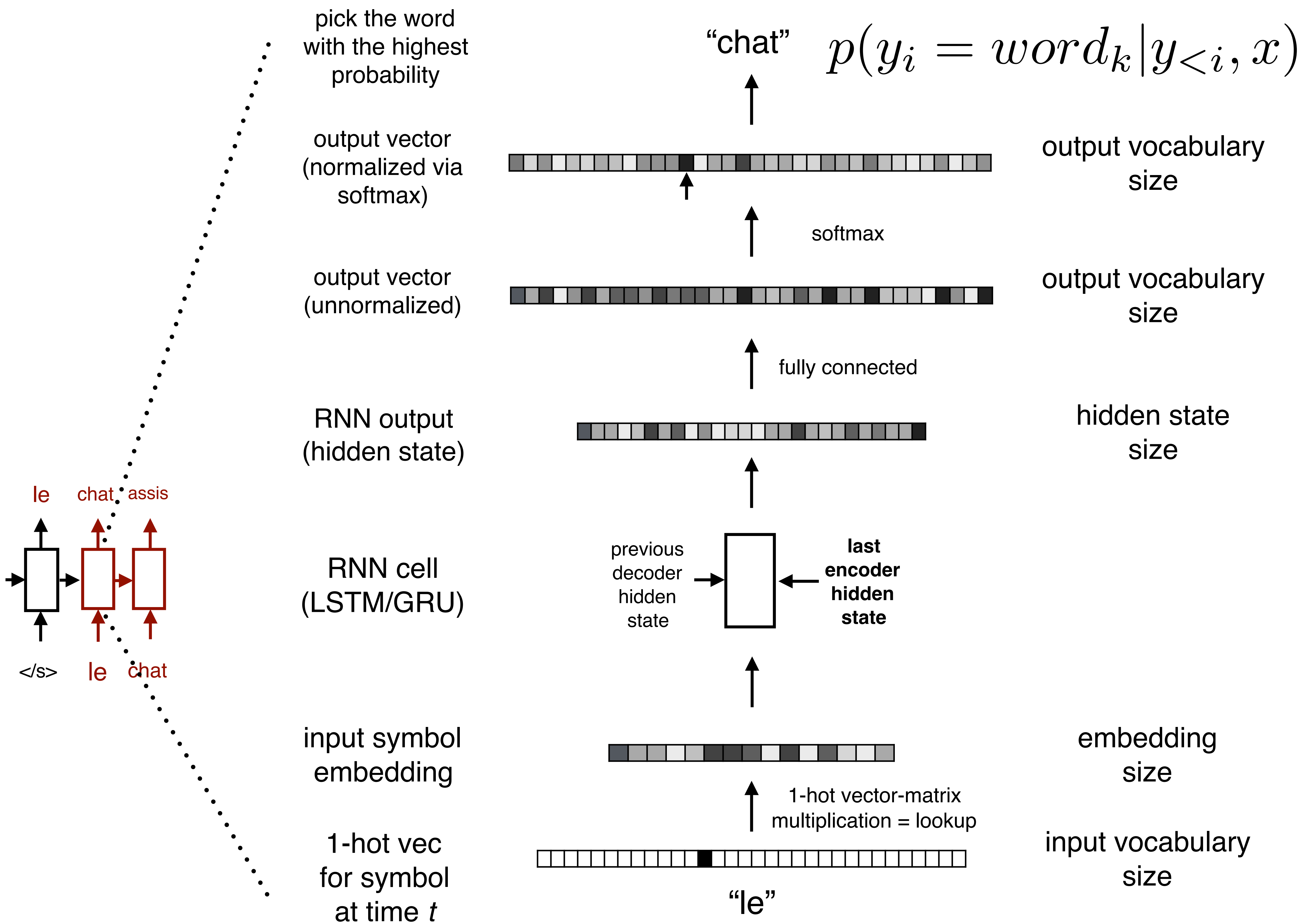
Seq2Seq decoder step - Zoom-In



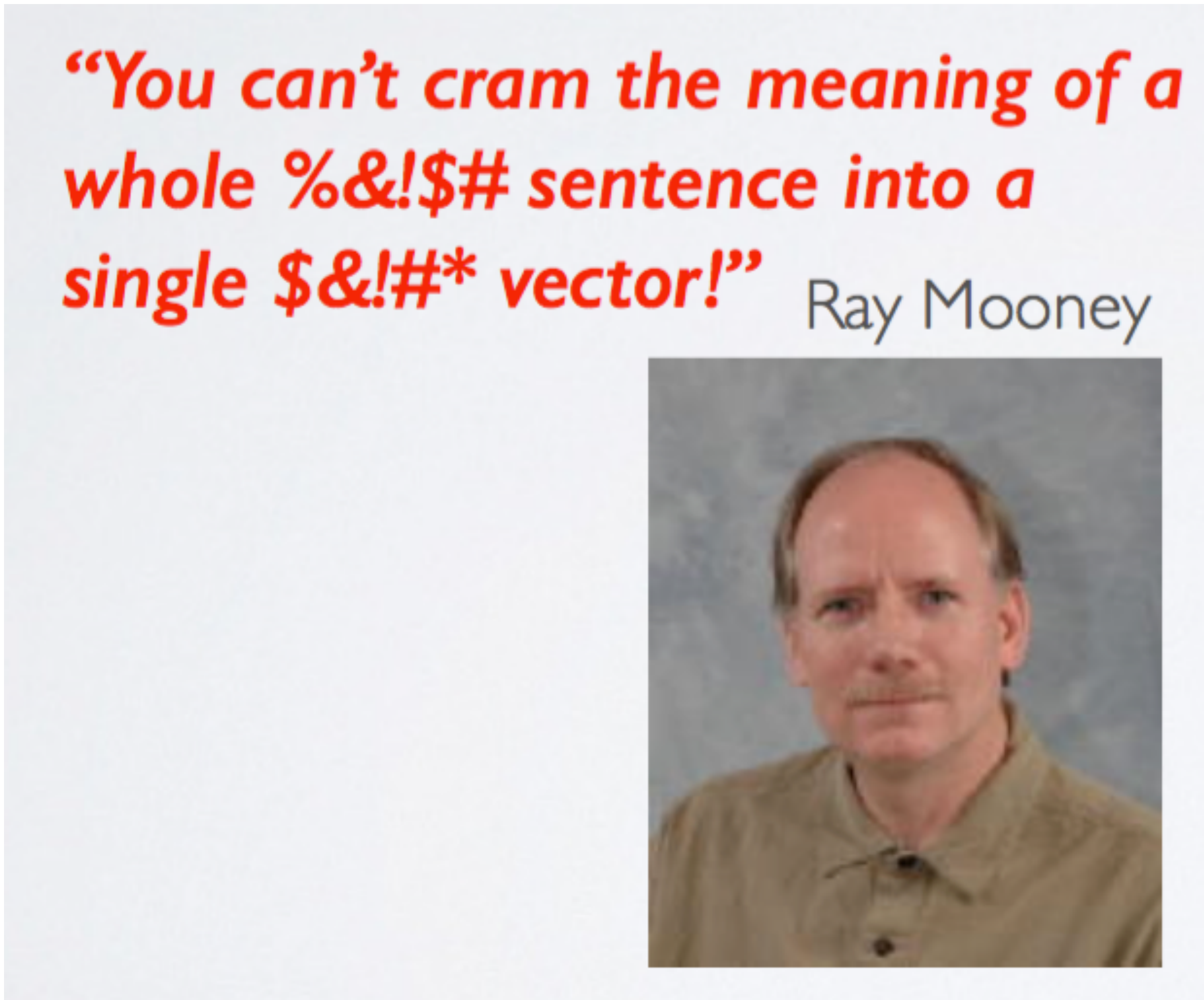
Seq2Seq decoder step - Zoom-In



Seq2Seq decoder step - Zoom-In



The problem with “vanilla” seq2seq



The Attention Mechanism

The Attention Mechanism

- Instead of using a single vector as a fixed representation of the input sequence, “**attend**” at each step to the **relevant parts** of the input

The Attention Mechanism

- Instead of using a single vector as a fixed representation of the input sequence, "**attend**" at each step to the **relevant parts** of the input

- The "**relevance**" of each input element to the current prediction is **computed** via a **feed-forward network** that gets the input element and the current decoder state

The Attention Mechanism

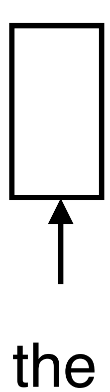
- Instead of using a single vector as a fixed representation of the input sequence, “**attend**” at each step to the **relevant parts** of the input

- The “**relevance**” of each input element to the current prediction is **computed** via a **feed-forward network** that gets the input element and the current decoder state

- Coined as “**Resolution Preserving**” - longer sequences get longer representations

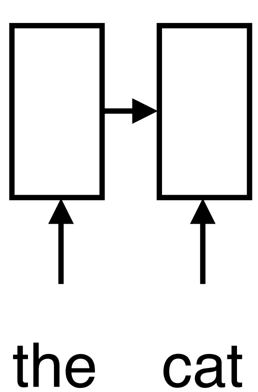
The Attention Mechanism

- Instead of using a single vector as a fixed representation of the input sequence, “**attend**” at each step to the **relevant parts** of the input
- The “**relevance**” of each input element to the current prediction is **computed** via a **feed-forward network** that gets the input element and the current decoder state
- Coined as “**Resolution Preserving**” - longer sequences get longer representations



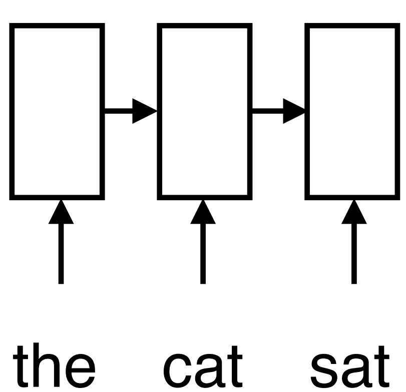
The Attention Mechanism

- Instead of using a single vector as a fixed representation of the input sequence, "**attend**" at each step to the **relevant parts** of the input
- The "**relevance**" of each input element to the current prediction is **computed** via a **feed-forward network** that gets the input element and the current decoder state
- Coined as "**Resolution Preserving**" - longer sequences get longer representations



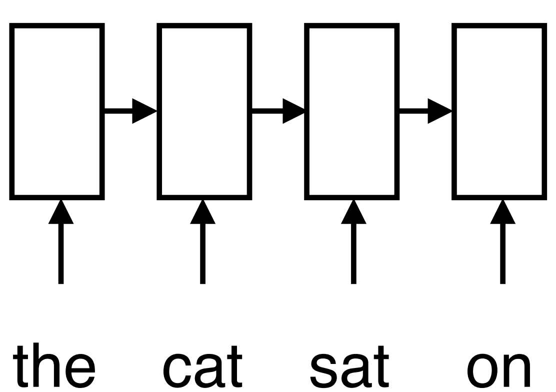
The Attention Mechanism

- Instead of using a single vector as a fixed representation of the input sequence, "**attend**" at each step to the **relevant parts** of the input
- The "**relevance**" of each input element to the current prediction is **computed** via a **feed-forward network** that gets the input element and the current decoder state
- Coined as "**Resolution Preserving**" - longer sequences get longer representations



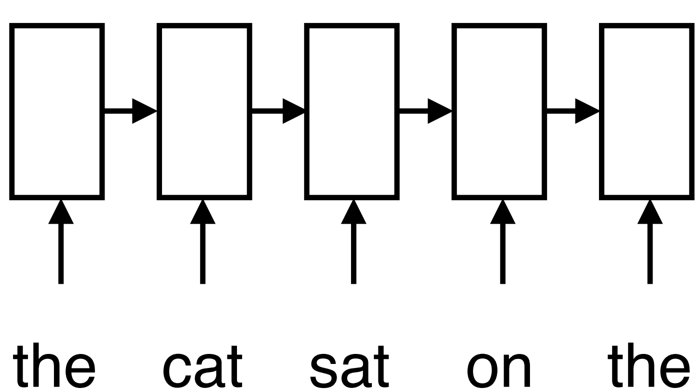
The Attention Mechanism

- Instead of using a single vector as a fixed representation of the input sequence, "**attend**" at each step to the **relevant parts** of the input
- The "**relevance**" of each input element to the current prediction is **computed** via a **feed-forward network** that gets the input element and the current decoder state
- Coined as "**Resolution Preserving**" - longer sequences get longer representations



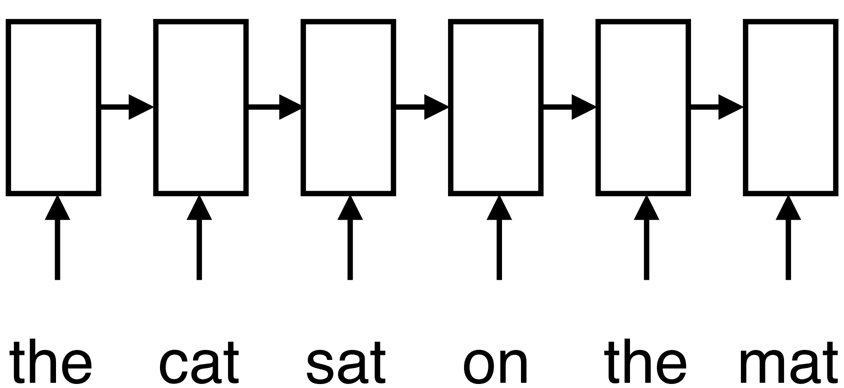
The Attention Mechanism

- Instead of using a single vector as a fixed representation of the input sequence, "**attend**" at each step to the **relevant parts** of the input
- The "**relevance**" of each input element to the current prediction is **computed** via a **feed-forward network** that gets the input element and the current decoder state
- Coined as "**Resolution Preserving**" - longer sequences get longer representations



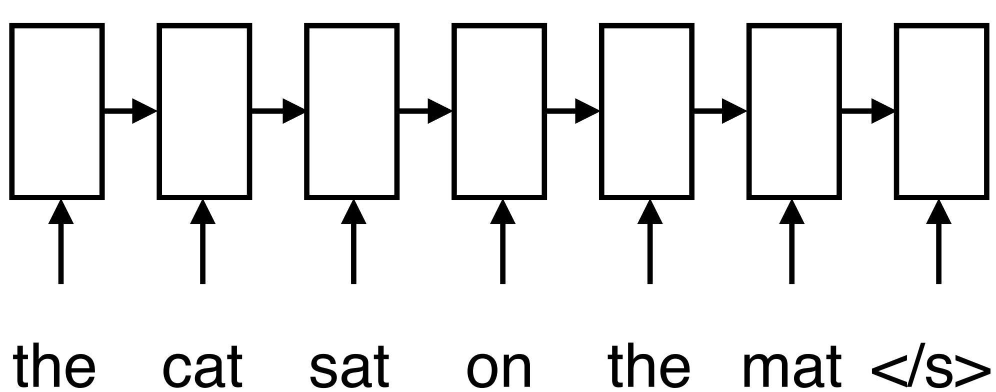
The Attention Mechanism

- Instead of using a single vector as a fixed representation of the input sequence, "**attend**" at each step to the **relevant parts** of the input
- The "**relevance**" of each input element to the current prediction is **computed** via a **feed-forward network** that gets the input element and the current decoder state
- Coined as "**Resolution Preserving**" - longer sequences get longer representations



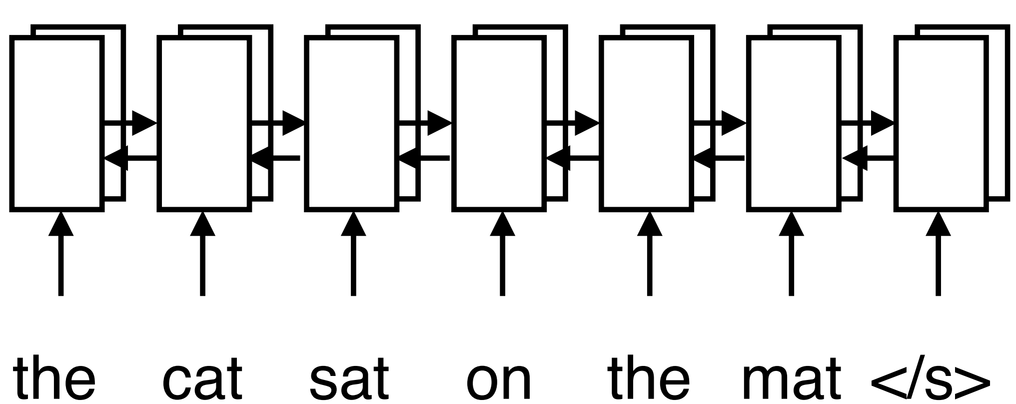
The Attention Mechanism

- Instead of using a single vector as a fixed representation of the input sequence, “**attend**” at each step to the **relevant parts** of the input
- The “**relevance**” of each input element to the current prediction is **computed** via a **feed-forward network** that gets the input element and the current decoder state
- Coined as “**Resolution Preserving**” - longer sequences get longer representations



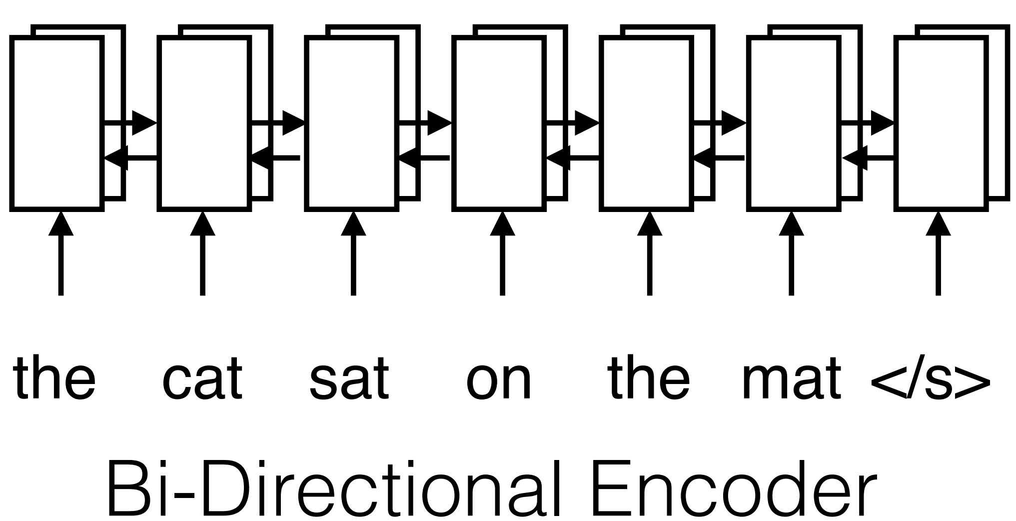
The Attention Mechanism

- Instead of using a single vector as a fixed representation of the input sequence, "**attend**" at each step to the **relevant parts** of the input
- The "**relevance**" of each input element to the current prediction is **computed** via a **feed-forward network** that gets the input element and the current decoder state
- Coined as "**Resolution Preserving**" - longer sequences get longer representations



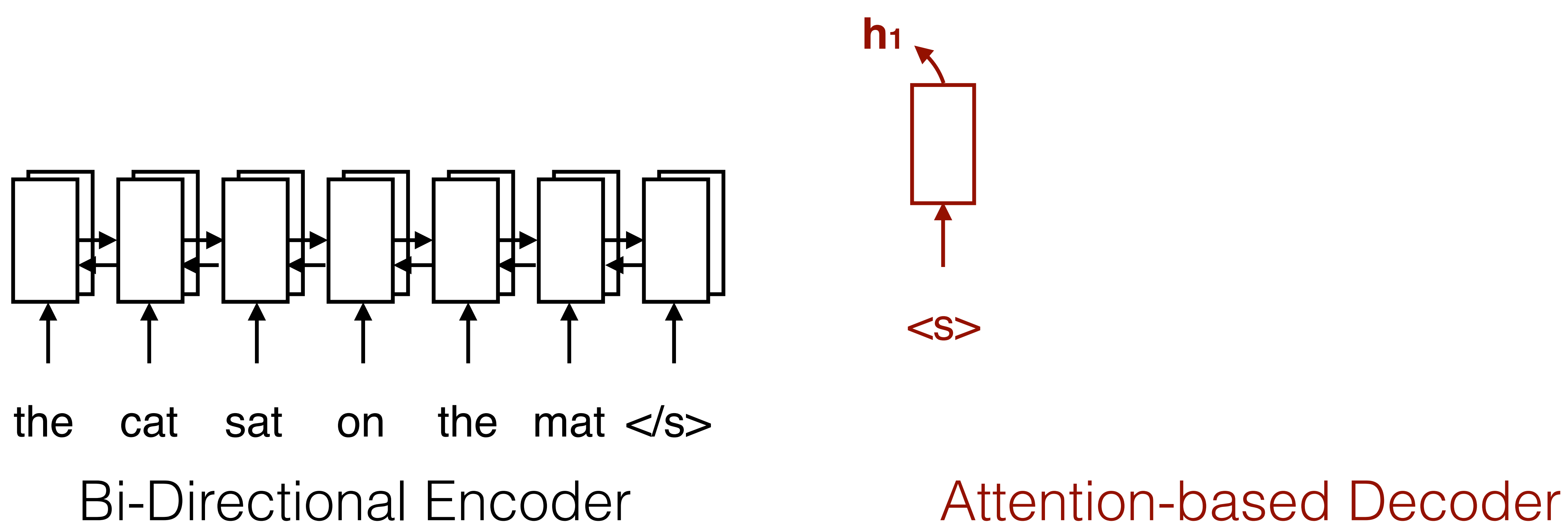
The Attention Mechanism

- Instead of using a single vector as a fixed representation of the input sequence, “**attend**” at each step to the **relevant parts** of the input
- The “**relevance**” of each input element to the current prediction is **computed** via a **feed-forward network** that gets the input element and the current decoder state
- Coined as “**Resolution Preserving**” - longer sequences get longer representations



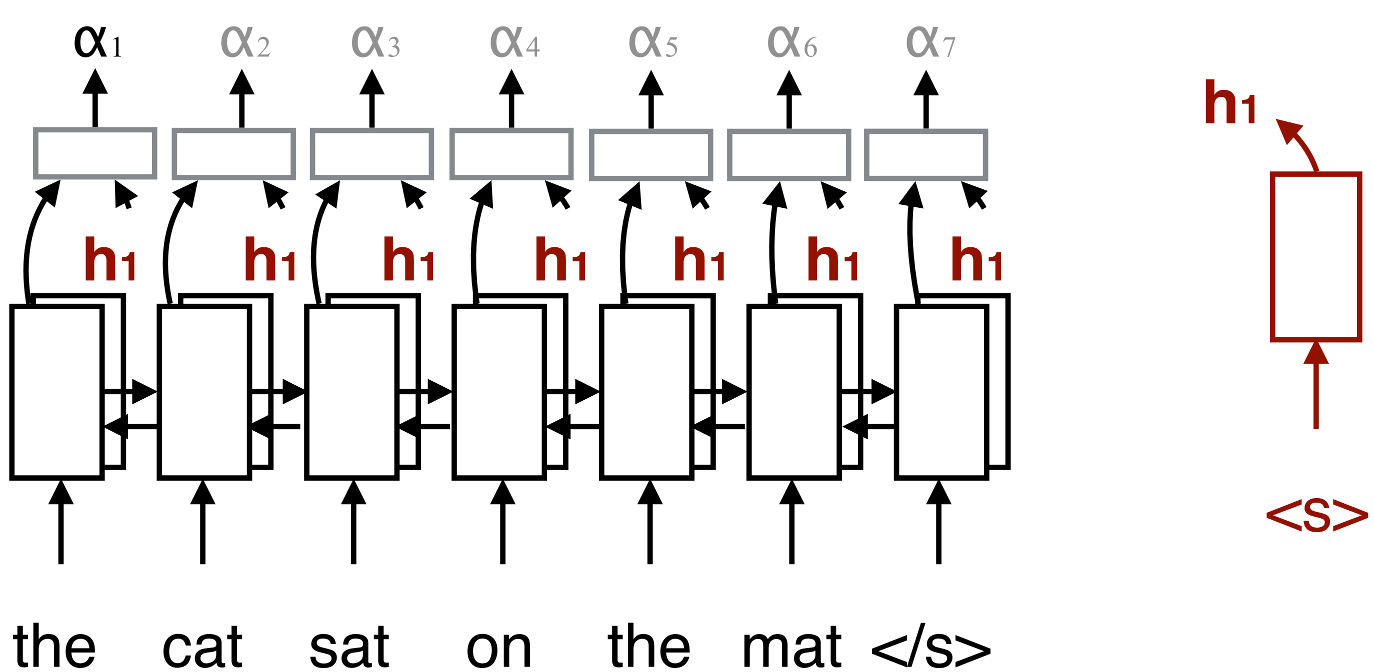
The Attention Mechanism

- Instead of using a single vector as a fixed representation of the input sequence, “**attend**” at each step to the **relevant parts** of the input
- The “**relevance**” of each input element to the current prediction is **computed** via a **feed-forward network** that gets the input element and the current decoder state
- Coined as “**Resolution Preserving**” - longer sequences get longer representations



