Lecture 13: Machine Translation II

Alan Ritter

(many slides from Greg Durrett)

Neural MT Details

Encoder-Decoder MT

- Sutskever seq2seq paper: first major application of LSTMs to NLP

Sutskever et al