The Attention Mechanism

- Instead of using a single vector as a fixed representation of the input sequence, “**attend**” at each step to the **relevant parts** of the input
- The “**relevance**” of each input element to the current prediction is **computed** via a **feed-forward network** that gets the input element and the current decoder state
- Coined as “**Resolution Preserving**” - longer sequences get longer representations

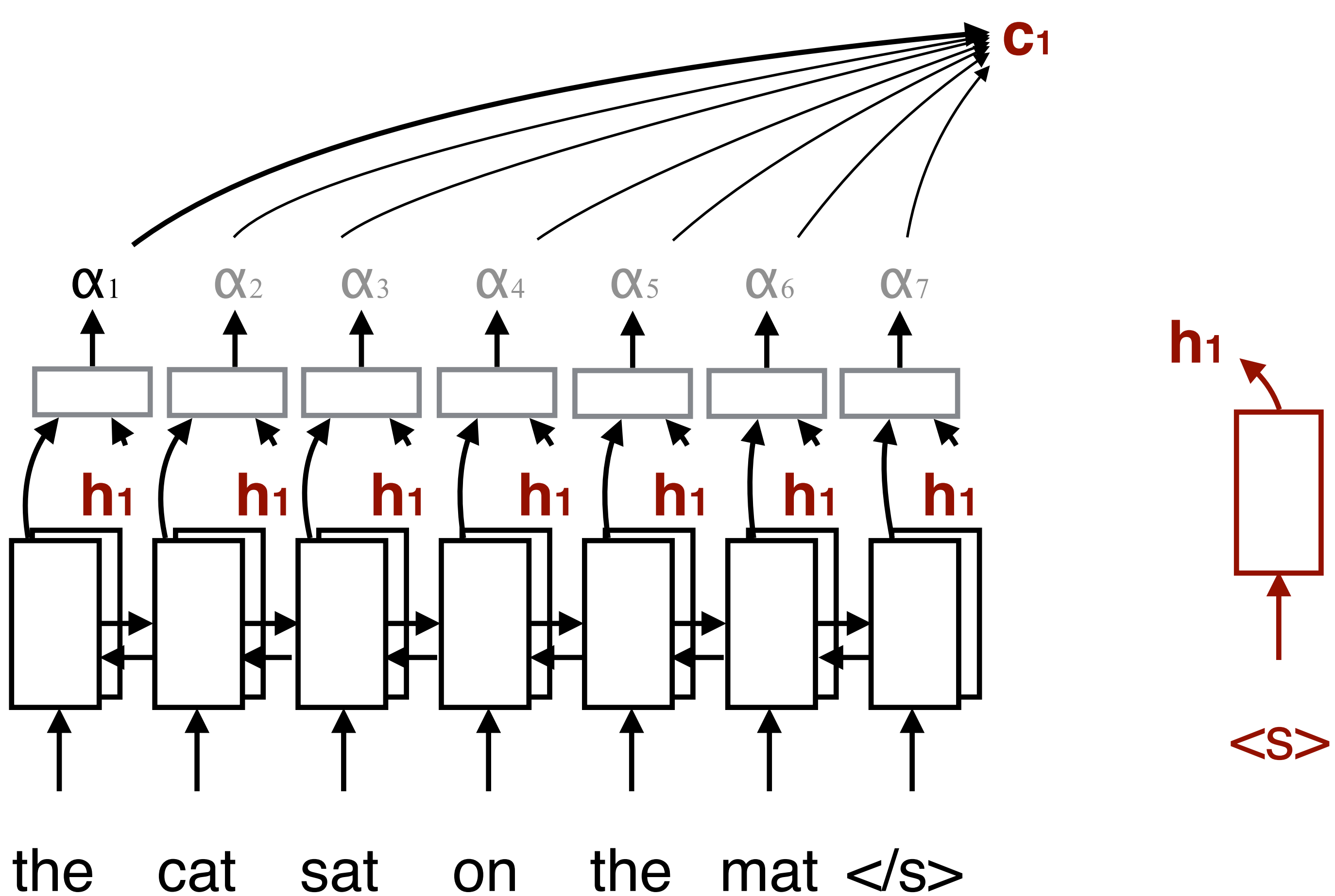


Bi-Directional Encoder

Attention-based Decoder

The Attention Mechanism

- Instead of using a single vector as a fixed representation of the input sequence, “**attend**” at each step to the **relevant parts** of the input
- The “**relevance**” of each input element to the current prediction is **computed** via a **feed-forward network** that gets the input element and the current decoder state
- Coined as “**Resolution Preserving**” - longer sequences get longer representations

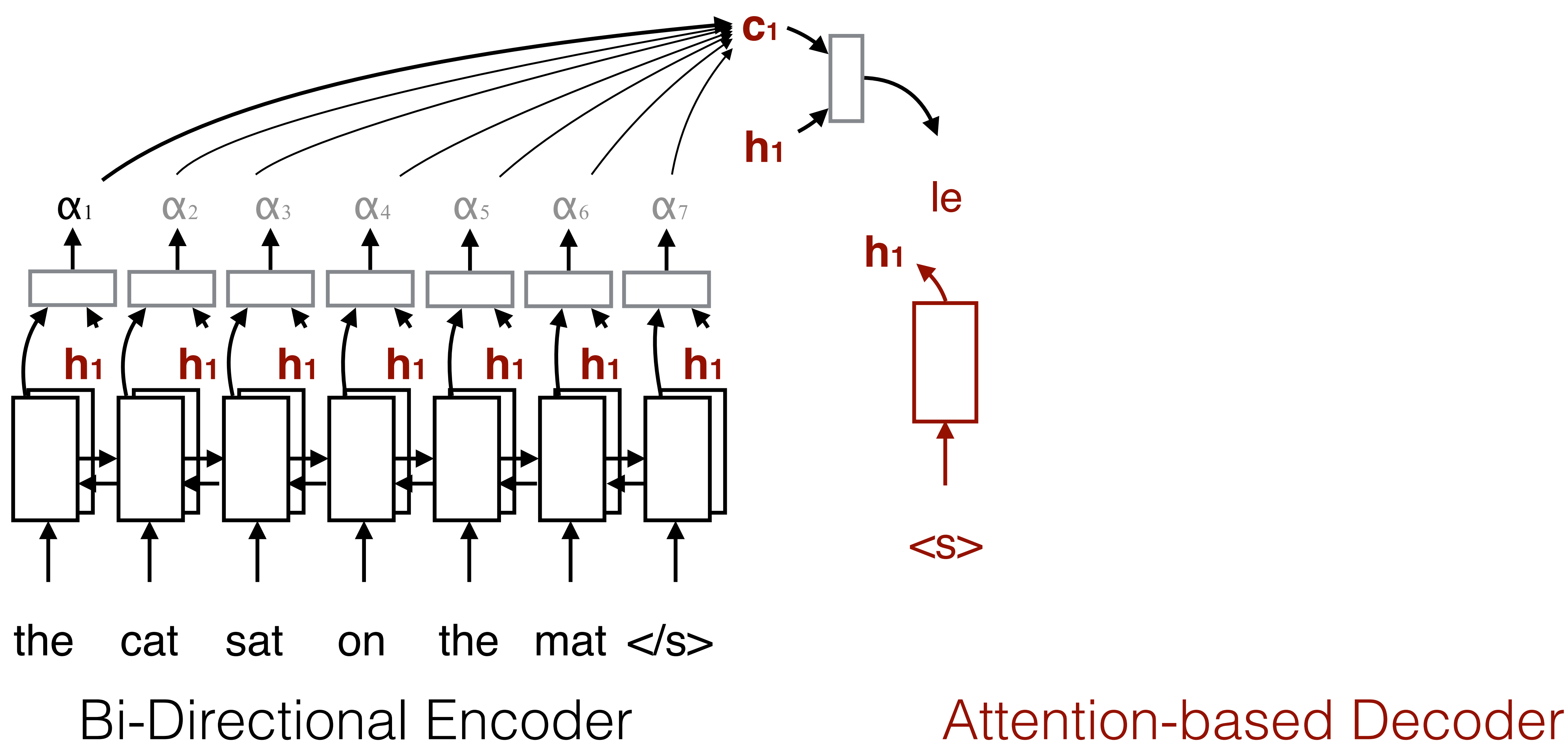


Bi-Directional Encoder

Attention-based Decoder

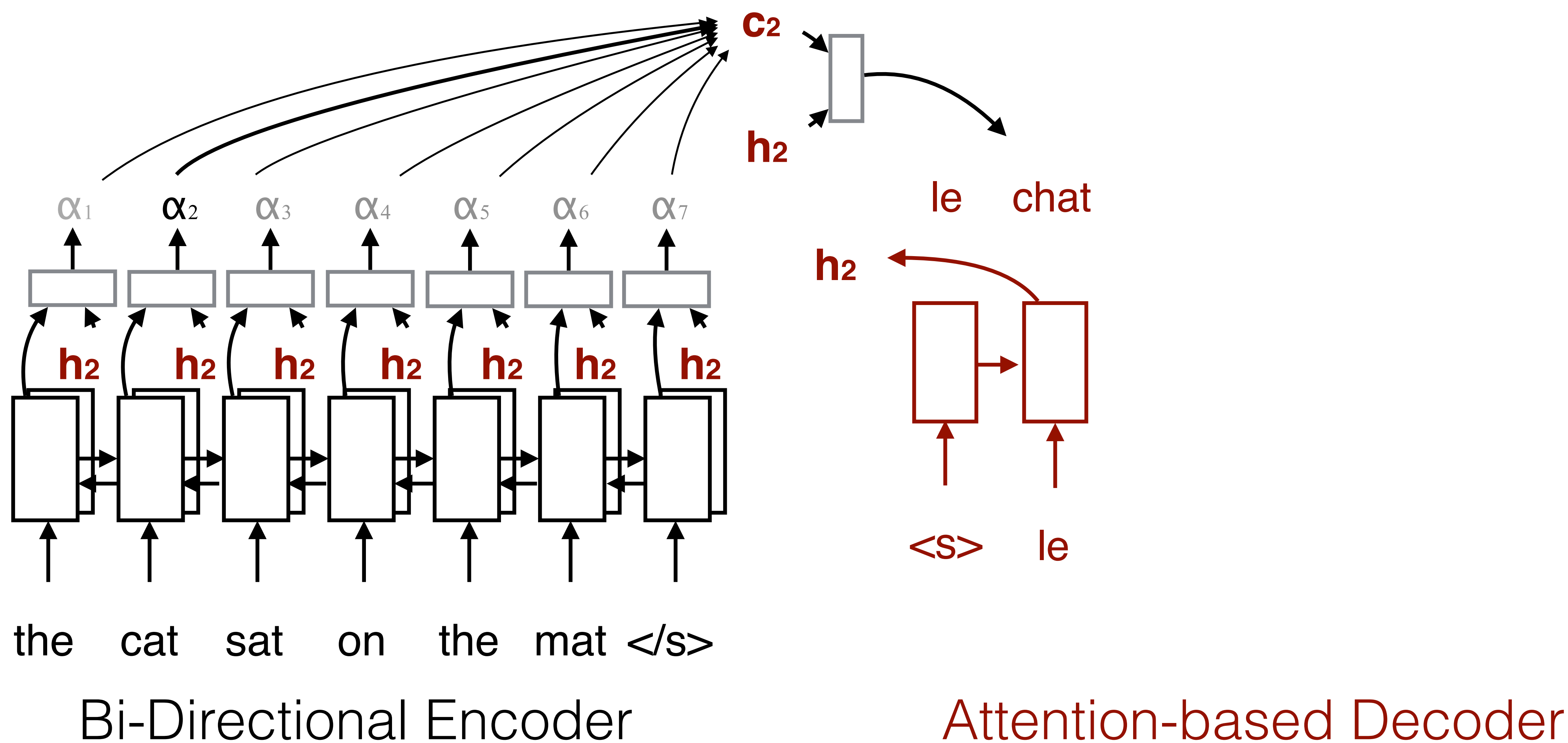
The Attention Mechanism

- Instead of using a single vector as a fixed representation of the input sequence, "**attend**" at each step to the **relevant parts** of the input
- The "**relevance**" of each input element to the current prediction is **computed** via a **feed-forward network** that gets the input element and the current decoder state
- Coined as "**Resolution Preserving**" - longer sequences get longer representations



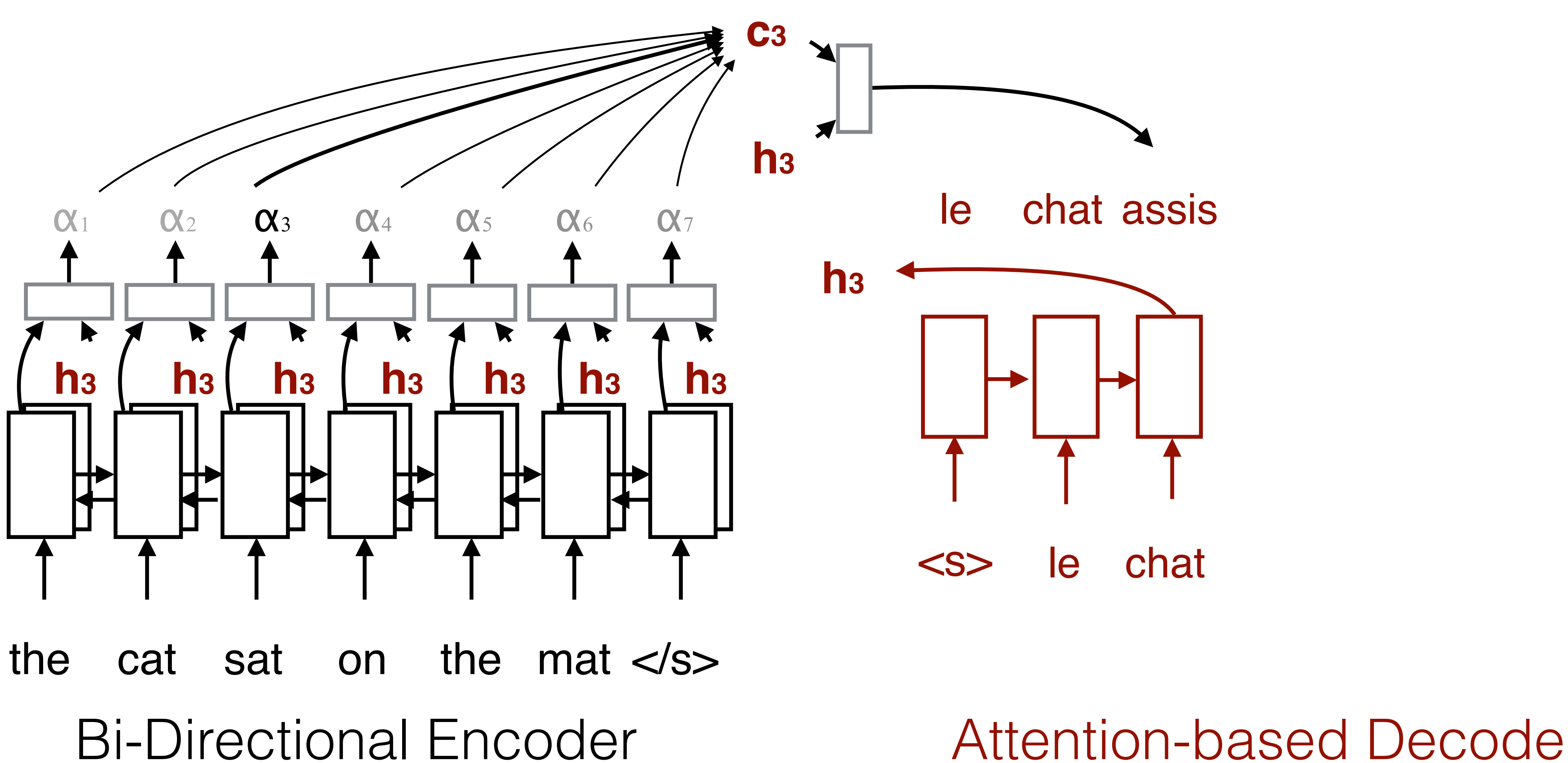
The Attention Mechanism

- Instead of using a single vector as a fixed representation of the input sequence, "**attend**" at each step to the **relevant parts** of the input
- The "**relevance**" of each input element to the current prediction is **computed** via a **feed-forward network** that gets the input element and the current decoder state
- Coined as "**Resolution Preserving**" - longer sequences get longer representations



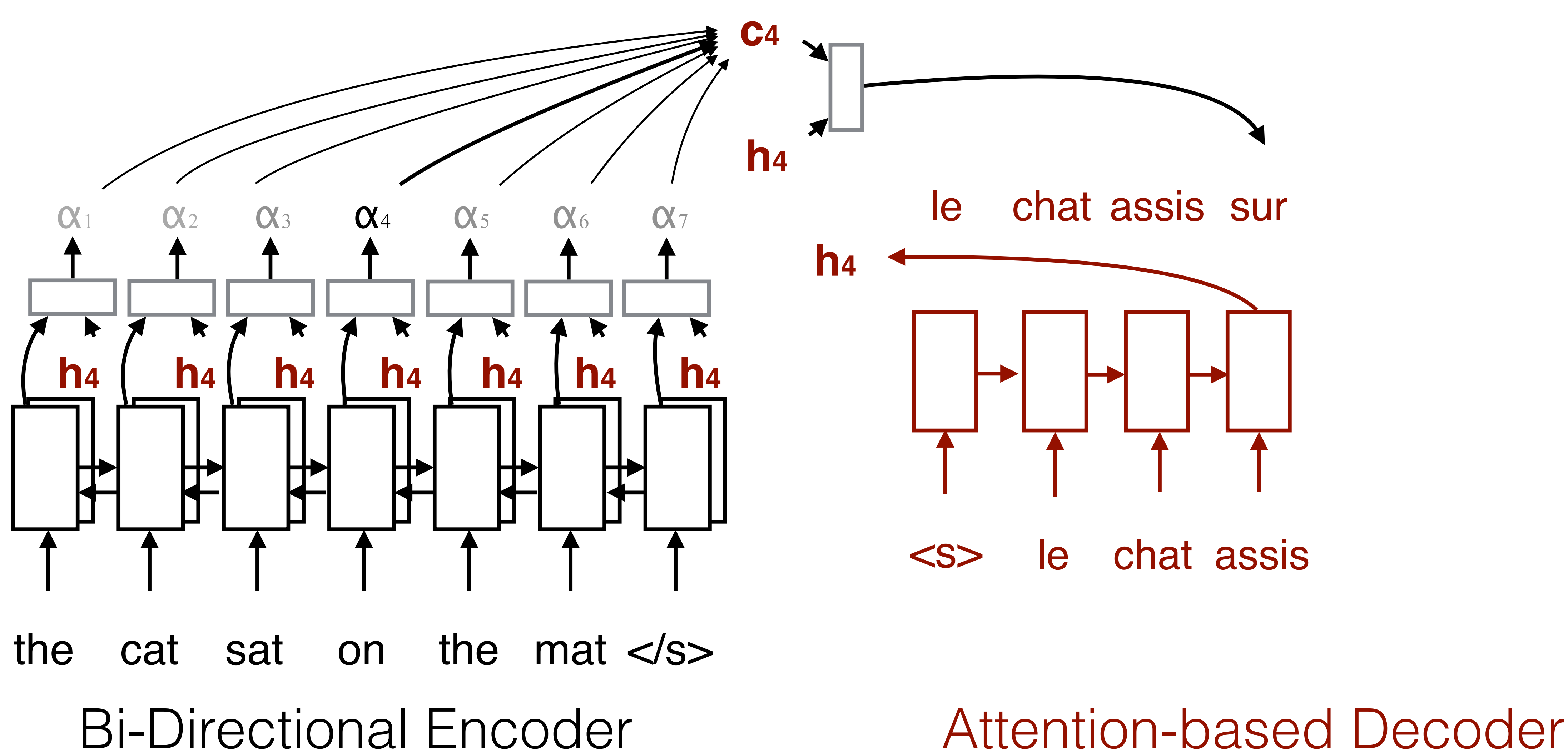
The Attention Mechanism

- Instead of using a single vector as a fixed representation of the input sequence, "**attend**" at each step to the **relevant parts** of the input
- The "**relevance**" of each input element to the current prediction is **computed** via a **feed-forward network** that gets the input element and the current decoder state
- Coined as "**Resolution Preserving**" - longer sequences get longer representations



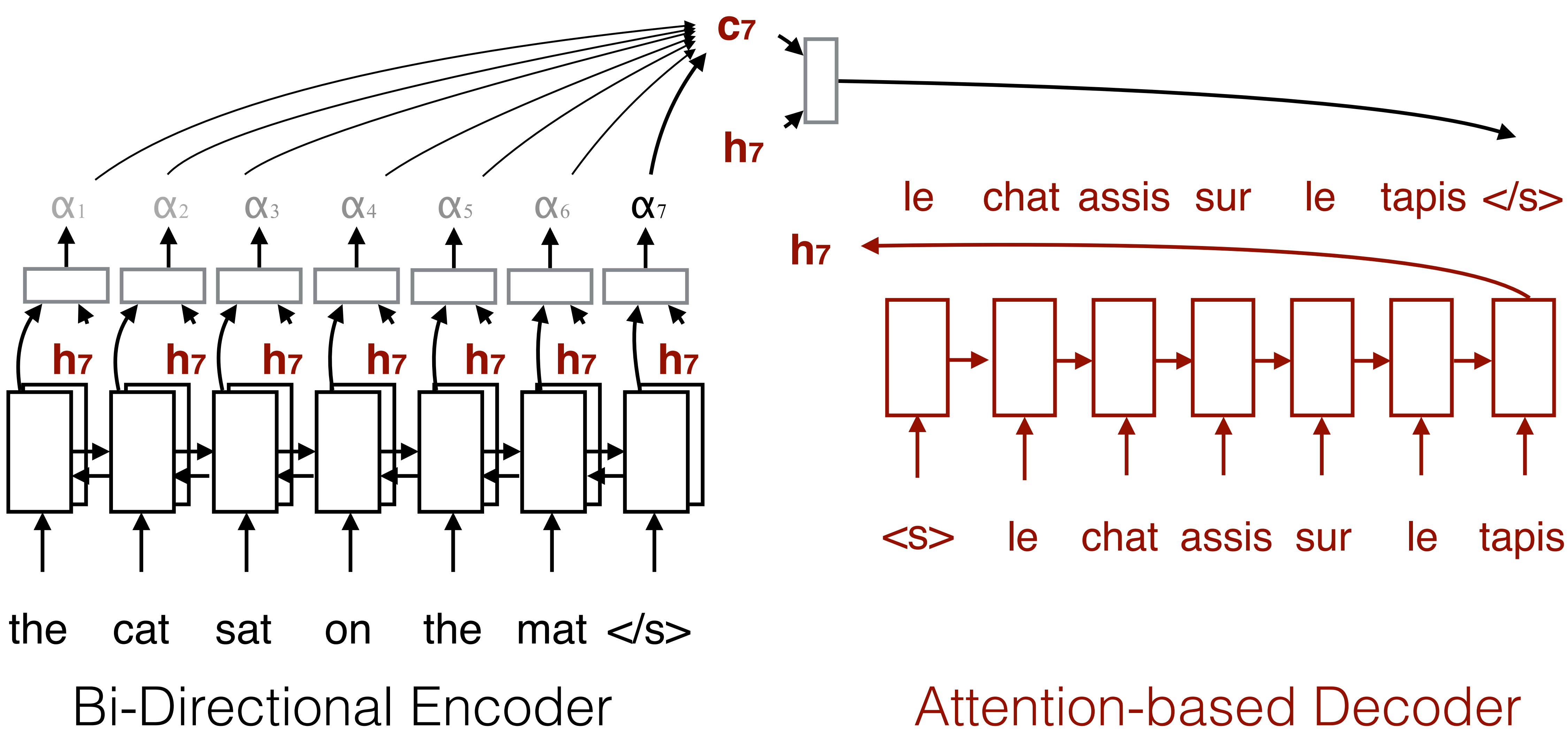
The Attention Mechanism

- Instead of using a single vector as a fixed representation of the input sequence, "**attend**" at each step to the **relevant parts** of the input
- The "**relevance**" of each input element to the current prediction is **computed** via a **feed-forward network** that gets the input element and the current decoder state
- Coined as "**Resolution Preserving**" - longer sequences get longer representations



The Attention Mechanism

- Instead of using a single vector as a fixed representation of the input sequence, "**attend**" at each step to the **relevant parts** of the input
- The "**relevance**" of each input element to the current prediction is **computed** via a **feed-forward network** that gets the input element and the current decoder state
- Coined as "**Resolution Preserving**" - longer sequences get longer representations



The Attention Mechanism

The Attention Mechanism

- And a bit more formally - in each decoder step:

The Attention Mechanism

- And a bit more formally - in each decoder step:
  - Compute attention scores for each input element:

$$\mathrm{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) = \tanh(\boldsymbol{W_a}[\boldsymbol{h}_t; \bar{\boldsymbol{h}}_s])$$

The Attention Mechanism

- And a bit more formally - in each decoder step:
  - Compute attention scores for each input element:

$$\operatorname{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) = \tanh(\boldsymbol{W}_{\boldsymbol{a}}[\boldsymbol{h}_t; \bar{\boldsymbol{h}}_s])$$

  - Normalize the attention scores so they sum up to 1:

$$\boldsymbol{a}_t(s) = \operatorname{align}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) = \frac{\exp\left(\operatorname{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s)\right)}{\sum_{s'} \exp\left(\operatorname{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_{s'})\right)}$$

The Attention Mechanism

- And a bit more formally - in each decoder step:
  - Compute attention scores for each input element:

$$\mathrm{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) = \tanh(\boldsymbol{W_a}[\boldsymbol{h}_t; \bar{\boldsymbol{h}}_s])$$

  - Normalize the attention scores so they sum up to 1:

$$\boldsymbol{a}_t(s) = \mathrm{align}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) = \frac{\exp\left(\mathrm{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s)\right)}{\sum_{s'} \exp\left(\mathrm{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_{s'})\right)}$$

  - Compute c_t:

$$c_t = \sum_{j=1}^{T_x} a_j \bar{h}_j$$

The Attention Mechanism

- And a bit more formally - in each decoder step:
  - Compute attention scores for each input element:

$$\operatorname{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) = \tanh(\boldsymbol{W_a}[\boldsymbol{h}_t; \bar{\boldsymbol{h}}_s])$$

  - Normalize the attention scores so they sum up to 1:

$$\boldsymbol{a}_t(s) = \operatorname{align}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) = \frac{\exp\left(\operatorname{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s)\right)}{\sum_{s'} \exp\left(\operatorname{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_{s'})\right)}$$

  - Compute c_t:

$$c_t = \sum_{j=1}^{T_x} a_j \bar{h}_j$$

  - Compute attention output state:

$$\tilde{\boldsymbol{h}}_t = \tanh(\boldsymbol{W_c}[\boldsymbol{c}_t; \boldsymbol{h}_t])$$

The Attention Mechanism

- And a bit more formally - in each decoder step:
  - Compute attention scores for each input element:

$$\mathrm{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) = \tanh(\boldsymbol{W_a}[\boldsymbol{h}_t; \bar{\boldsymbol{h}}_s])$$

  - Normalize the attention scores so they sum up to 1:

$$\boldsymbol{a}_t(s) = \mathrm{align}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) = \frac{\exp\left(\mathrm{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s)\right)}{\sum_{s'} \exp\left(\mathrm{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_{s'})\right)}$$

  - Compute c_t:

$$c_t = \sum_{j=1}^{T_x} a_j \bar{h}_j$$

  - Compute attention output state:

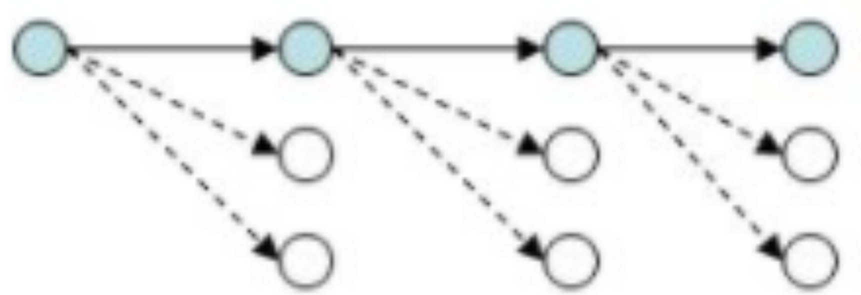
$$\tilde{\boldsymbol{h}}_t = \tanh(\boldsymbol{W_c}[\boldsymbol{c}_t; \boldsymbol{h}_t])$$

  - Compute output probability distribution:

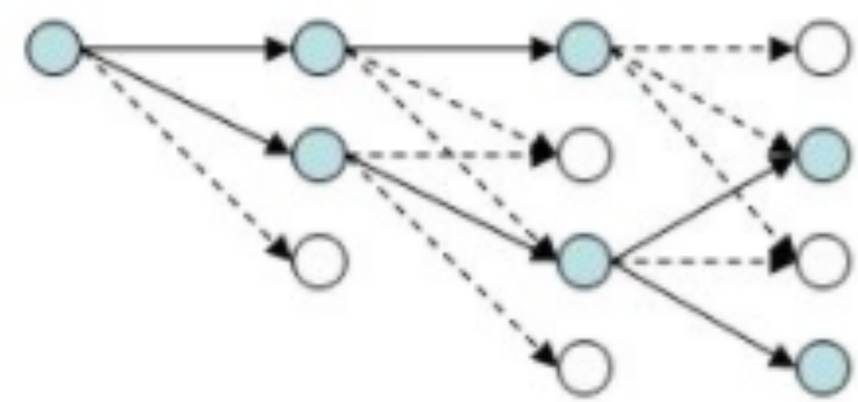
$$p(y_t | y_{<t}, x) = \mathrm{softmax}(\boldsymbol{W_s} \tilde{\boldsymbol{h}}_t)$$

Decoding with Beam Search

- Instead of keeping one best option on each time step, keep k best options which are updated as-you-go
- Usually a small beam size is enough (5-12)

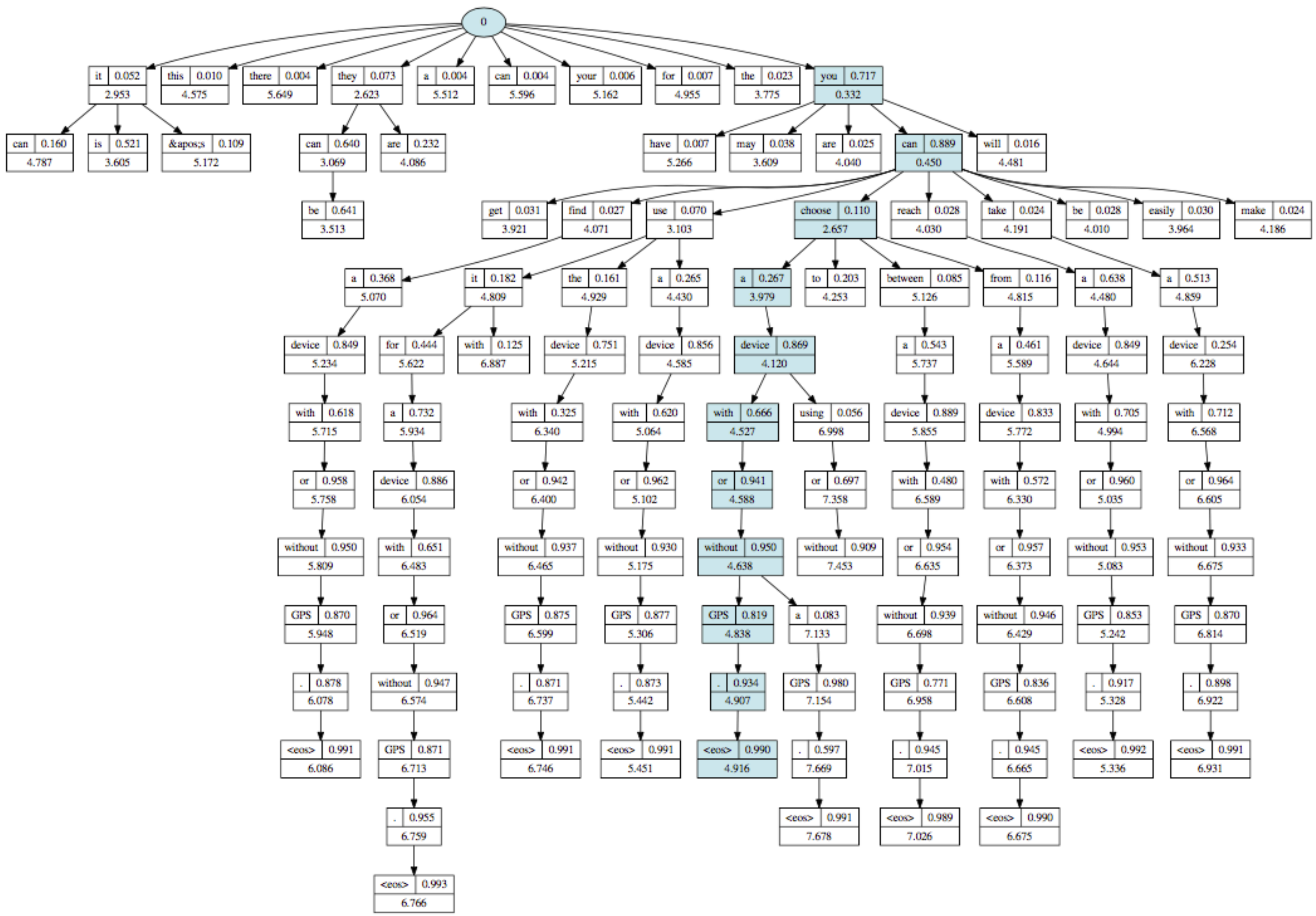


Greedy Search



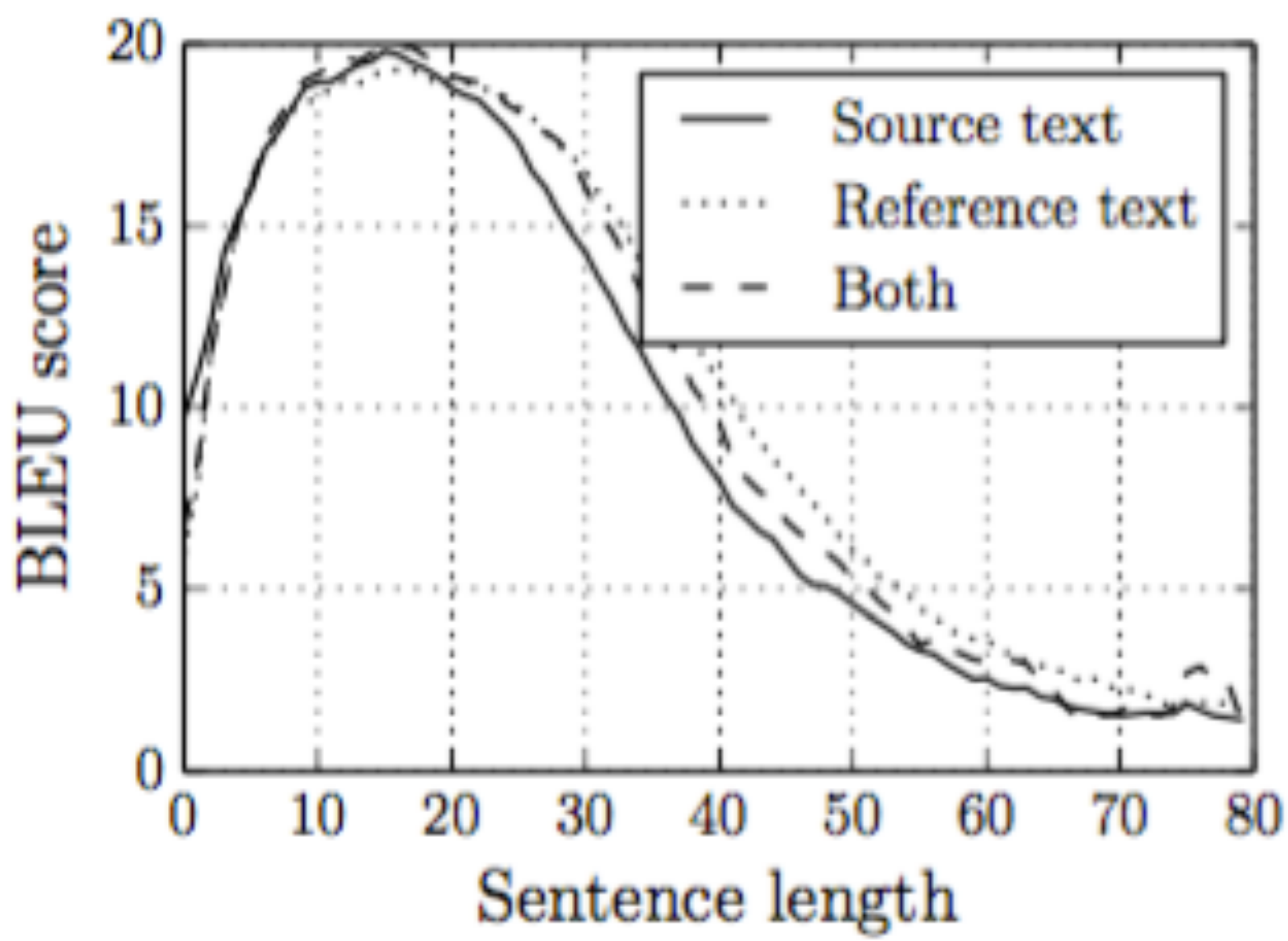
Beam Search (k=2)

Decoding with Beam Search



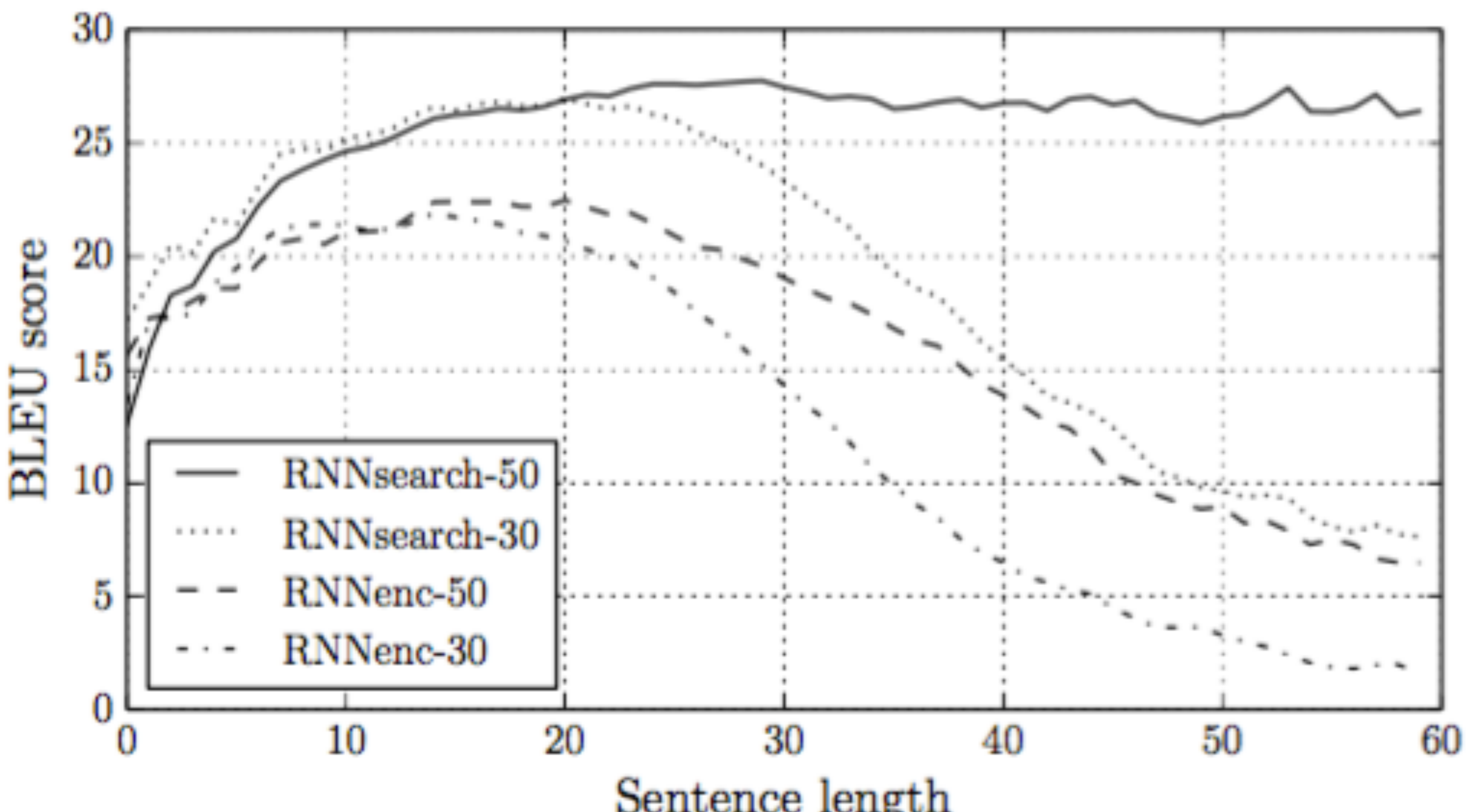
BLEU by Sentence Length - No Attention

- Long sentences are very hard as they are "compressed" to a fixed length vector



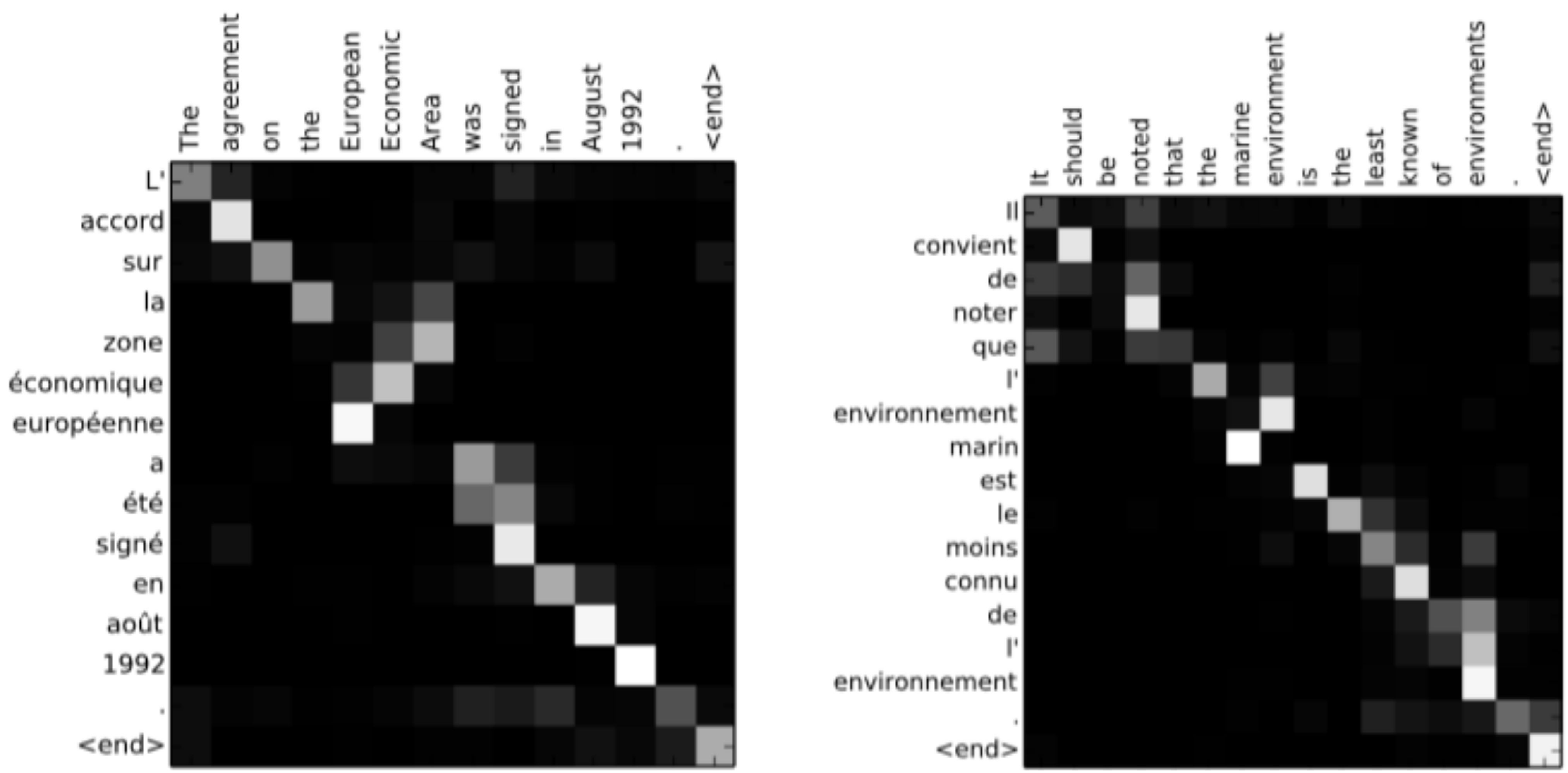
BLEU by Sentence Length - With Attention

- The attention mechanism overcomes the issue

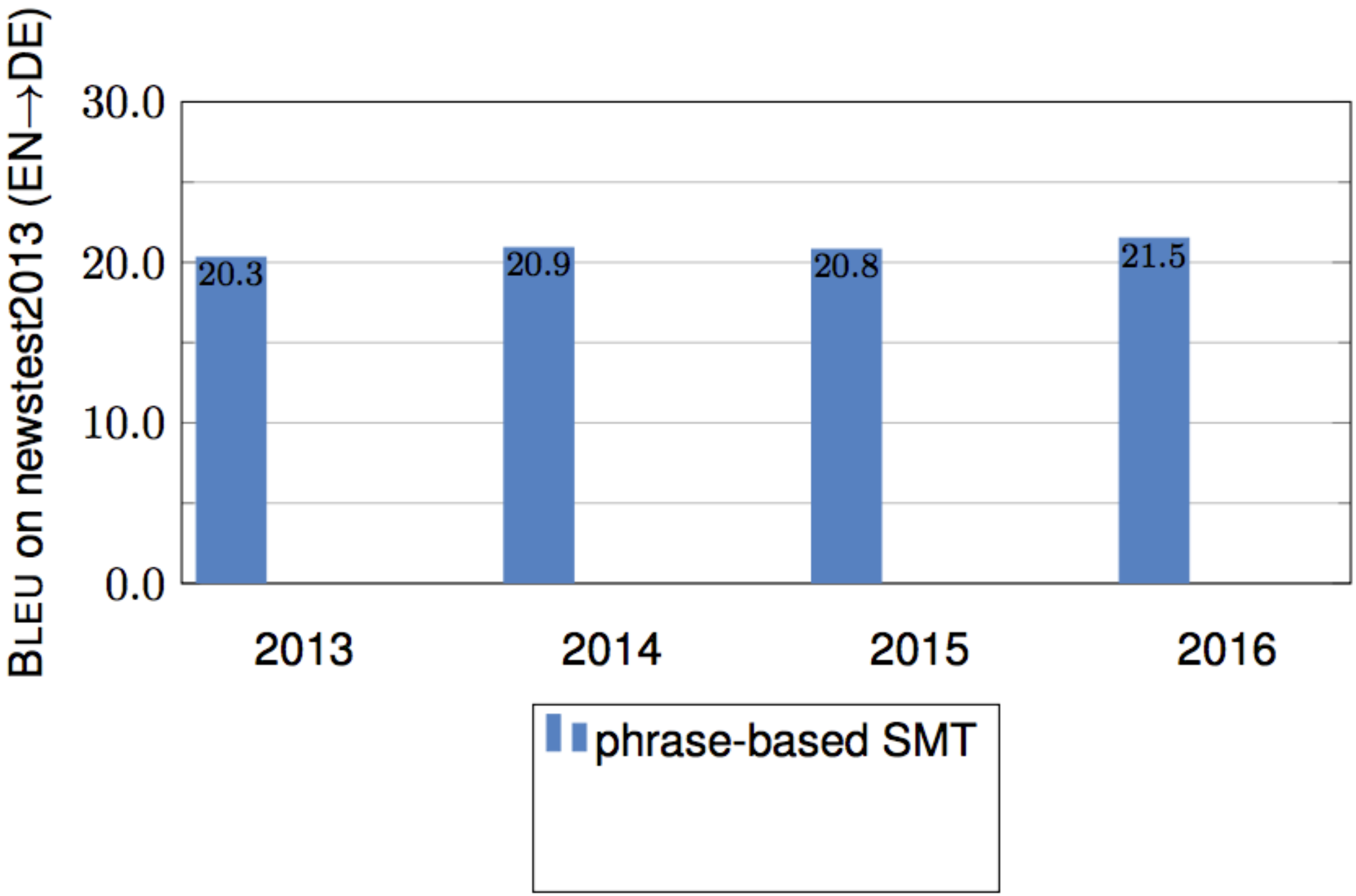


Results - With Attention

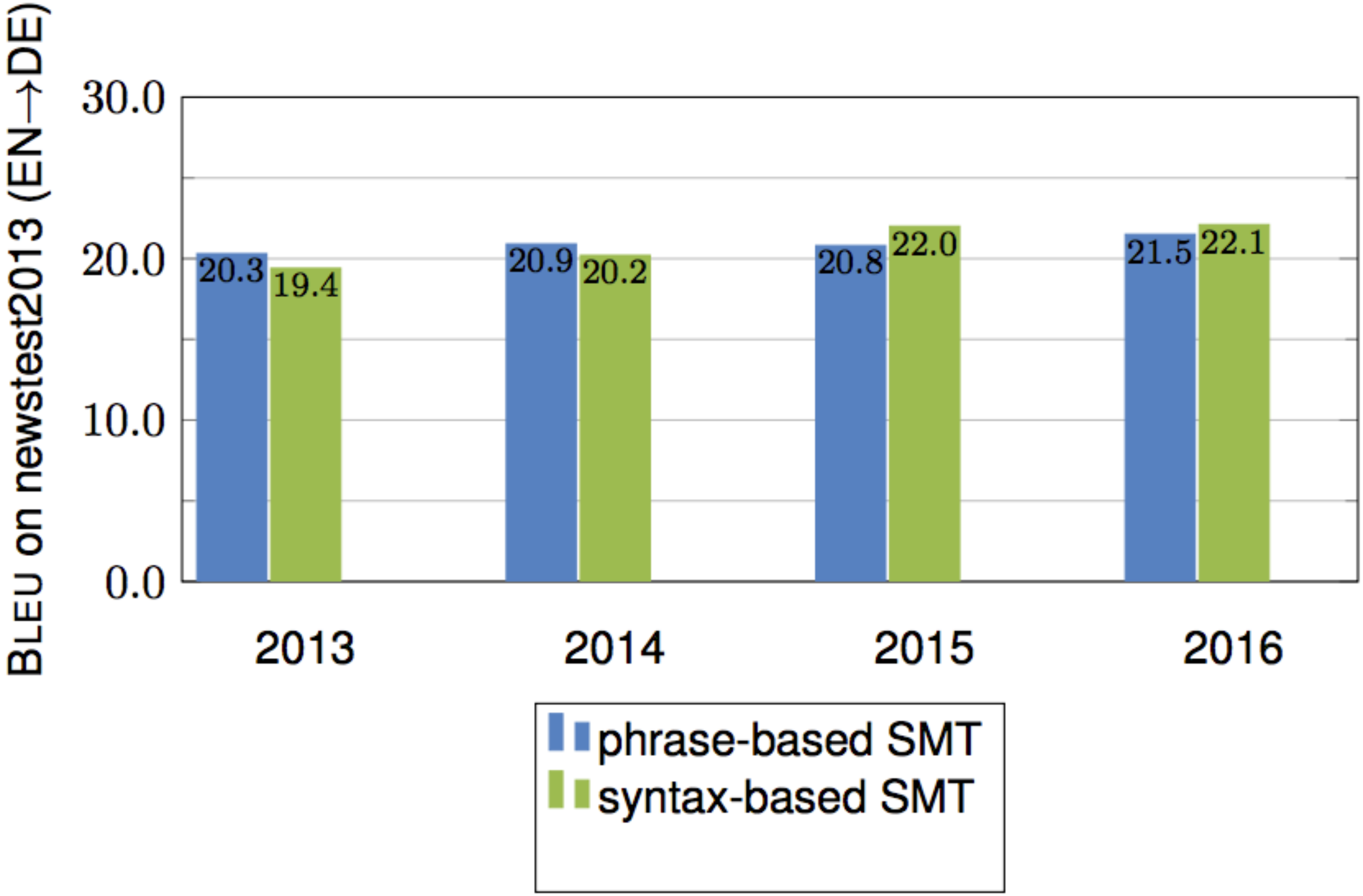
- The model learns nice alignments as a by-product (important for **interpretation**):



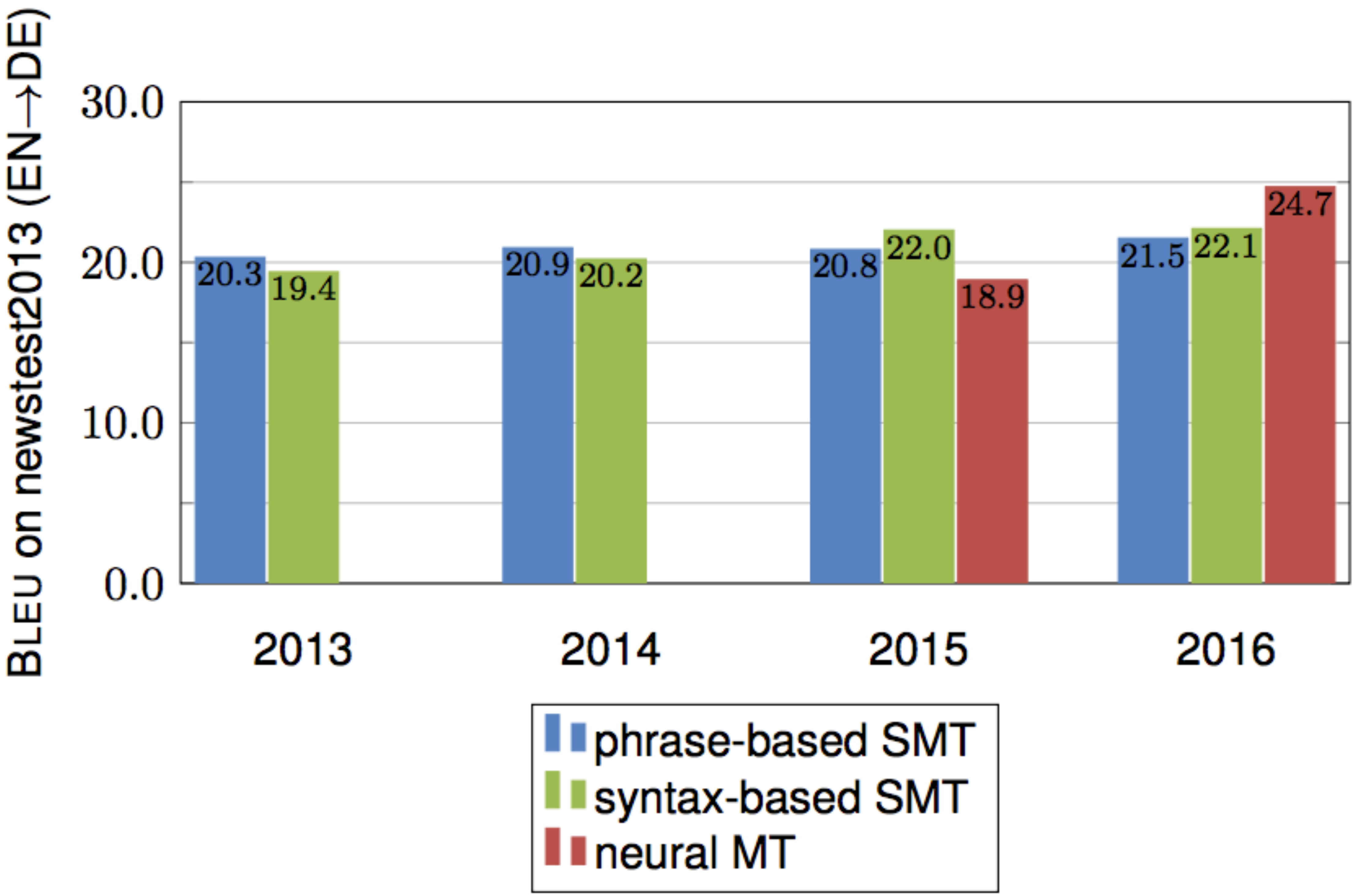
Edinburgh's* WMT results over the years



Edinburgh's* WMT results over the years



Edinburgh's* WMT results over the years



What made NMT win? (Sennrich et. al. , 2016)

What made NMT win? (Sennrich et. al. , 2016)

What made NMT win? (Sennrich et. al. , 2016)

- BPE - work at **sub-word** level to enable an open vocabulary

’l o w e s t </w>’ $\xrightarrow{\text{BPE}}$ ’low est</w>’

What made NMT win? (Sennrich et. al. , 2016)

- BPE - work at **sub-word** level to enable an open vocabulary

'l o w e s t </w>' $\xrightarrow{\text{BPE}}$ 'low est</w>'

- Use **monolingual data** for training through **back-translation**

english (real-mono) $\longrightarrow$ german (**synthetic**)

What made NMT win? (Sennrich et. al. , 2016)

- BPE - work at **sub-word** level to enable an open vocabulary

'l o w e s t </w>' $\xrightarrow{\text{BPE}}$ 'low est</w>'

- Use **monolingual data** for training through **back-translation**

english (real-mono) → german (**synthetic**)

german (real-parallel+**synthetic**) → english (real-parallel+real-mono)

What made NMT win? (Sennrich et. al. , 2016)

- BPE - work at **sub-word** level to enable an open vocabulary

'l o w e s t </w>' $\xrightarrow{\text{BPE}}$ 'low est</w>'

- Use **monolingual data** for training through **back-translation**

english (real-mono) → german (**synthetic**)

german (real-parallel+**synthetic**) → english (real-parallel+real-mono)

- **Bi-directional** decoding:

a b c → **x y z**

What made NMT win? (Sennrich et. al. , 2016)

- BPE - work at **sub-word** level to enable an open vocabulary

'l o w e s t </w>' $\xrightarrow{\text{BPE}}$ 'low est</w>'

- Use **monolingual data** for training through **back-translation**

english (real-mono) → german (**synthetic**)

german (real-parallel+**synthetic**) → english (real-parallel+real-mono)

- **Bi-directional** decoding:

a b c → **x y z**

a b c → **z y x**

The Transformer Architecture

The Transformer Architecture

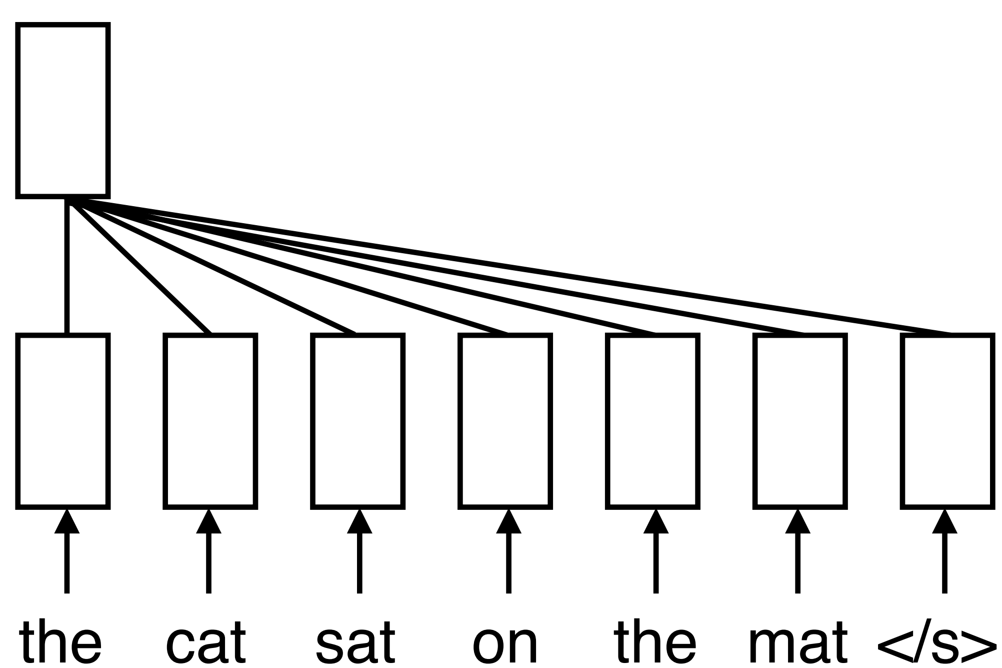
- Vaswani et al. (2017)

The Transformer Architecture

- Vaswani et al. (2017)
- Main idea: use multiple **self-attention** layers instead of recurrence

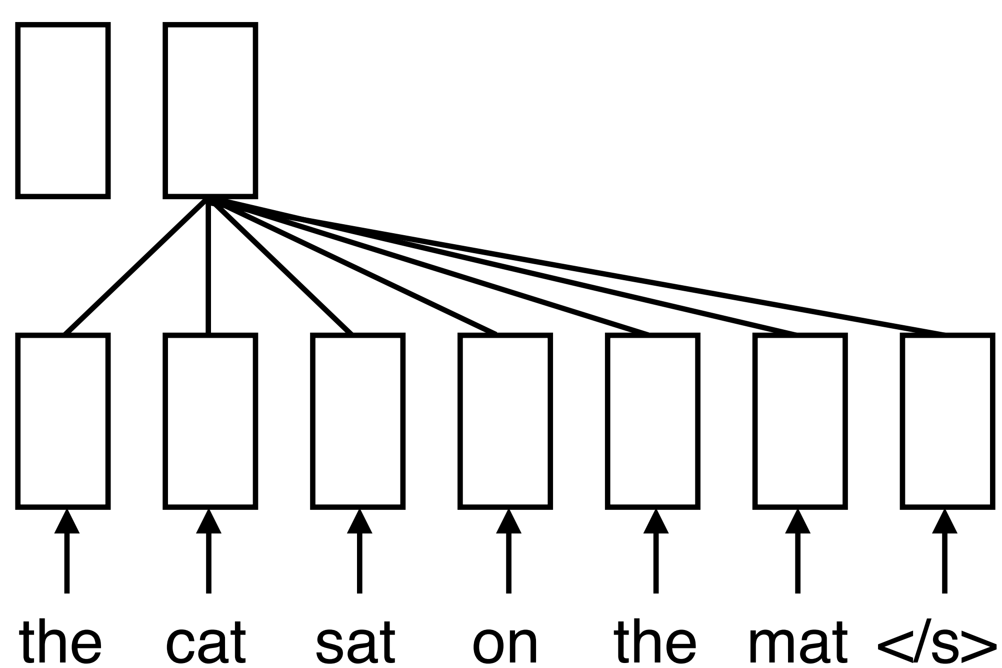
The Transformer Architecture

- Vaswani et al. (2017)
- Main idea: use multiple **self-attention** layers instead of recurrence



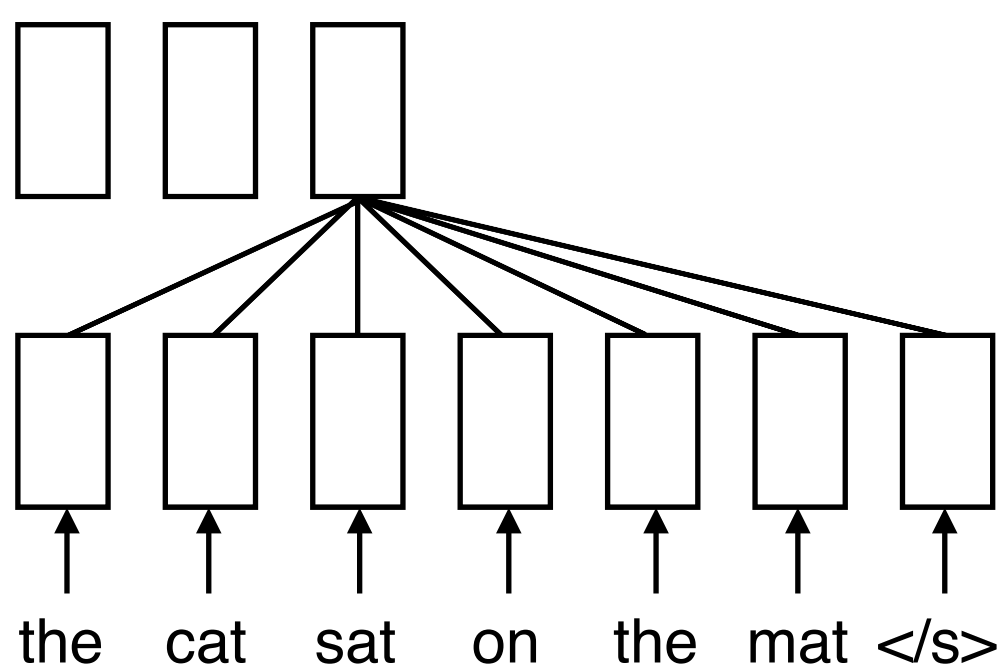
The Transformer Architecture

- Vaswani et al. (2017)
- Main idea: use multiple **self-attention** layers instead of recurrence



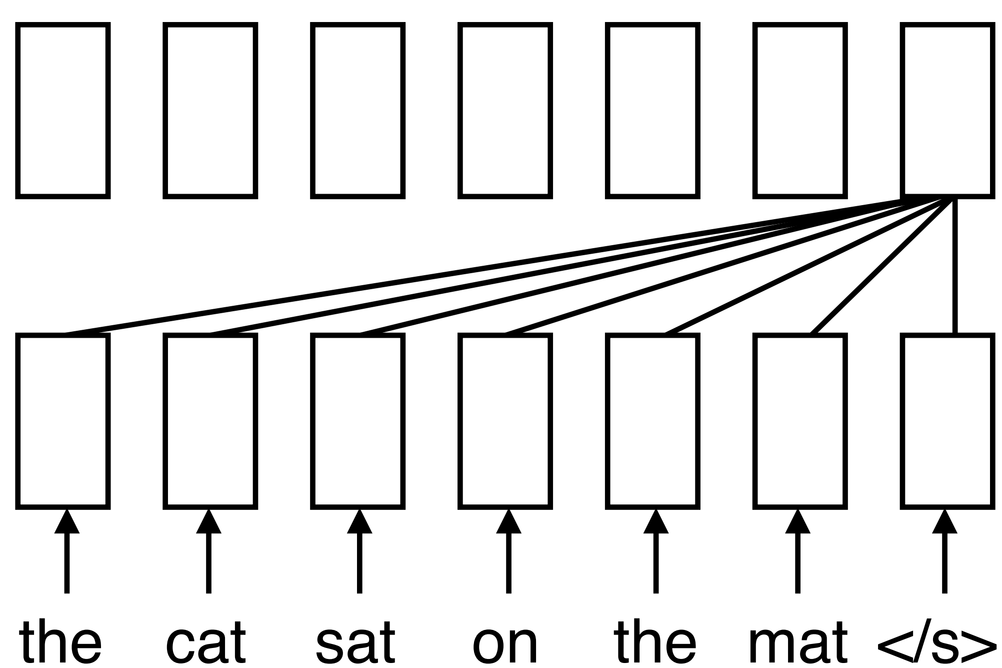
The Transformer Architecture

- Vaswani et al. (2017)
- Main idea: use multiple **self-attention** layers instead of recurrence



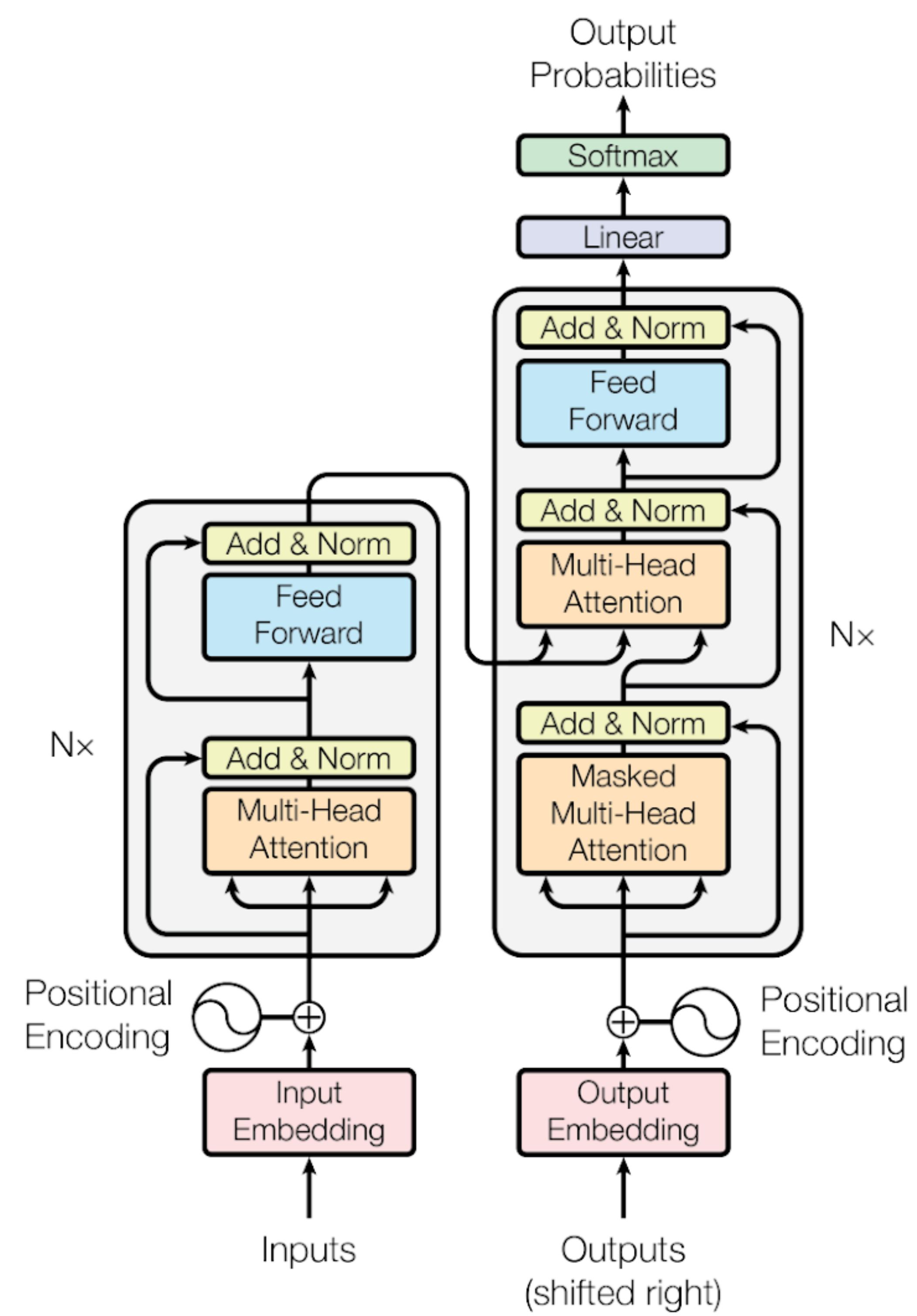
The Transformer Architecture

- Vaswani et al. (2017)
- Main idea: use multiple **self-attention** layers instead of recurrence
- Similar representation power as a bi-LSTM (both left and right context)
- Can be **parallelized** at the sequence level - faster training



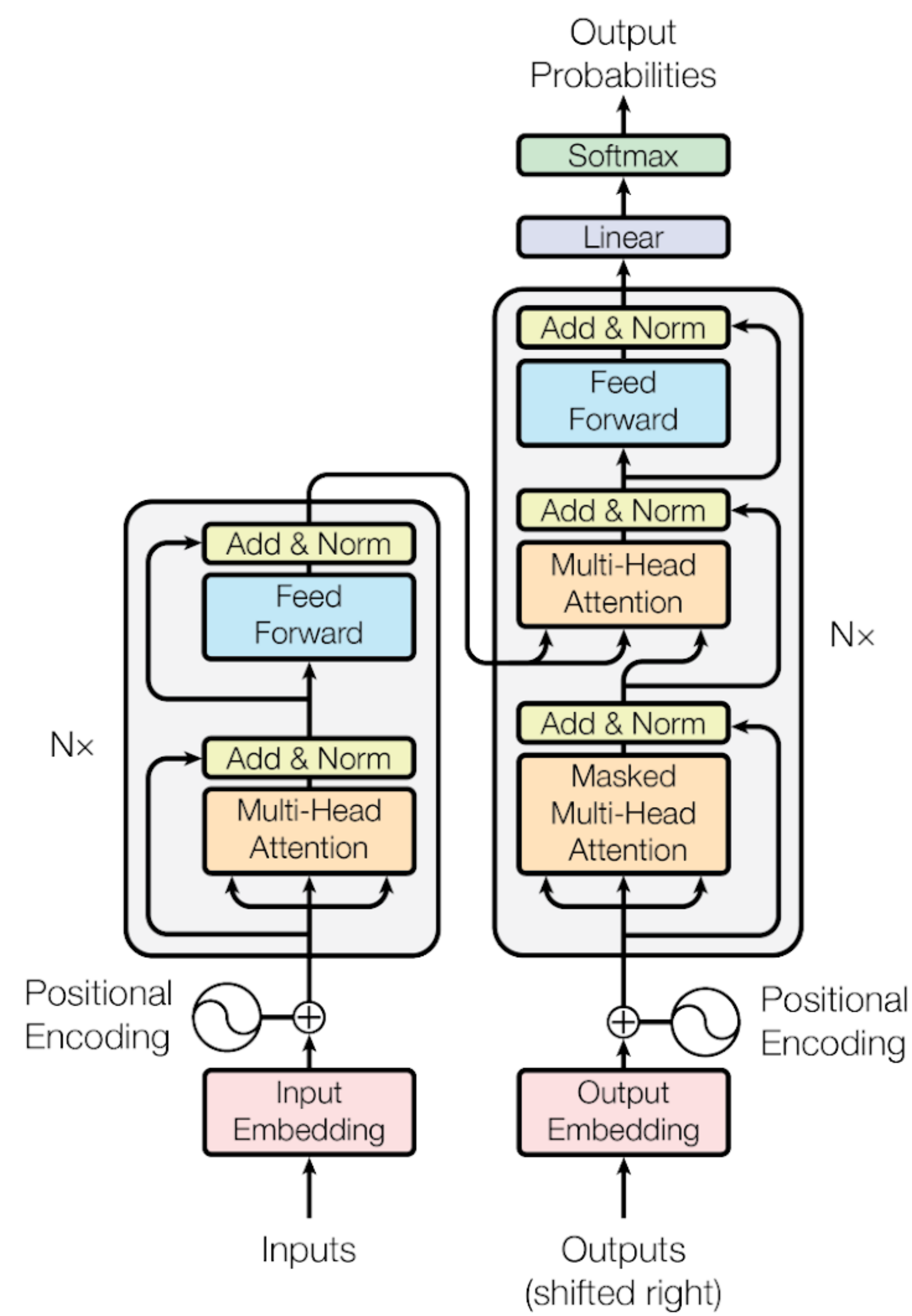
Other Important Details

Other Important Details



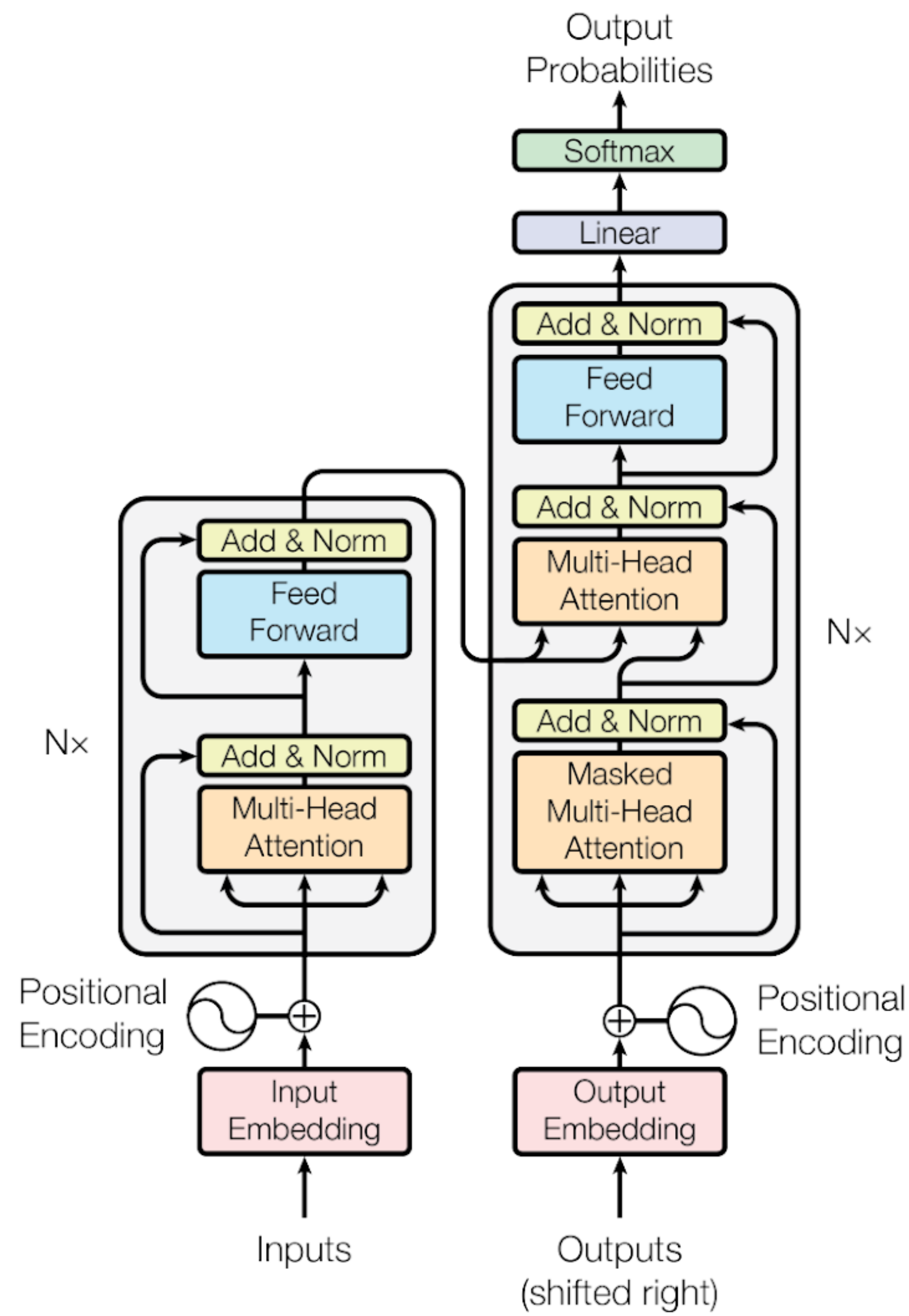
Other Important Details

- Positional encodings



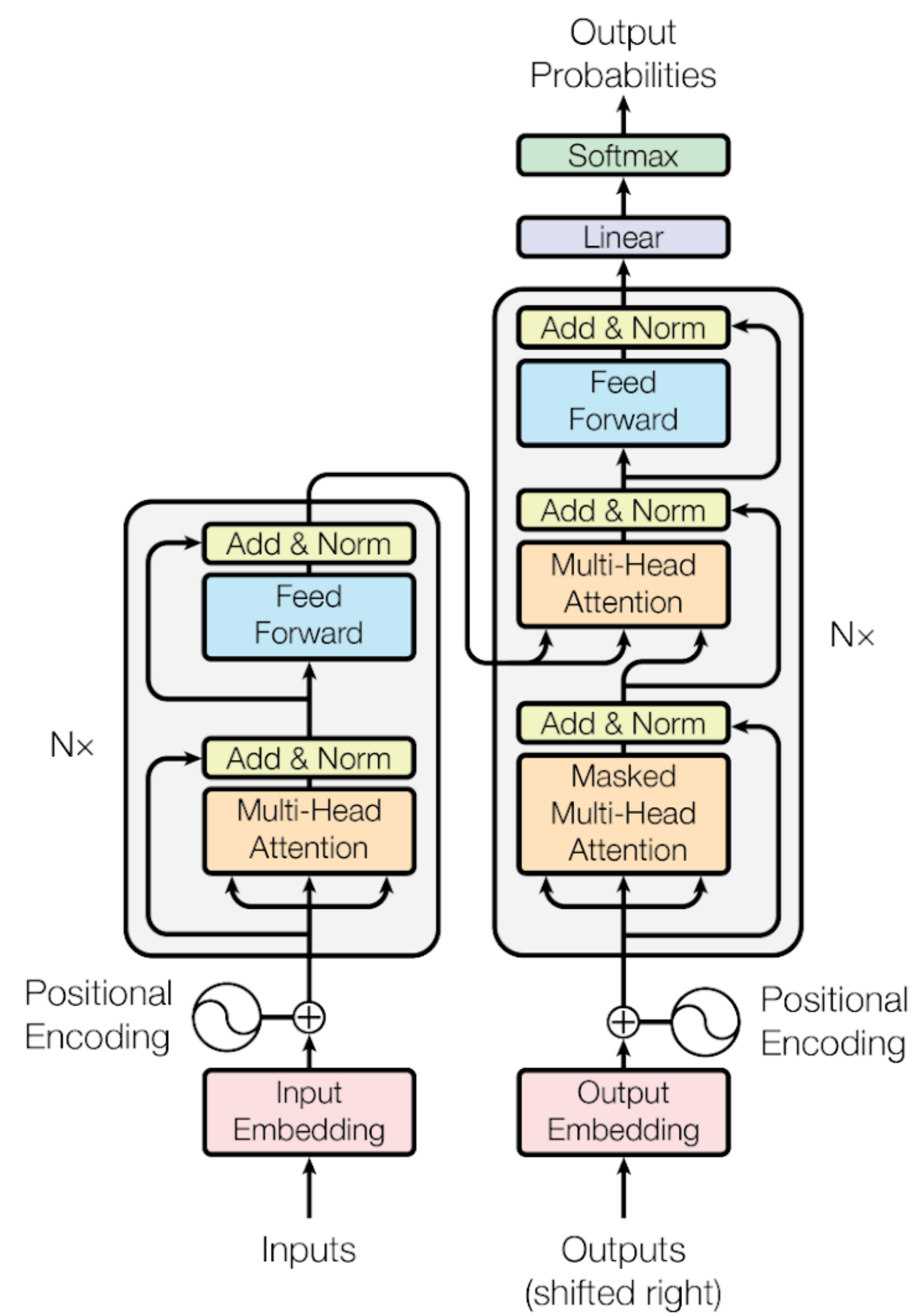
Other Important Details

- Positional encodings
- Multi-head attention



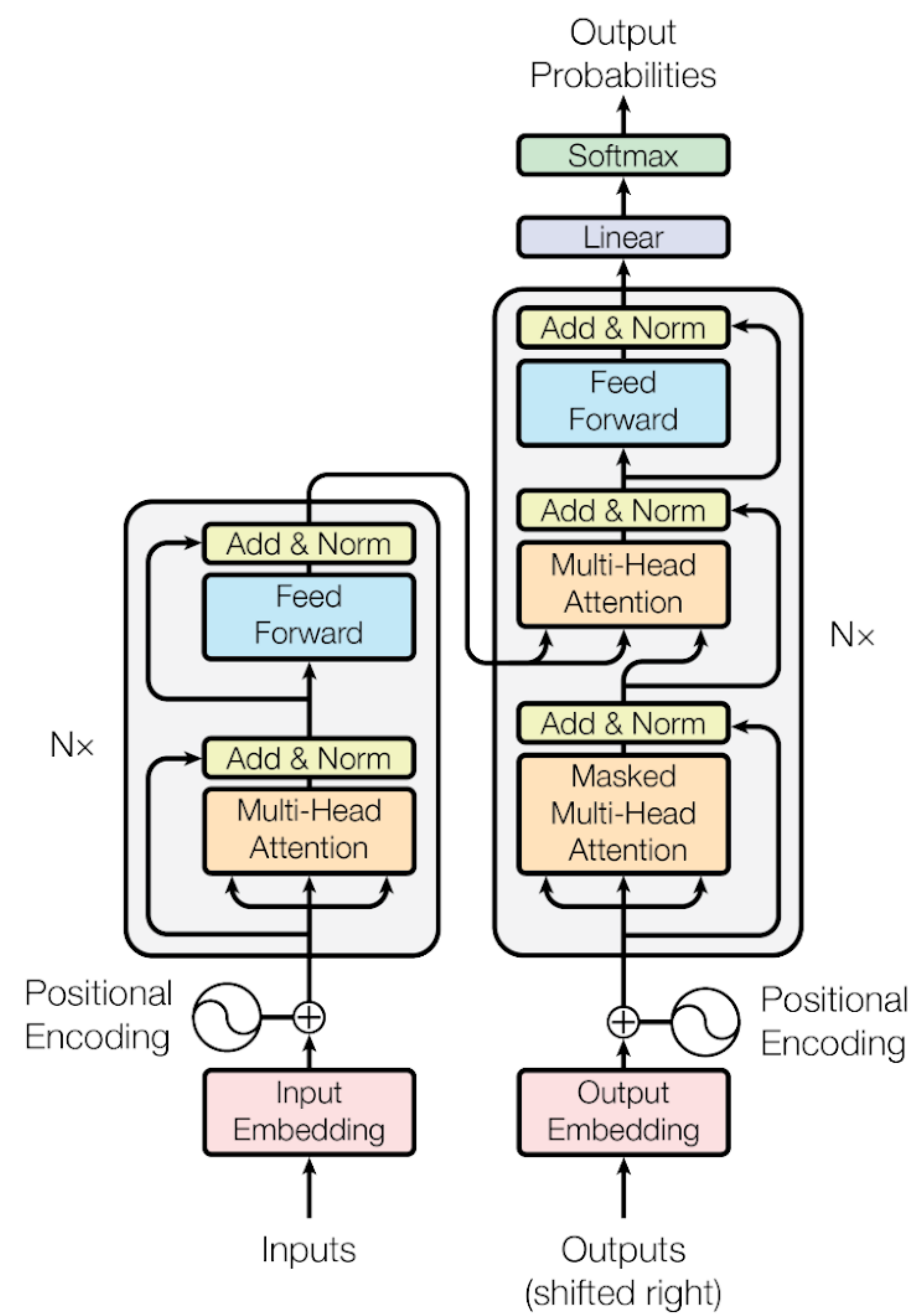
Other Important Details

- Positional encodings
- Multi-head attention
- Layer normalization



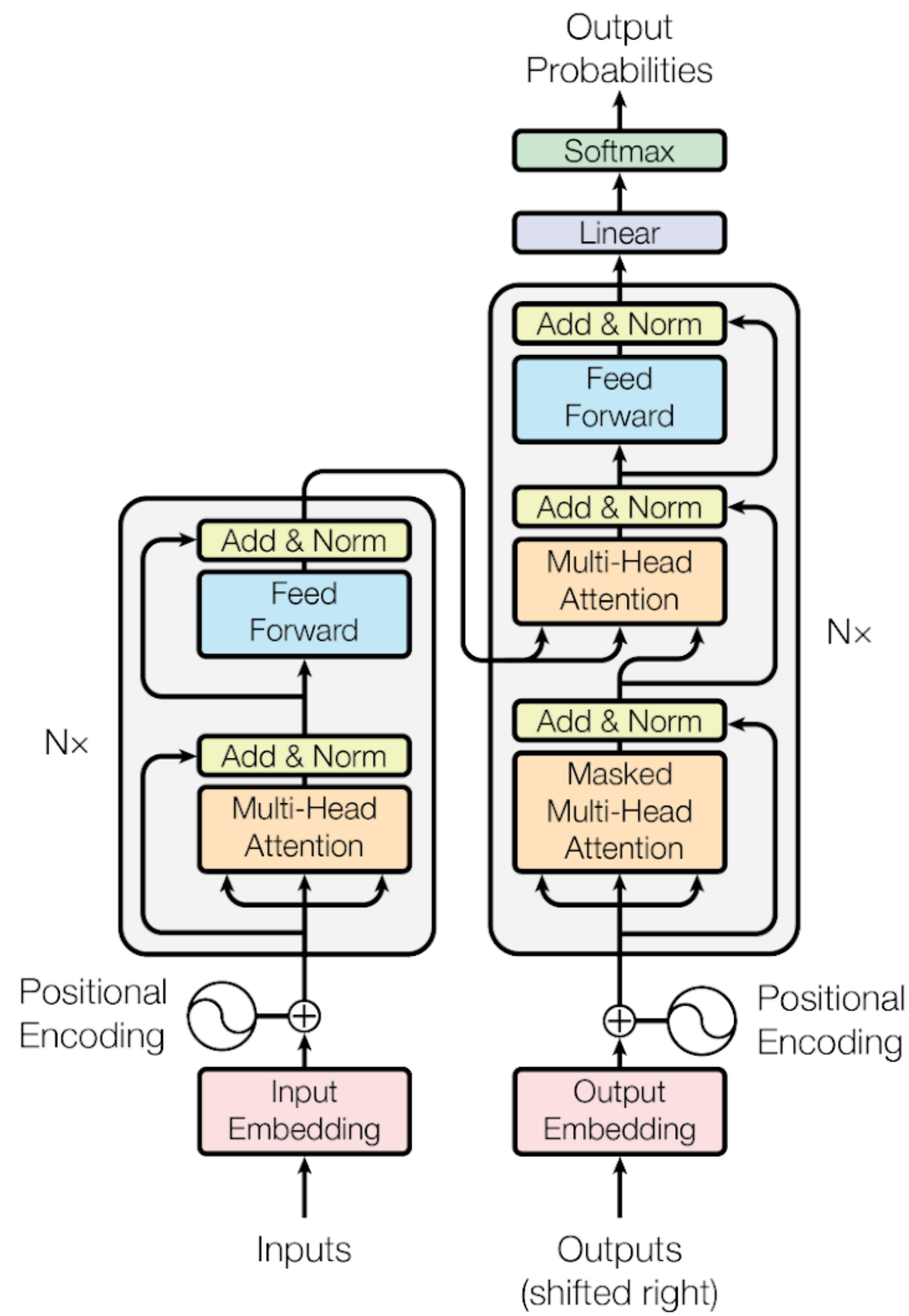
Other Important Details

- Positional encodings
- Multi-head attention
- Layer normalization
- Decoder - masked self attention



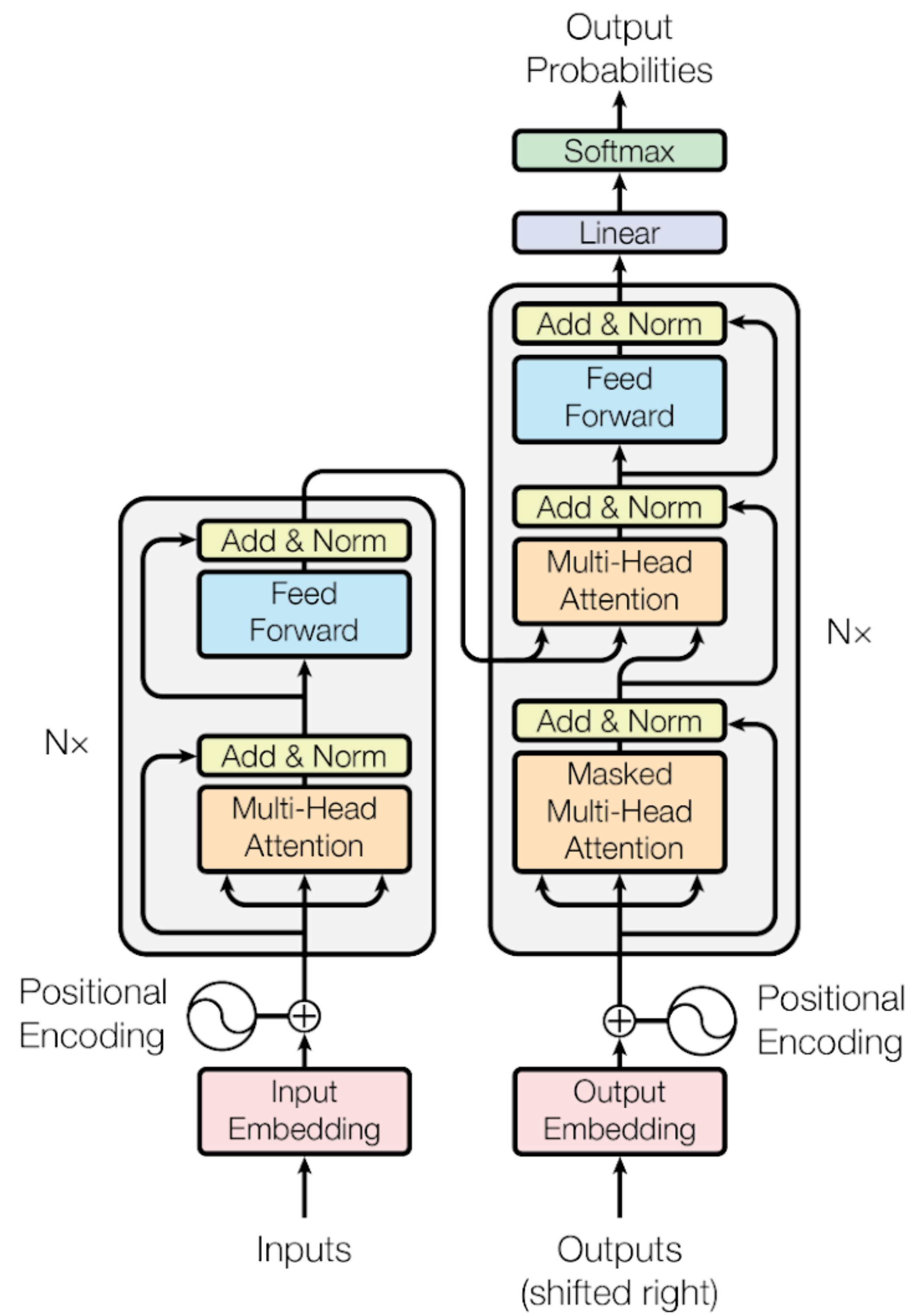
Other Important Details

- Positional encodings
- Multi-head attention
- Layer normalization
- Decoder - masked self attention
- Unlike LSTM based models-



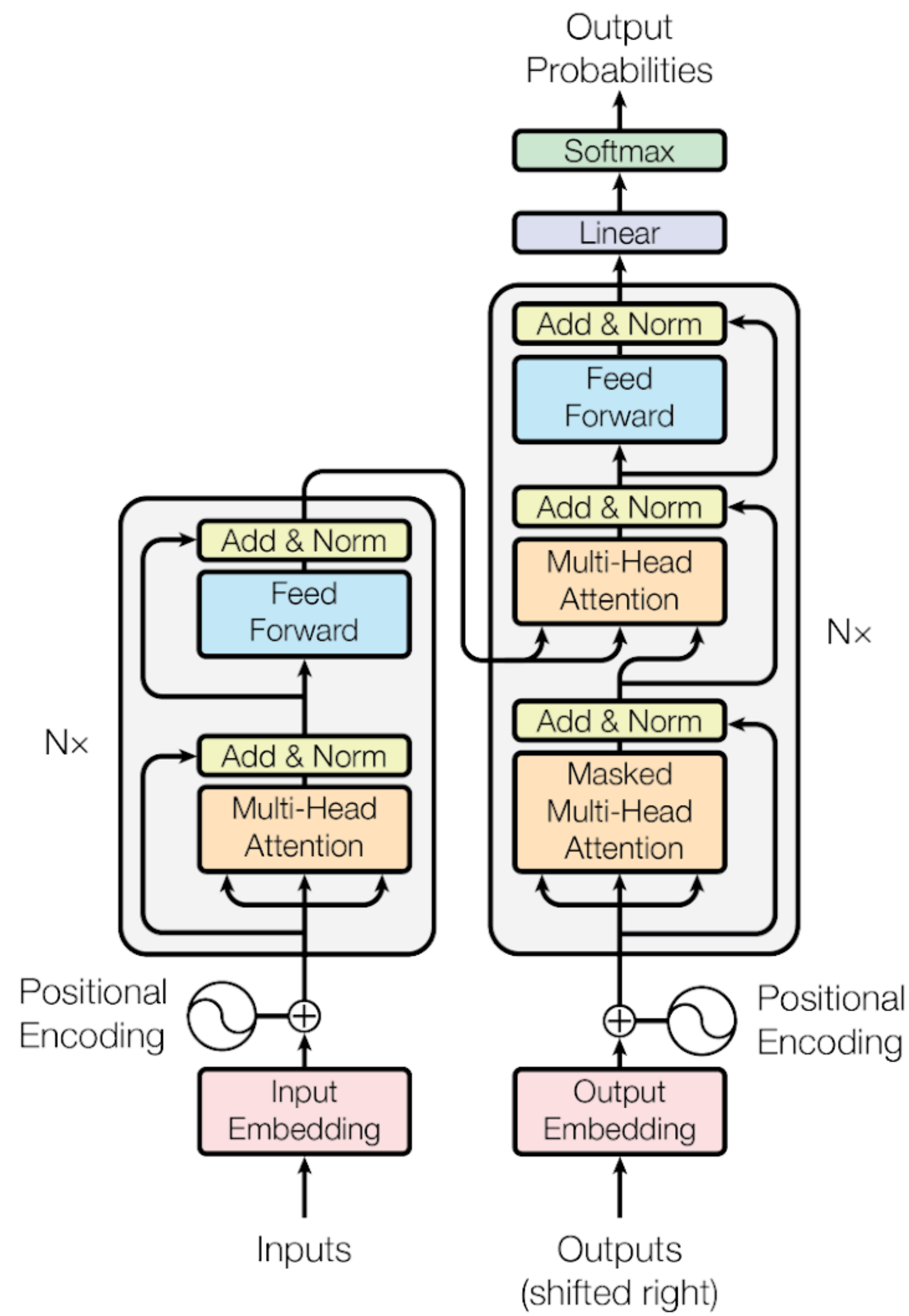
Other Important Details

- Positional encodings
- Multi-head attention
- Layer normalization
- Decoder - masked self attention
- Unlike LSTM based models-
  - encoder-decoder-attention in each layer!



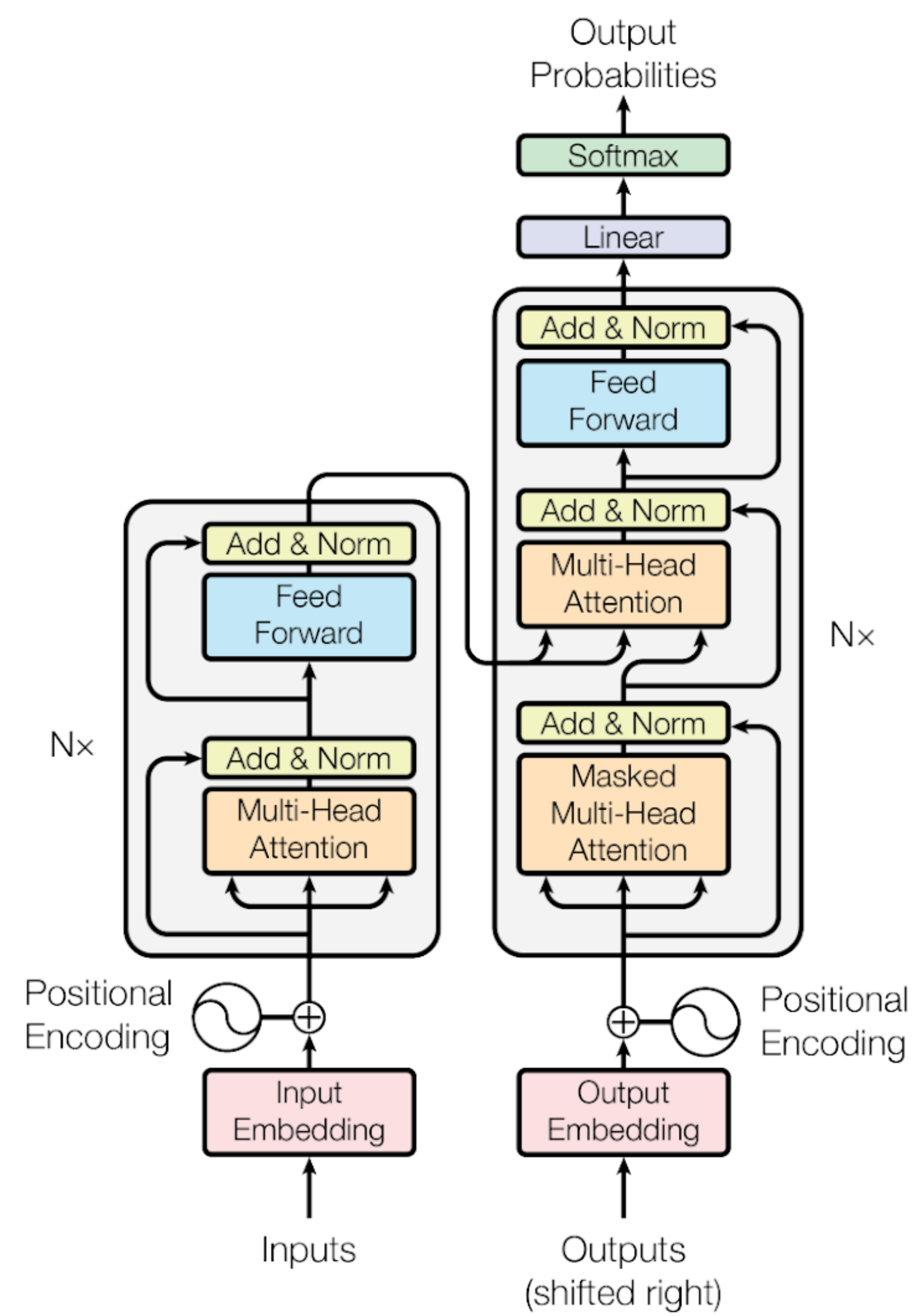
Other Important Details

- Positional encodings
- Multi-head attention
- Layer normalization
- Decoder - masked self attention
- Unlike LSTM based models-
    - encoder-decoder-attention in each layer!
    - Less interpretable



Other Important Details

- Positional encodings
- Multi-head attention
- Layer normalization
- Decoder - masked self attention
- Unlike LSTM based models-
  - encoder-decoder-attention in each layer!
  - Less interpretable
- Learning rate schedule - harder to optimize than LSTM-based models



Multilingual Neural Machine Translation

Motivation

Motivation

- Why Multilingual NMT?

Motivation

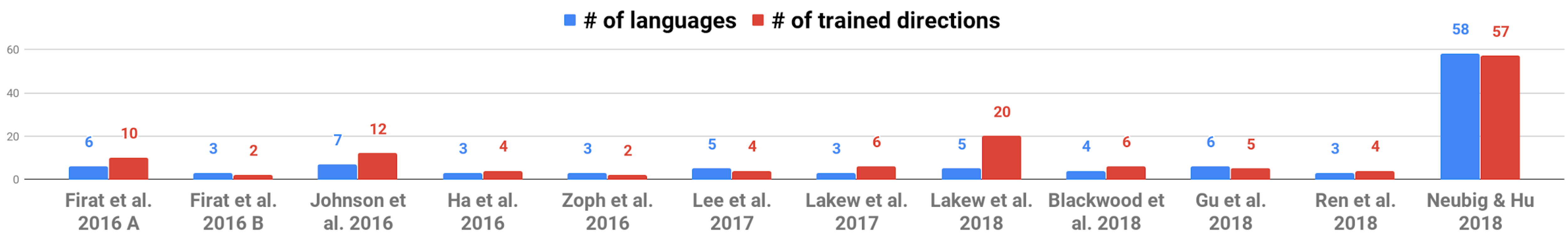
- Why Multilingual NMT?
  - Allows transfer learning: better performance (especially for low resource language pairs)

Motivation

- Why Multilingual NMT?
  - Allows transfer learning: better performance (especially for low resource language pairs)
  - Reduces hardware requirements: much simpler deployment

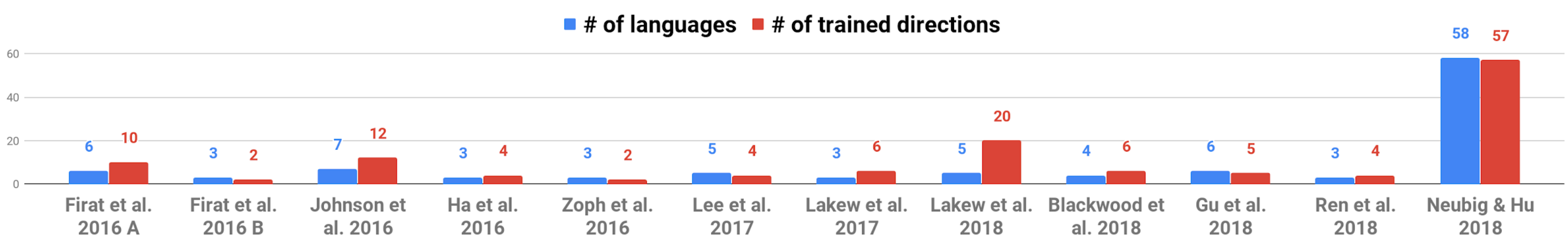
Previous Work

Previous Work



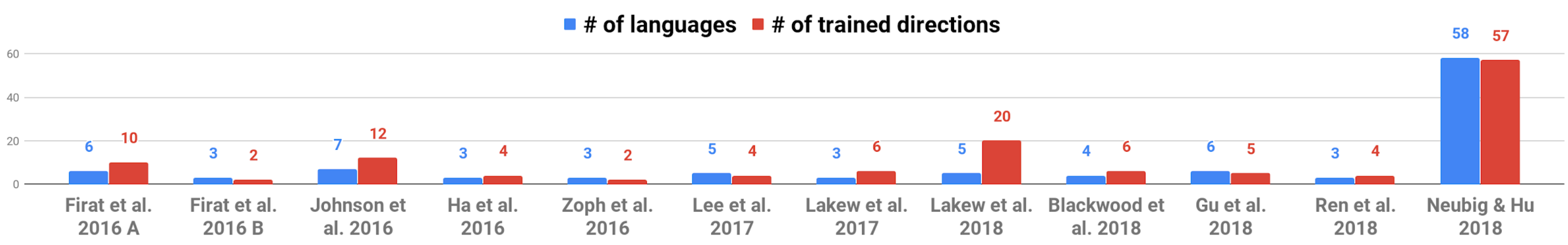
Previous Work

- Up to 5 languages and 20 translation directions



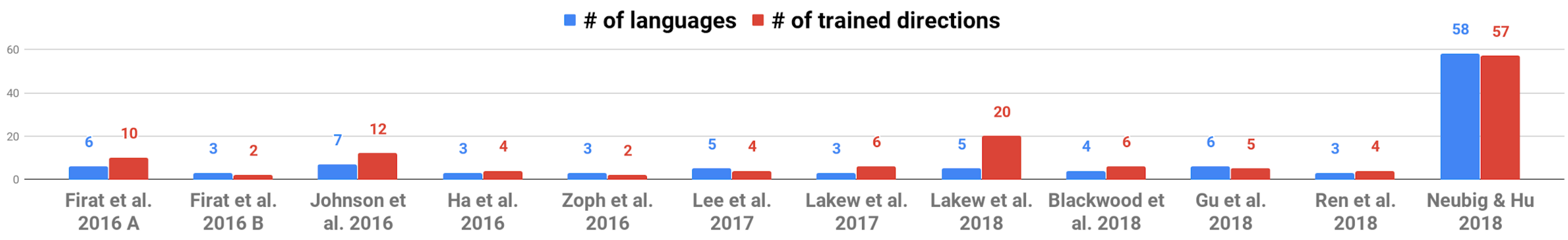
Previous Work

- Up to 5 languages and 20 translation directions
  - One outlier :)



Previous Work

- Up to 5 languages and 20 translation directions
    - One outlier :)
- Why stop here?



Our Work

Our Work

- “Massively Multilingual” NMT

Our Work

- “Massively Multilingual” NMT
- Scale a single NMT model to support 103 languages - still works!

Our Work

- “Massively Multilingual” NMT
- Scale a single NMT model to support 103 languages - still works!
- Effective in low resource settings - state of the art results with 58 languages in a single model

Multilingual Models - Lots of Moving Parts

Multilingual Models - Lots of Moving Parts

- **Multilinguality** - How many languages? (our main focus)

Multilingual Models - Lots of Moving Parts

- **Multilinguality** - How many languages? (our main focus)
- **Data settings** - Many-to-one/one-to-many/many-to-many?

Multilingual Models - Lots of Moving Parts

- **Multilinguality** - How many languages? (our main focus)
- **Data settings** - Many-to-one/one-to-many/many-to-many?
- **Vocabulary** - Joint wpm/separate wpms/characters?

Multilingual Models - Lots of Moving Parts

- **Multilinguality** - How many languages? (our main focus)
- **Data settings** - Many-to-one/one-to-many/many-to-many?
- **Vocabulary** - Joint wpm/separate wpms/characters?
- **Capacity** - effect of model size?

Multilingual Models - Lots of Moving Parts

- **Multilinguality** - How many languages? (our main focus)
- **Data settings** - Many-to-one/one-to-many/many-to-many?
- **Vocabulary** - Joint wpm/separate wpms/characters?
- **Capacity** - effect of model size?
- **Parameter Sharing** - share everything or separate components?

Multilingual Models - Lots of Moving Parts

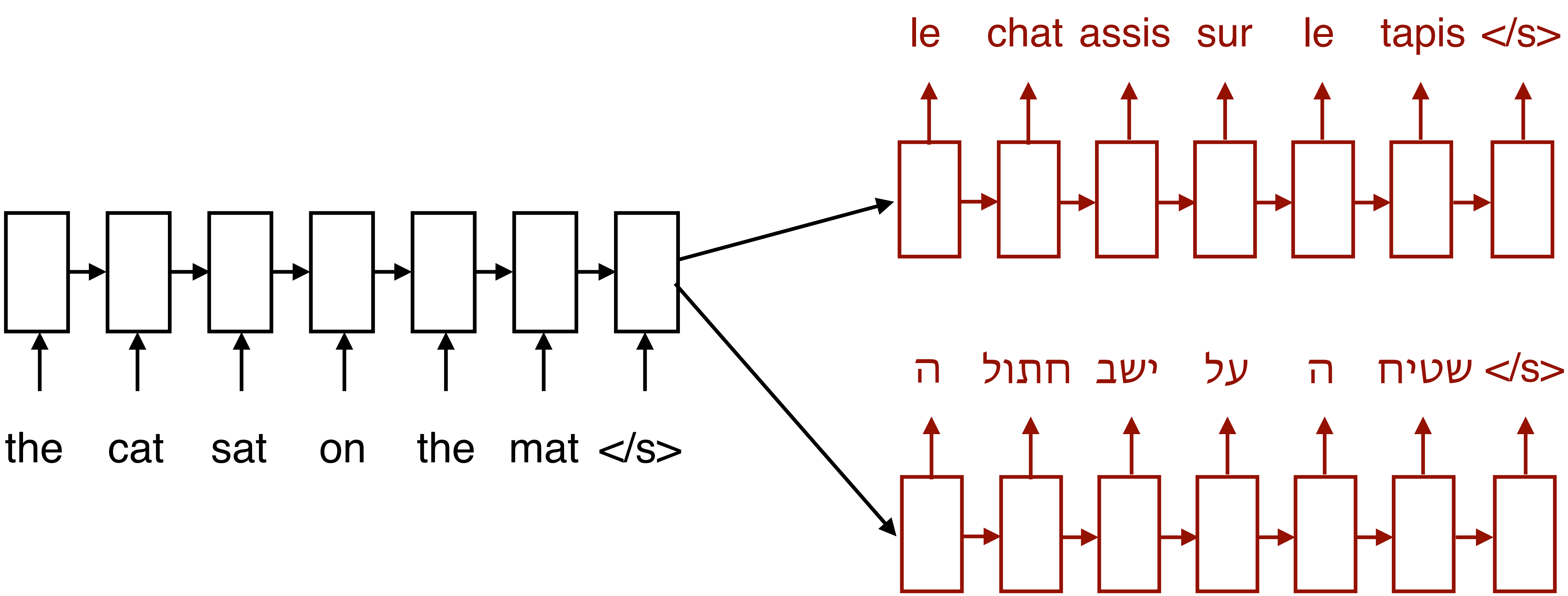
- **Multilinguality** - How many languages? (our main focus)
- **Data settings** - Many-to-one/one-to-many/many-to-many?
- **Vocabulary** - Joint wpm/separate wpms/characters?
- **Capacity** - effect of model size?
- **Parameter Sharing** - share everything or separate components?
- **Loss Functions** - tailored multilingual loss functions?

Multilingual Models - Lots of Moving Parts

- **Multilinguality** - How many languages? (our main focus)
- **Data settings** - Many-to-one/one-to-many/many-to-many?
- **Vocabulary** - Joint wpm/separate wpms/characters?
- **Capacity** - effect of model size?
- **Parameter Sharing** - share everything or separate components?
- **Loss Functions** - tailored multilingual loss functions?
- **Optimization** - Individual pair in a single batch or mixed batches?

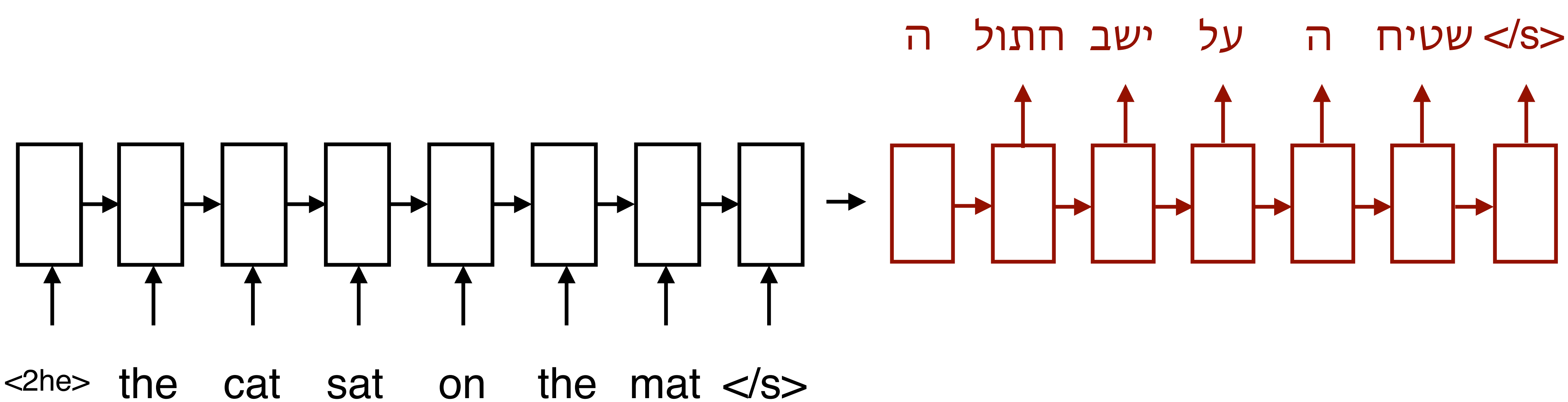
Multilingual NMT Methods

- **Separate** Encoder/Decoder per language (Dong et al. 2015, Firat et al. 2016)
    - Pros - each language its own parameters, no interference
    - Cons - complex models, less parameter sharing



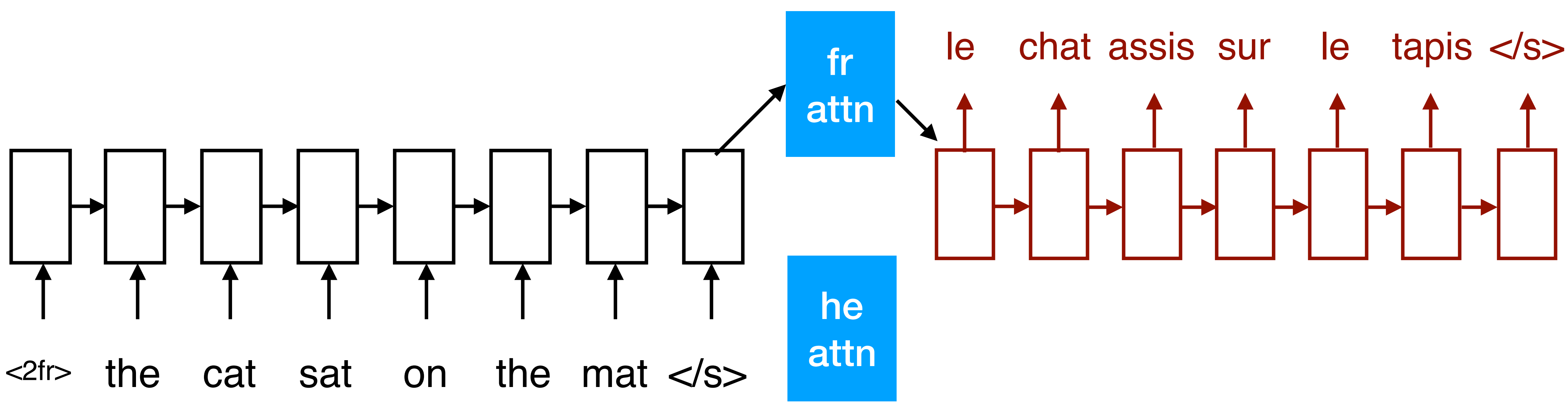
Multilingual NMT Methods

- **Joint** Encoder/Decoder/Attention model (Ha et al. 2016, Johnson et al. 2017)
  - Use a special “language token”
  - Pros - Full parameter sharing, simple (unchanged) model
  - Cons - Languages may interfere each other



Multilingual NMT Methods

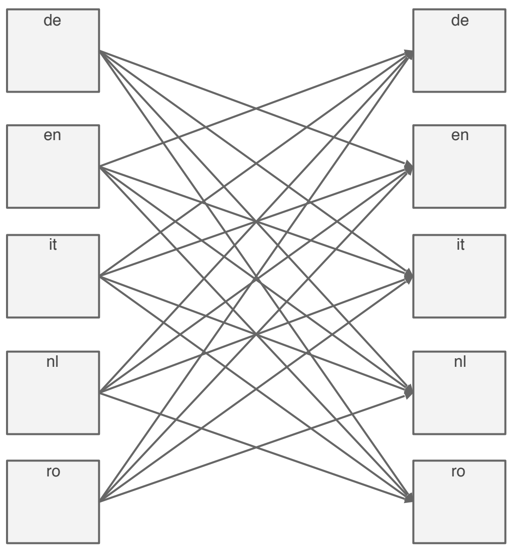
- **“In Between”** Share only some of the parameters, i.e. all but the attention mechanism (i.e. Blackwood et al. 2018, Sachan & Neubig 2018)
  - Pros - may reduce interference
  - Cons - adds implementation complexity



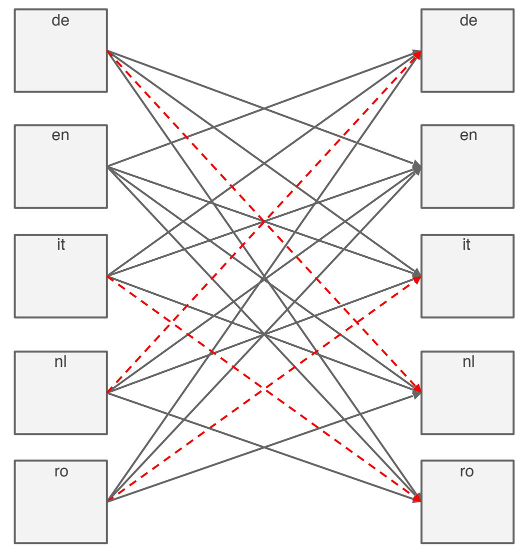
Data Settings

Many-to-Many

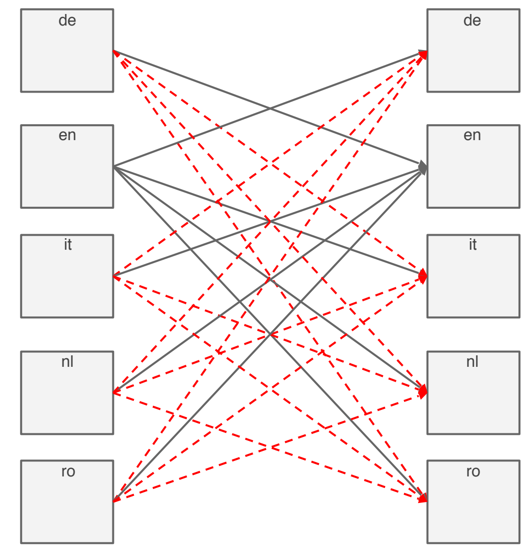
Fully-Supervised



Zero-Shot

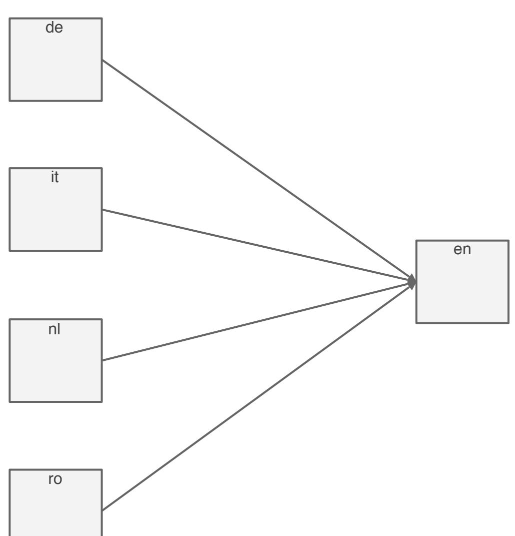


“English Centric”



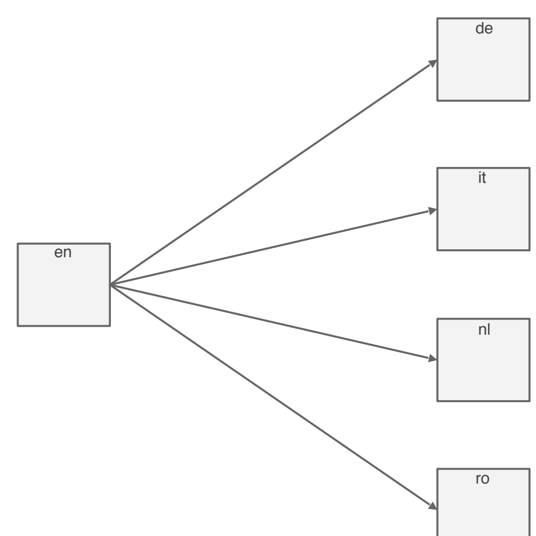
Many-to-One

all-en



One-to-Many

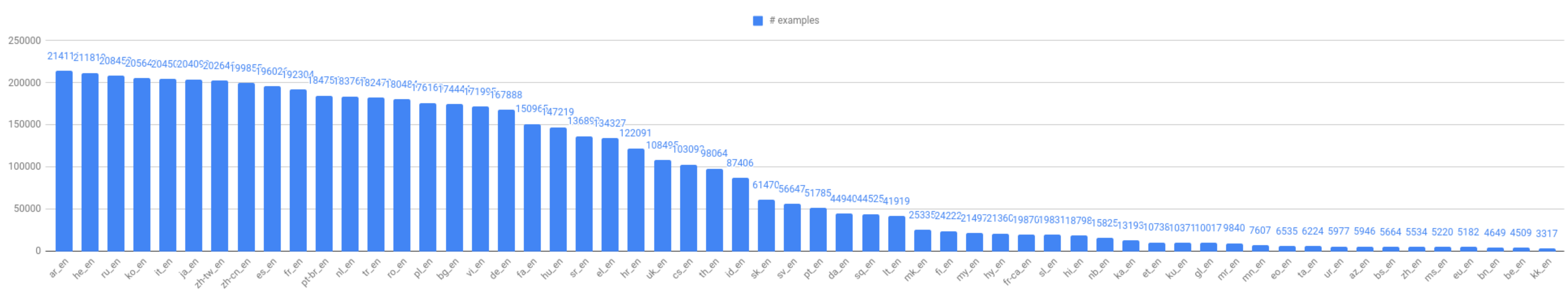
en-all



Experiments - Low Resource

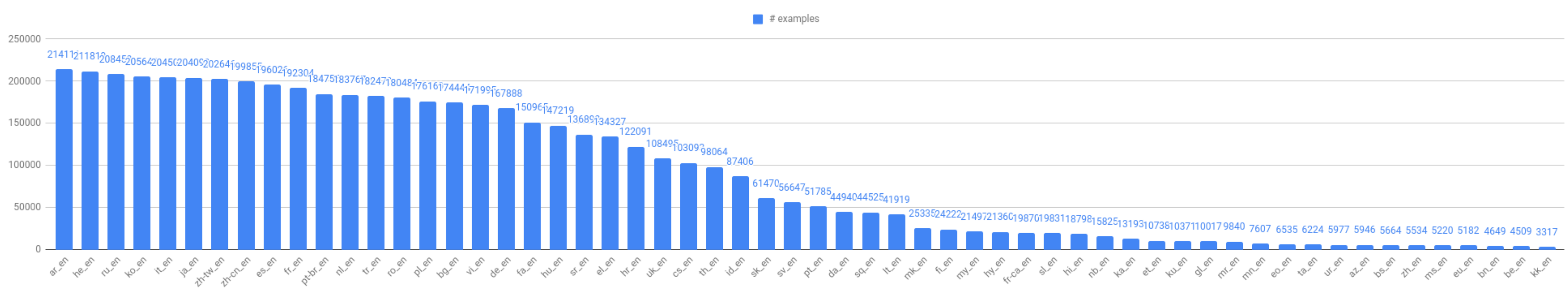
Experiments - Low Resource

- The TED talks dataset



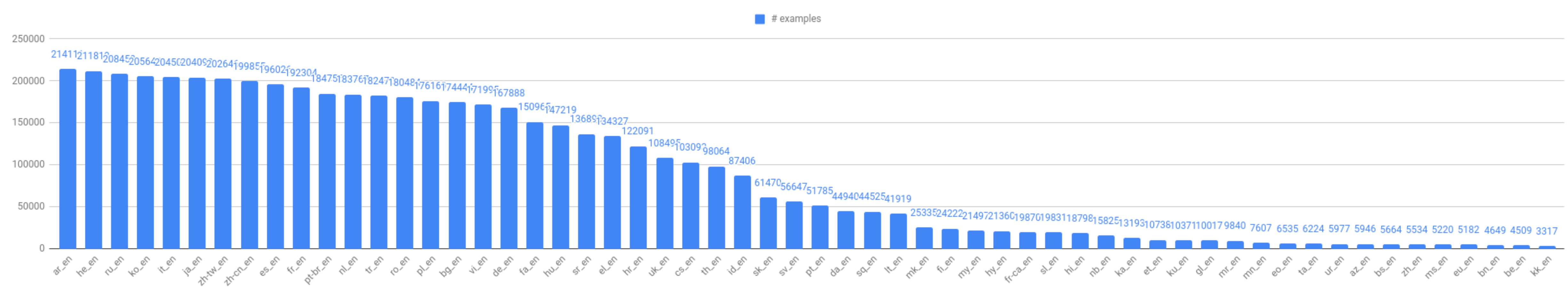
Experiments - Low Resource

- The TED talks dataset
  - 58 languages, to and from English



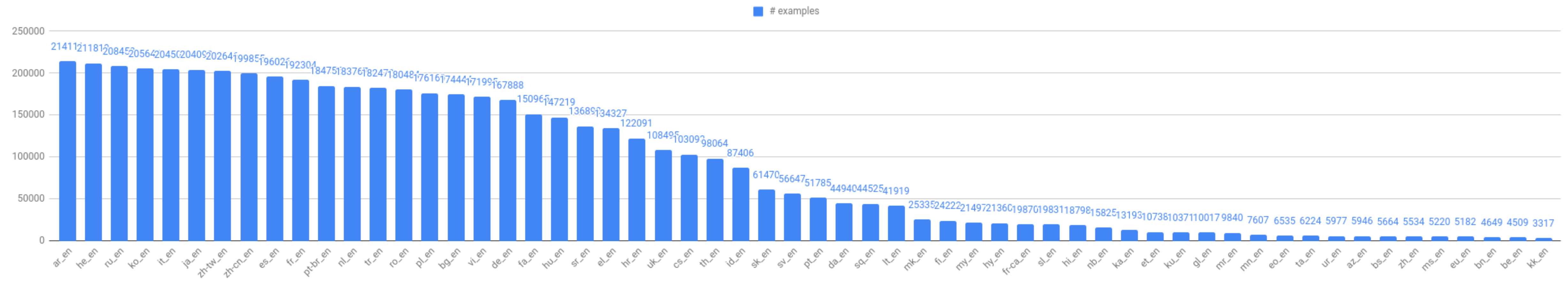
Experiments - Low Resource

- The TED talks dataset
  - 58 languages, to and from English
    - 3k-214k training examples per language - imbalanced



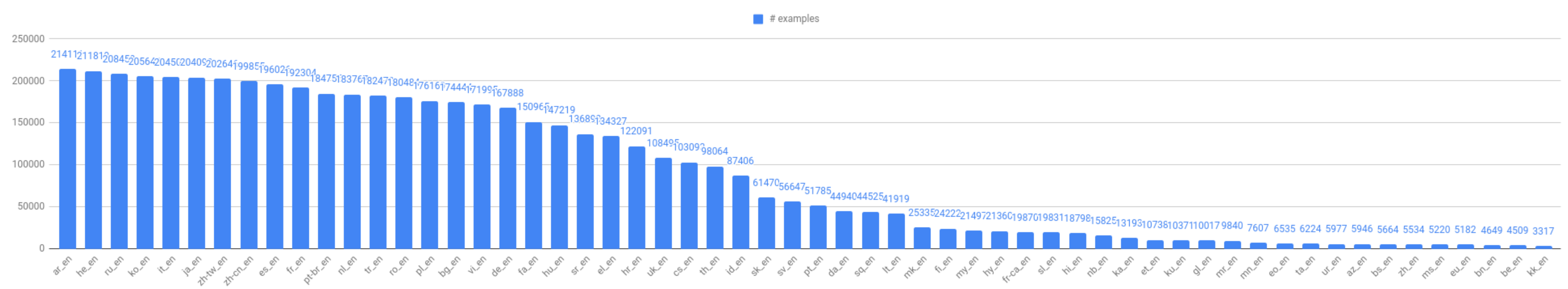
Experiments - Low Resource

- The TED talks dataset
    - 58 languages, to and from English
        - 3k-214k training examples per language - imbalanced
        - 258k original sentences in train set → mostly multi-way parallel



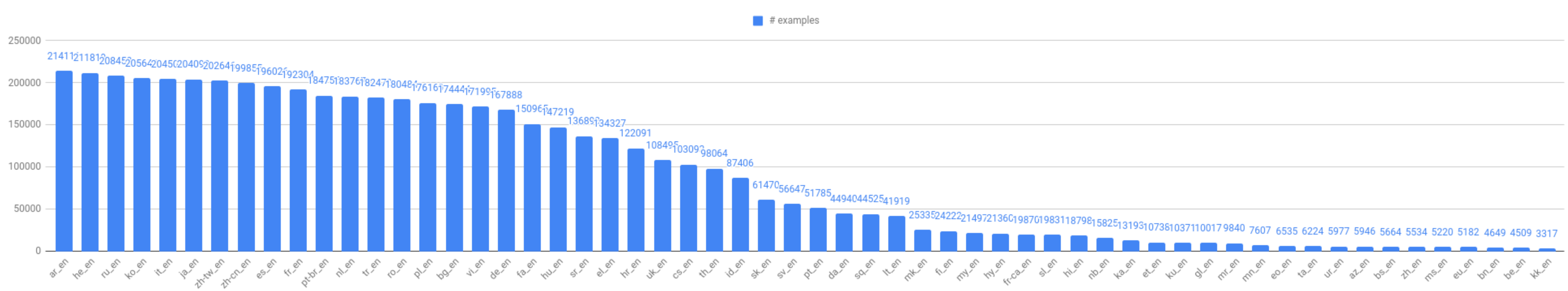
Experiments - Low Resource

- The TED talks dataset
    - 58 languages, to and from English
        - 3k-214k training examples per language - imbalanced
        - 258k original sentences in train set → mostly multi-way parallel
- Transformer-Base models, similar capacity (93M parameters)



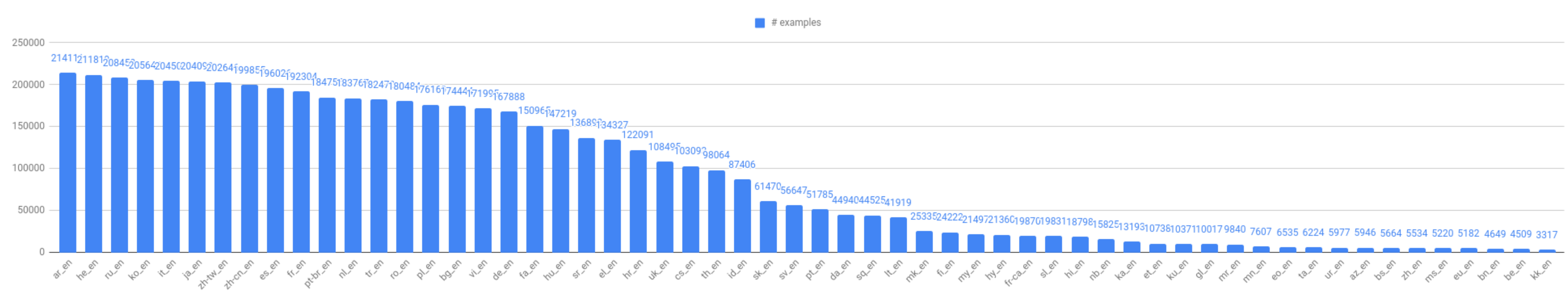
Experiments - Low Resource

- The TED talks dataset
    - 58 languages, to and from English
        - 3k-214k training examples per language - imbalanced
        - 258k original sentences in train set → mostly multi-way parallel
- Transformer-Base models, similar capacity (93M parameters)
    - Shared wordpiece vocabulary, 32k symbols



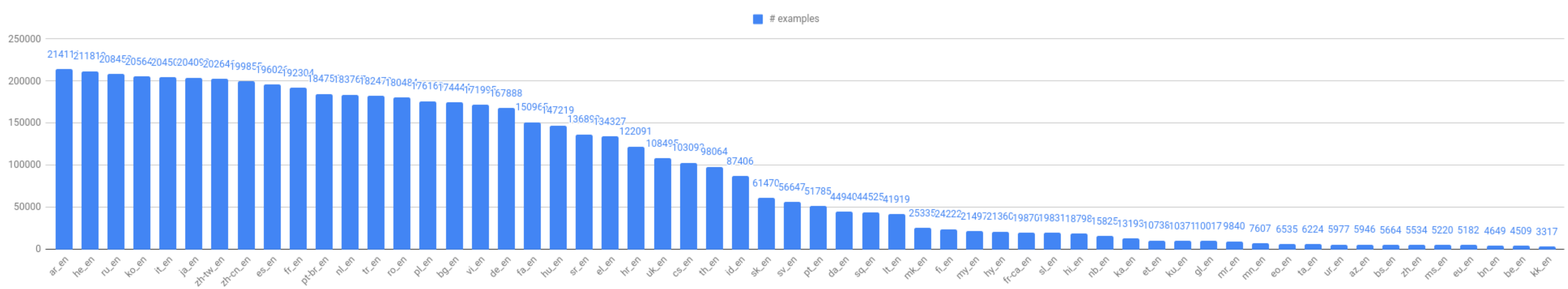
Experiments - Low Resource

- The TED talks dataset
  - 58 languages, to and from English
    - 3k-214k training examples per language - imbalanced
    - 258k original sentences in train set → mostly multi-way parallel
- Transformer-Base models, similar capacity (93M parameters)
  - Shared wordpiece vocabulary, 32k symbols
  - Many-to-Many (English-Centric), Many-to-One, One-to-Many, One-to-One



Experiments - Low Resource

- The TED talks dataset
  - 58 languages, to and from English
    - 3k-214k training examples per language - imbalanced
    - 258k original sentences in train set → mostly multi-way parallel
- Transformer-Base models, similar capacity (93M parameters)
  - Shared wordpiece vocabulary, 32k symbols
  - Many-to-Many (English-Centric), Many-to-One, One-to-Many, One-to-One
  - Joint Multilingual models



Results - Low Resource

Results - Low Resource

- Multilingual models significantly outperform baselines

	Az-En	Be-En	Gl-En	Sk-En	Avg.
# of examples	5.9k	4.5k	10k	61k	20.3k
Neubig & Hu 18					
baselines	2.7	2.8	16.2	24	11.42
many-to-one	11.7	18.3	29.1	28.3	21.85
Ours					
many-to-one	11.24	18.28	28.63	26.78	21.23
many-to-many	**12.78**	**21.73**	**30.65**	**29.54**	**23.67**

Results - Low Resource

- Multilingual models significantly outperform baselines
- Many-to-Many models outperform fine-tuned Many-to-One models

	Az-En	Be-En	Gl-En	Sk-En	Avg.
# of examples	5.9k	4.5k	10k	61k	20.3k
Neubig & Hu 18					
baselines	2.7	2.8	16.2	24	11.42
many-to-one	11.7	18.3	29.1	28.3	21.85
Ours					
many-to-one	11.24	18.28	28.63	26.78	21.23
many-to-many	**12.78**	**21.73**	**30.65**	**29.54**	**23.67**

Results - Low Resource

- Multilingual models significantly outperform baselines
- Many-to-Many models outperform fine-tuned Many-to-One models
- Similar result in language pairs with more data (baselines stronger here)

	Az-En	Be-En	Gl-En	Sk-En	Avg.
# of examples	5.9k	4.5k	10k	61k	20.3k
Neubig & Hu 18					
baselines	2.7	2.8	16.2	24	11.42
many-to-one	11.7	18.3	29.1	28.3	21.85
Ours					
many-to-one	11.24	18.28	28.63	26.78	21.23
many-to-many	**12.78**	**21.73**	**30.65**	**29.54**	**23.67**

	Ar-En	De-En	He-En	It-En	Avg.
# of examples	213k	167k	211k	203k	198.5k
baselines	27.84	30.5	**34.37**	33.64	31.59
many-to-one	25.93	28.87	30.19	32.42	29.35
many-to-many	**28.32**	**32.97**	33.18	**35.14**	**32.4**

Results - Low Resource

- Multilingual models significantly outperform baselines
- Many-to-Many models outperform fine-tuned Many-to-One models
- Similar result in language pairs with more data (baselines stronger here)
- Why? many-to-many is “harder”

	Az-En	Be-En	Gl-En	Sk-En	Avg.
# of examples	5.9k	4.5k	10k	61k	20.3k
Neubig & Hu 18					
baselines	2.7	2.8	16.2	24	11.42
many-to-one	11.7	18.3	29.1	28.3	21.85
Ours					
many-to-one	11.24	18.28	28.63	26.78	21.23
many-to-many	**12.78**	**21.73**	**30.65**	**29.54**	**23.67**

	Ar-En	De-En	He-En	It-En	Avg.
# of examples	213k	167k	211k	203k	198.5k
baselines	27.84	30.5	**34.37**	33.64	31.59
many-to-one	25.93	28.87	30.19	32.42	29.35
many-to-many	**28.32**	**32.97**	33.18	**35.14**	**32.4**

Multilinguality as a Regularizer

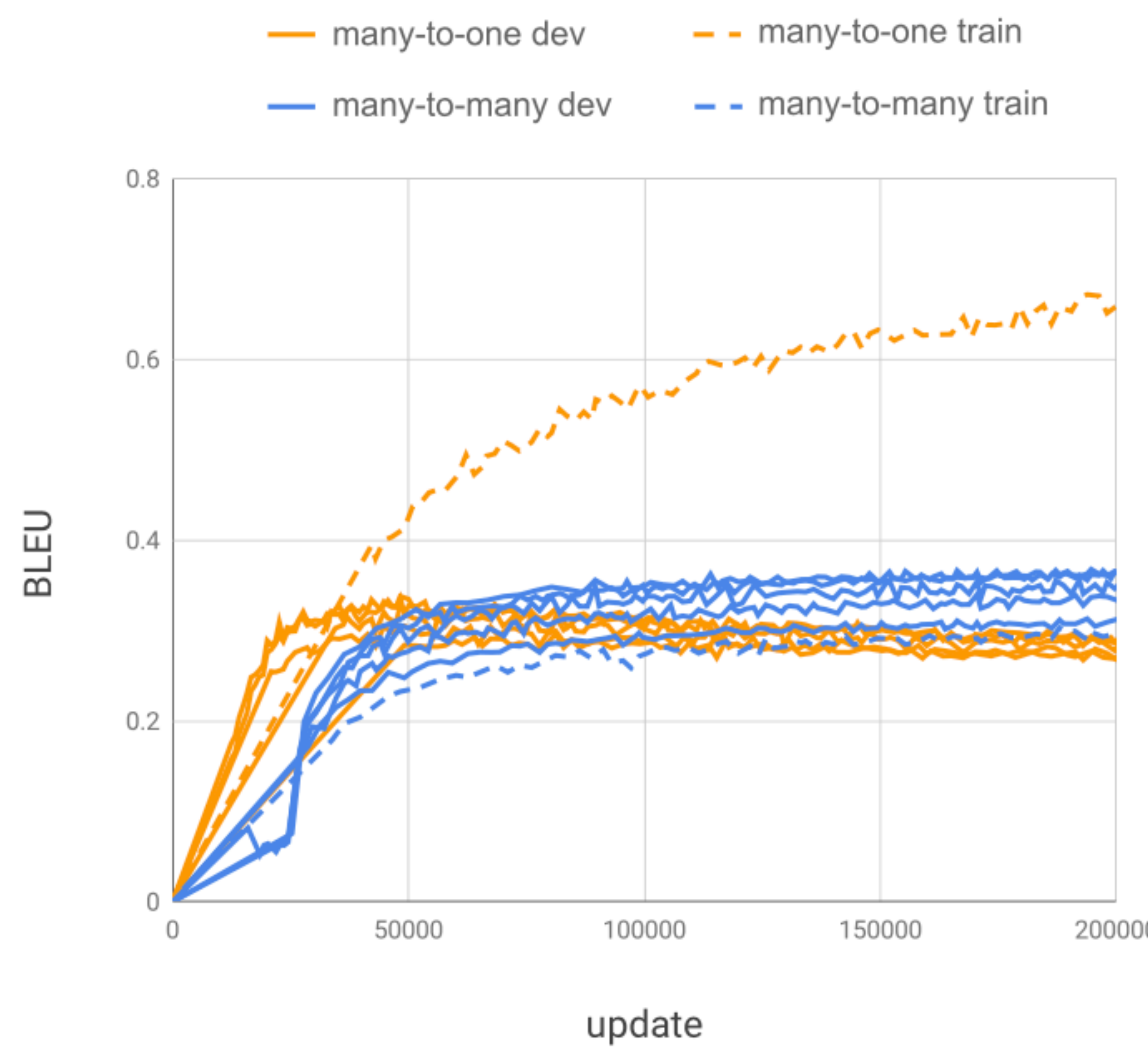
Multilinguality as a Regularizer

- The models we used are very large - prone to overfitting on the small datasets

Multilinguality as a Regularizer

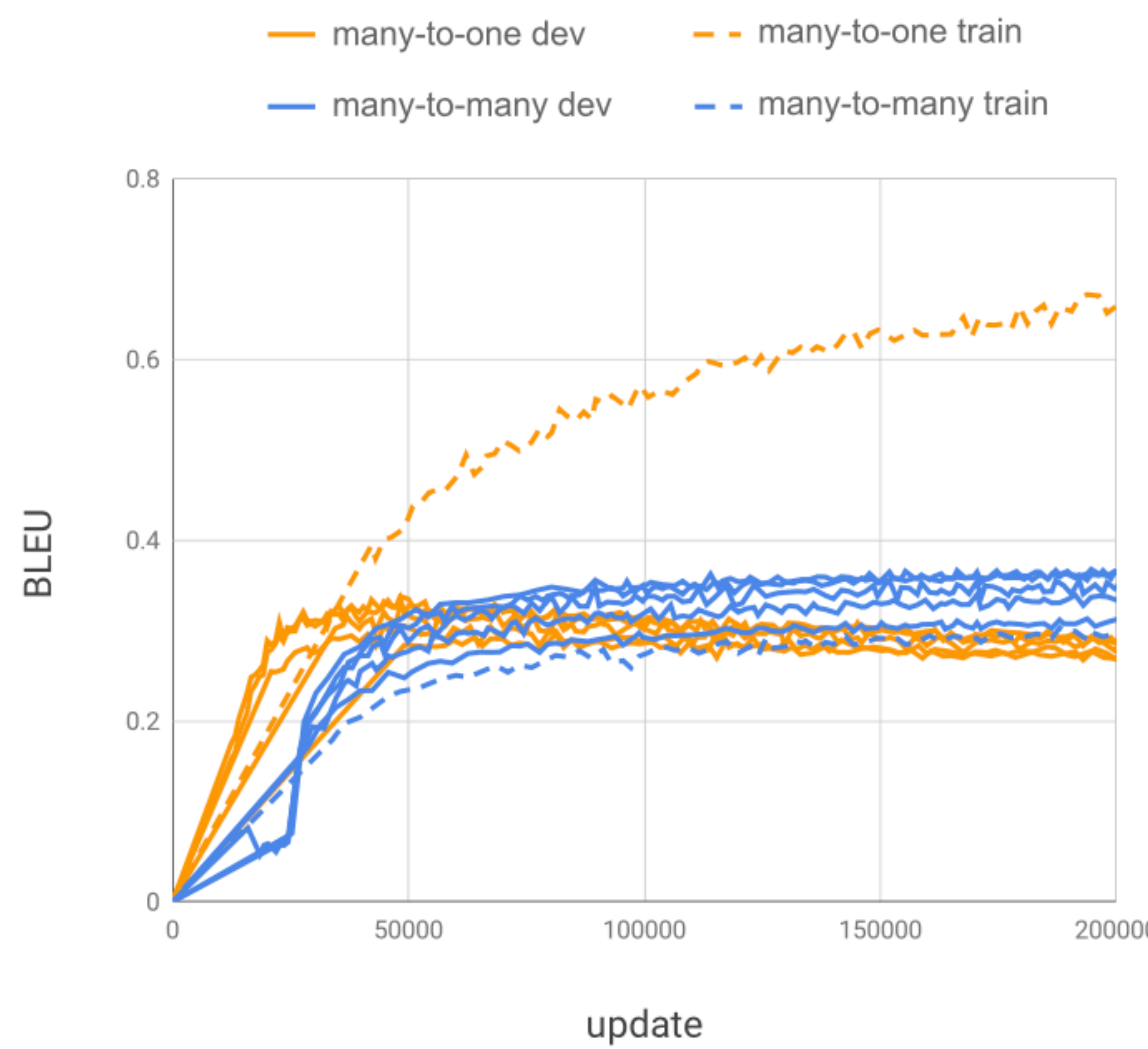
- The models we used are very large - prone to overfitting on the small datasets

- Having many target languages makes it harder to memorize, even with small data



Multilinguality as a Regularizer

- The models we used are very large - prone to overfitting on the small datasets
- Having many target languages makes it harder to memorize, even with small data
- Also easy to memorize since multi-way parallel



Evaluating out of English

Evaluating out of English

- One-to-Many outperform Many-to-Many and baselines

	En-Az	En-Be	En-Gl	En-Sk	Avg.
# of examples	5.9k	4.5k	10k	61k	20.3k
baselines	2.16	2.47	3.26	5.8	3.42
one-to-many	**5.06**	**10.72**	**26.59**	**24.52**	**16.72**
many-to-many	3.9	7.24	23.78	21.83	14.19

	En-Ar	En-De	En-He	En-It	Avg.
# of examples	213k	167k	211k	203k	198.5k
baselines	12.95	23.31	23.66	30.33	22.56
one-to-many	**16.67**	**30.54**	**27.62**	**35.89**	**27.68**
many-to-many	14.25	27.95	24.16	33.26	24.9

Evaluating out of English

- One-to-Many outperform Many-to-Many and baselines
- Many-to-Many models are biased towards English in the target

	En-Az	En-Be	En-Gl	En-Sk	Avg.
# of examples	5.9k	4.5k	10k	61k	20.3k
baselines	2.16	2.47	3.26	5.8	3.42
one-to-many	**5.06**	**10.72**	**26.59**	**24.52**	**16.72**
many-to-many	3.9	7.24	23.78	21.83	14.19

	En-Ar	En-De	En-He	En-It	Avg.
# of examples	213k	167k	211k	203k	198.5k
baselines	12.95	23.31	23.66	30.33	22.56
one-to-many	**16.67**	**30.54**	**27.62**	**35.89**	**27.68**
many-to-many	14.25	27.95	24.16	33.26	24.9

Evaluating out of English

- One-to-Many outperform Many-to-Many and baselines

- Many-to-Many models are biased towards English in the target

- When English memorization is not an issue, better to train on fewer directions

	En-Az	En-Be	En-Gl	En-Sk	Avg.
# of examples	5.9k	4.5k	10k	61k	20.3k
baselines	2.16	2.47	3.26	5.8	3.42
one-to-many	**5.06**	**10.72**	**26.59**	**24.52**	**16.72**
many-to-many	3.9	7.24	23.78	21.83	14.19

	En-Ar	En-De	En-He	En-It	Avg.
# of examples	213k	167k	211k	203k	198.5k
baselines	12.95	23.31	23.66	30.33	22.56
one-to-many	**16.67**	**30.54**	**27.62**	**35.89**	**27.68**
many-to-many	14.25	27.95	24.16	33.26	24.9

Experiments - High Resource

Experiments - High Resource

- We saw that:

Experiments - High Resource

- We saw that:
  - Massively multilingual many-to-many models win when going into-English (reduce memorization)

Experiments - High Resource

- We saw that:
  - Massively multilingual many-to-many models win when going into-English (reduce memorization)
  - One-to-many models are better when going out of English (not biased to English)

Experiments - High Resource

- We saw that:
    - Massively multilingual many-to-many models win when going into-English (reduce memorization)
    - One-to-many models are better when going out of English (not biased to English)
- Does this hold:

Experiments - High Resource

- We saw that:
  - Massively multilingual many-to-many models win when going into-English (reduce memorization)
  - One-to-many models are better when going out of English (not biased to English)
- Does this hold:
  - With even more languages?

Experiments - High Resource

- We saw that:
  - Massively multilingual many-to-many models win when going into-English (reduce memorization)
  - One-to-many models are better when going out of English (not biased to English)
- Does this hold:
  - With even more languages?
  - With larger, balanced, “real-world” datasets?

Experiments - High Resource

Experiments - High Resource

- Transformer Big(ger) models

Experiments - High Resource

- Transformer Big(ger) models
  - 473.7M parameters (vs. 213M in Big)

Experiments - High Resource

- Transformer Big(ger) models
  - 473.7M parameters (vs. 213M in Big)
  - Joint subword vocabulary with 64k symbols (24k unique characters)

Experiments - High Resource

- Transformer Big(ger) models
  - 473.7M parameters (vs. 213M in Big)
  - Joint subword vocabulary with 64k symbols (24k unique characters)
- In-house dataset

Experiments - High Resource

- Transformer Big(ger) models
  - 473.7M parameters (vs. 213M in Big)
  - Joint subword vocabulary with 64k symbols (24k unique characters)
- In-house dataset
  - English-Centric: 102 Languages to/from English (mirrored)

Experiments - High Resource

- Transformer Big(ger) models
  - 473.7M parameters (vs. 213M in Big)
  - Joint subword vocabulary with 64k symbols (24k unique characters)
- In-house dataset
  - English-Centric: 102 Languages to/from English (mirrored)
  - ~1M examples per language pair (balanced)

Experiments - High Resource

- Transformer Big(ger) models
  - 473.7M parameters (vs. 213M in Big)
  - Joint subword vocabulary with 64k symbols (24k unique characters)
- In-house dataset
  - English-Centric: 102 Languages to/from English (mirrored)
  - ~1M examples per language pair (balanced)
  - Not multi-way parallel

Results - Into English

Results - Into English

	Ar	Az	Be	De	He	It	Nl	Ro	Sk	Tr	Avg.
baselines	23.34	16.3	21.93	30.18	31.83	**36.47**	36.12	34.59	24.01	27.13	28.19
many-to-one	**26.04**	**23.68**	**25.36**	35.05	**33.61**	35.69	**36.28**	36.33	28.35	**29.75**	**31.01**
many-to-many	22.17	21.45	23.03	**37.06**	30.71	35.0	36.18	**36.57**	**29.87**	27.64	29.97

- Many-to-one model outperforms baselines and Many-to-Many

Results - Into English

	Ar	Az	Be	De	He	It	Nl	Ro	Sk	Tr	Avg.
baselines	23.34	16.3	21.93	30.18	31.83	**36.47**	36.12	34.59	24.01	27.13	28.19
many-to-one	**26.04**	**23.68**	**25.36**	35.05	**33.61**	35.69	**36.28**	36.33	28.35	**29.75**	**31.01**
many-to-many	22.17	21.45	23.03	**37.06**	30.71	35.0	36.18	**36.57**	**29.87**	27.64	29.97

- Many-to-one model outperforms baselines and Many-to-Many
    - When the data is large enough and not multi-way-parallel, memorization is not an issue and “less is more”

Results - Into English

	Ar	Az	Be	De	He	It	Nl	Ro	Sk	Tr	Avg.
baselines	23.34	16.3	21.93	30.18	31.83	**36.47**	36.12	34.59	24.01	27.13	28.19
many-to-one	**26.04**	**23.68**	**25.36**	35.05	**33.61**	35.69	**36.28**	36.33	28.35	**29.75**	**31.01**
many-to-many	22.17	21.45	23.03	**37.06**	30.71	35.0	36.18	**36.57**	**29.87**	27.64	29.97

- Many-to-one model outperforms baselines and Many-to-Many
    - When the data is large enough and not multi-way-parallel, memorization is not an issue and “less is more”
- German and Italian outliers - due to interference

Results - Into English

	Ar	Az	Be	De	He	It	Nl	Ro	Sk	Tr	Avg.
baselines	23.34	16.3	21.93	30.18	31.83	**36.47**	36.12	34.59	24.01	27.13	28.19
many-to-one	**26.04**	**23.68**	**25.36**	35.05	**33.61**	35.69	**36.28**	36.33	28.35	**29.75**	**31.01**
many-to-many	22.17	21.45	23.03	**37.06**	30.71	35.0	36.18	**36.57**	**29.87**	27.64	29.97

- Many-to-one model outperforms baselines and Many-to-Many
  - When the data is large enough and not multi-way-parallel, memorization is not an issue and “less is more”
- German and Italian outliers - due to interference
  - Many-to-one reached 38 BLEU when evaluated using German only dev-set, but degraded

Results - Out of English

Results - Out of English

	Ar	Az	Be	De	He	It	Nl	Ro	Sk	Tr	Avg.
baselines	10.57	8.07	15.3	23.24	19.47	31.42	28.68	27.92	11.08	15.54	19.13
one-to-many	**12.08**	**9.92**	**15.6**	**31.39**	**20.01**	**33**	**31.06**	**28.43**	**17.67**	**17.68**	**21.68**
many-to-many	10.57	9.84	14.3	28.48	17.91	30.39	29.67	26.23	18.15	15.58	20.11

- Clear advantage to the one-to-many model in all cases

Results - Out of English

	Ar	Az	Be	De	He	It	Nl	Ro	Sk	Tr	Avg.
baselines	10.57	8.07	15.3	23.24	19.47	31.42	28.68	27.92	11.08	15.54	19.13
one-to-many	**12.08**	**9.92**	**15.6**	**31.39**	**20.01**	**33**	**31.06**	**28.43**	**17.67**	**17.68**	**21.68**
many-to-many	10.57	9.84	14.3	28.48	17.91	30.39	29.67	26.23	18.15	15.58	20.11

- Clear advantage to the one-to-many model in all cases
- Up to 6-8 BLEU improvement over baseline (Slovak, German)

Results - Out of English

	Ar	Az	Be	De	He	It	Nl	Ro	Sk	Tr	Avg.
baselines	10.57	8.07	15.3	23.24	19.47	31.42	28.68	27.92	11.08	15.54	19.13
one-to-many	**12.08**	**9.92**	**15.6**	**31.39**	**20.01**	**33**	**31.06**	**28.43**	**17.67**	**17.68**	**21.68**
many-to-many	10.57	9.84	14.3	28.48	17.91	30.39	29.67	26.23	18.15	15.58	20.11

- Clear advantage to the one-to-many model in all cases
- Up to 6-8 BLEU improvement over baseline (Slovak, German)
- Less burden, not biased towards English

Analysis

Analysis

- The previous experiments present an extreme case (100+ languages in a single model)

Analysis

- The previous experiments present an extreme case (100+ languages in a single model)
- What is the trade-off between the number of languages and model performance?

Analysis

- The previous experiments present an extreme case (100+ languages in a single model)
- What is the trade-off between the number of languages and model performance?
    - Both supervised and Zero-Shot

Analysis

- The previous experiments present an extreme case (100+ languages in a single model)
- What is the trade-off between the number of languages and model performance?
  - Both supervised and Zero-Shot
- Keep model fixed, measure performance on 5 languages while varying the number of additional languages

Analysis - Supervised Directions

Analysis - Supervised Directions

	Ar-En	En-Ar	Fr-En	En-Fr	Ru-En	En-Ru	Uk-En	En-Uk	Avg.
5-to-5	**23.87**	**12.42**	**38.99**	**37.3**	29.07	**24.86**	**26.17**	16.48	**26.14**
25-to-25	23.43	11.77	38.87	36.79	**29.36**	23.24	25.81	**17.17**	25.8
50-to-50	23.7	11.65	37.81	35.83	29.22	21.95	26.02	15.32	25.18
75-to-75	22.23	10.69	37.97	34.35	28.55	20.7	25.89	14.59	24.37
103-to-103	21.16	10.25	35.91	34.42	27.25	19.9	24.53	13.89	23.41

Analysis - Supervised Directions

	Ar-En	En-Ar	Fr-En	En-Fr	Ru-En	En-Ru	Uk-En	En-Uk	Avg.
5-to-5	**23.87**	**12.42**	**38.99**	**37.3**	29.07	**24.86**	**26.17**	16.48	**26.14**
25-to-25	23.43	11.77	38.87	36.79	**29.36**	23.24	25.81	**17.17**	25.8
50-to-50	23.7	11.65	37.81	35.83	29.22	21.95	26.02	15.32	25.18
75-to-75	22.23	10.69	37.97	34.35	28.55	20.7	25.89	14.59	24.37
103-to-103	21.16	10.25	35.91	34.42	27.25	19.9	24.53	13.89	23.41

- Clear trade-off between number of languages and model accuracy

Analysis - Supervised Directions

	Ar-En	En-Ar	Fr-En	En-Fr	Ru-En	En-Ru	Uk-En	En-Uk	Avg.
5-to-5	**23.87**	**12.42**	**38.99**	**37.3**	29.07	**24.86**	**26.17**	16.48	**26.14**
25-to-25	23.43	11.77	38.87	36.79	**29.36**	23.24	25.81	**17.17**	25.8
50-to-50	23.7	11.65	37.81	35.83	29.22	21.95	26.02	15.32	25.18
75-to-75	22.23	10.69	37.97	34.35	28.55	20.7	25.89	14.59	24.37
103-to-103	21.16	10.25	35.91	34.42	27.25	19.9	24.53	13.89	23.41

- Clear trade-off between number of languages and model accuracy

- Maybe we need even bigger models? 1M examples per language pair is not very large… (in MT scale)

Analysis - Zero-Shot Directions

Analysis - Zero-Shot Directions

- 50-to-50 strikes a good balance between capacity and generalization

	Ar-Fr	Fr-Ar	Ru-Uk	Uk-Ru	Avg.
5-to-5	1.66	4.49	3.7	3.02	3.21
25-to-25	1.83	**5.52**	**16.67**	4.31	7.08
50-to-50	**4.34**	4.72	15.14	**20.23**	**11.1**
75-to-75	1.85	4.26	11.2	15.88	8.3
103-to-103	2.87	3.05	12.3	18.49	9.17

Analysis - Zero-Shot Directions

- 50-to-50 strikes a good balance between capacity and generalization

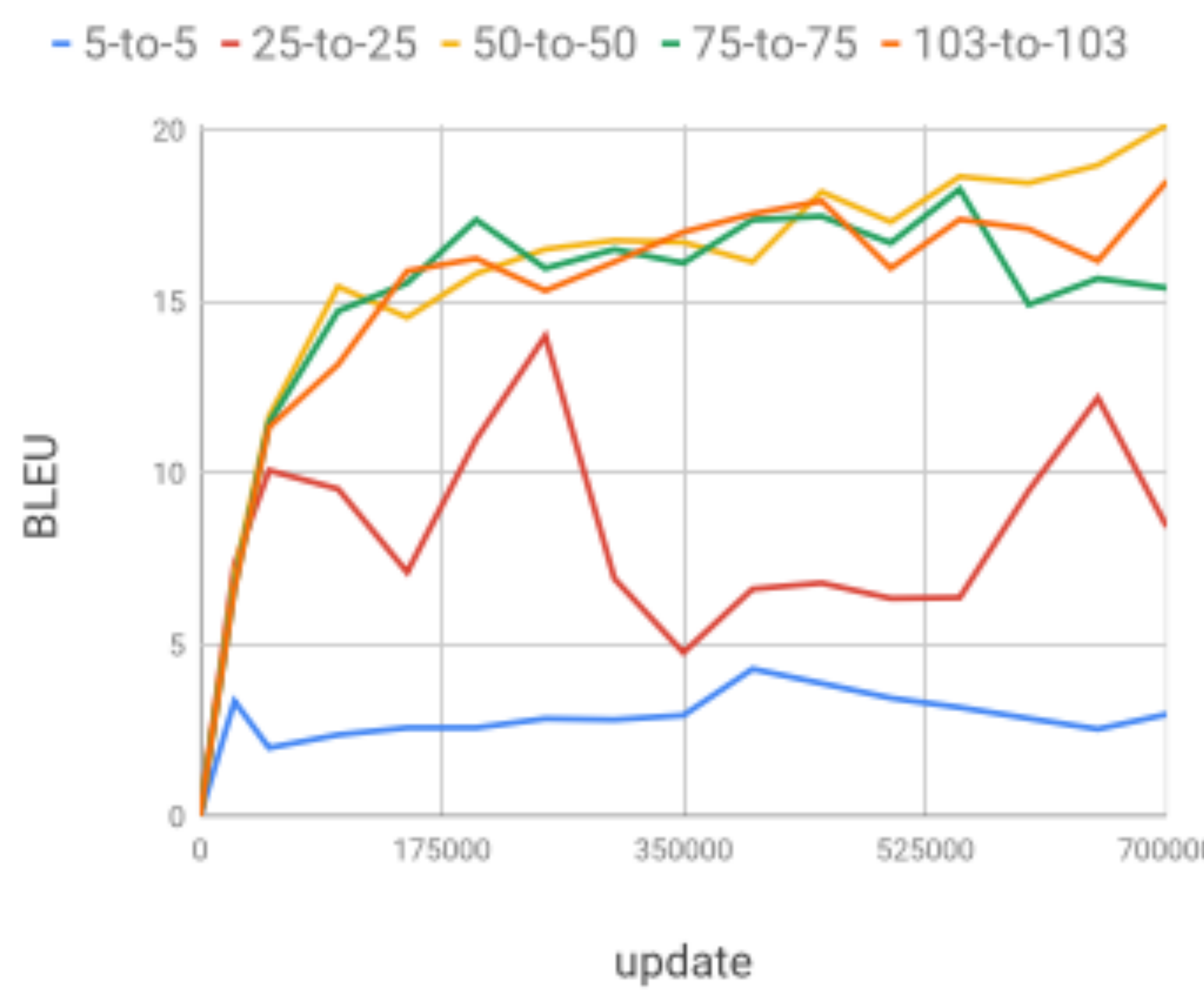
- Similar languages are much easier

	Ar-Fr	Fr-Ar	Ru-Uk	Uk-Ru	Avg.
5-to-5	1.66	4.49	3.7	3.02	3.21
25-to-25	1.83	**5.52**	**16.67**	4.31	7.08
50-to-50	**4.34**	4.72	15.14	**20.23**	**11.1**
75-to-75	1.85	4.26	11.2	15.88	8.3
103-to-103	2.87	3.05	12.3	18.49	9.17

Analysis - Zero-Shot Directions

- 50-to-50 strikes a good balance between capacity and generalization
- Similar languages are much easier
- General trend - more languages, more generalization (interlingua?)

	Ar-Fr	Fr-Ar	Ru-Uk	Uk-Ru	Avg.
5-to-5	1.66	4.49	3.7	3.02	3.21
25-to-25	1.83	**5.52**	**16.67**	4.31	7.08
50-to-50	**4.34**	4.72	15.14	**20.23**	**11.1**
75-to-75	1.85	4.26	11.2	15.88	8.3
103-to-103	2.87	3.05	12.3	18.49	9.17



Conclusions

Conclusions

- Massively multilingual NMT is possible!

Conclusions

- Massively multilingual NMT is possible!
- Especially helpful in low-resource settings

Conclusions

- Massively multilingual NMT is possible!
- Especially helpful in low-resource settings
- Can scale to high resource settings, 100+ languages (with some trade-off)

Conclusions

- Massively multilingual NMT is possible!
- Especially helpful in low-resource settings
- Can scale to high resource settings, 100+ languages (with some trade-off)
- Zero-shot analysis: more languages - more generalization?

Lots of Avenues for Future Work

Lots of Avenues for Future Work

- Improving zero-shot performance with interlingual-losses

Lots of Avenues for Future Work

- Improving zero-shot performance with interlingual-losses
    - Bring zero-shot performance on-par with bridging

Lots of Avenues for Future Work

- Improving zero-shot performance with interlingual-losses
  - Bring zero-shot performance on-par with bridging
- Smarter clustering of language pairs and data (typology, overlapping content)

Lots of Avenues for Future Work

- Improving zero-shot performance with interlingual-losses
  - Bring zero-shot performance on-par with bridging
- Smarter clustering of language pairs and data (typology, overlapping content)
- Methods to reduce interference

Lots of Avenues for Future Work

- Improving zero-shot performance with interlingual-losses
    - Bring zero-shot performance on-par with bridging
- Smarter clustering of language pairs and data (typology, overlapping content)
- Methods to reduce interference
    - Multilingual distillation

Lots of Avenues for Future Work

- Improving zero-shot performance with interlingual-losses
  - Bring zero-shot performance on-par with bridging
- Smarter clustering of language pairs and data (typology, overlapping content)
- Methods to reduce interference
  - Multilingual distillation
  - Clever parameter sharing schemes

Lots of Avenues for Future Work

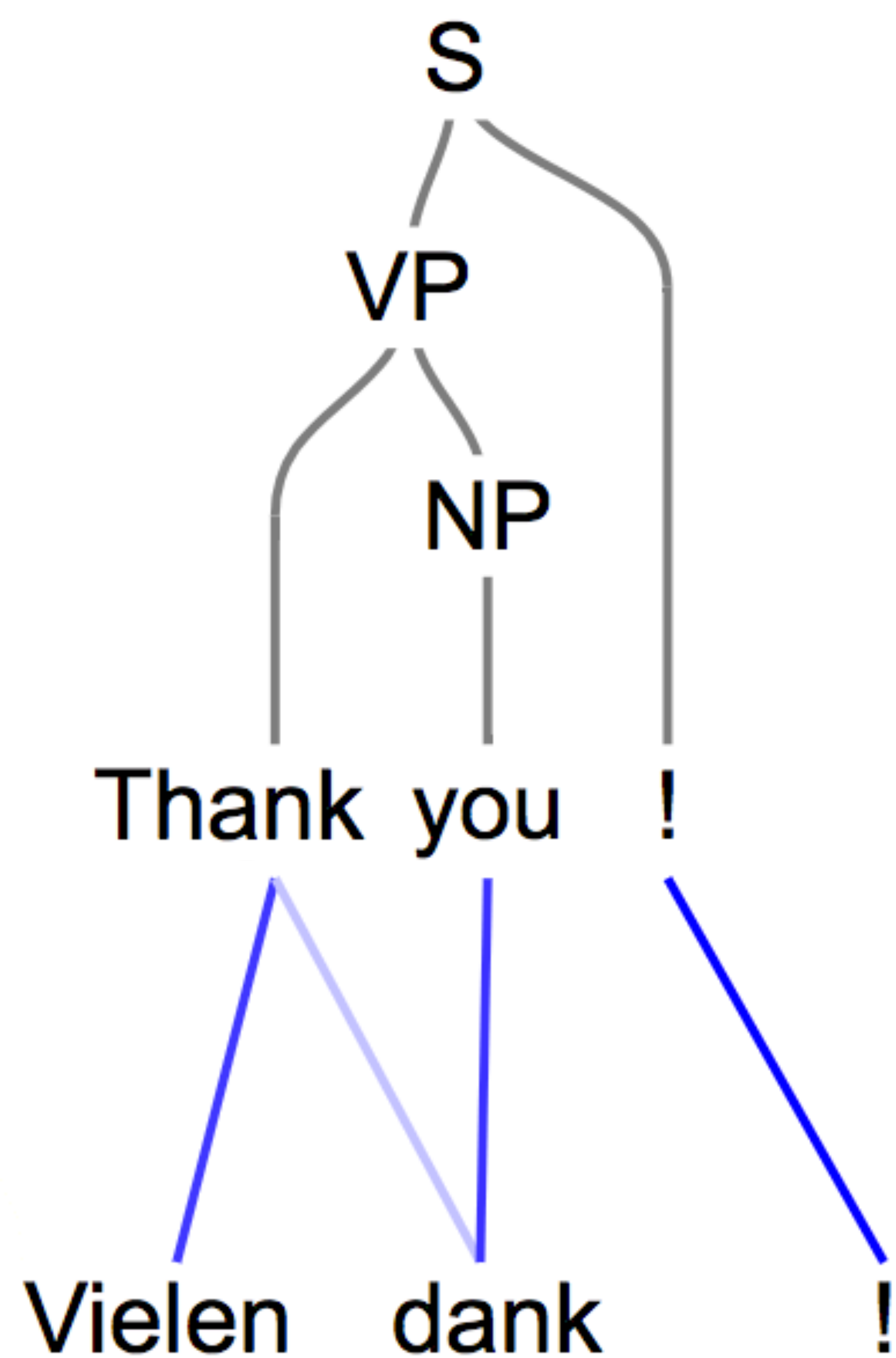
- Improving zero-shot performance with interlingual-losses
  - Bring zero-shot performance on-par with bridging
- Smarter clustering of language pairs and data (typology, overlapping content)
- Methods to reduce interference
  - Multilingual distillation
  - Clever parameter sharing schemes
- Massively multilingual NLP

Lots of Avenues for Future Work

- Improving zero-shot performance with interlingual-losses
  - Bring zero-shot performance on-par with bridging
- Smarter clustering of language pairs and data (typology, overlapping content)
- Methods to reduce interference
  - Multilingual distillation
  - Clever parameter sharing schemes
- Massively multilingual NLP
  - Multilingual BERT

Lots of Avenues for Future Work

- Improving zero-shot performance with interlingual-losses
  - Bring zero-shot performance on-par with bridging
- Smarter clustering of language pairs and data (typology, overlapping content)
- Methods to reduce interference
  - Multilingual distillation
  - Clever parameter sharing schemes
- Massively multilingual NLP
  - Multilingual BERT
  - Zero-shot Transfer Learning (Eriguchi et al 2018, Artetxe et al 2019)


S
VP
NP
Thank
you
!
Vielen
dank
!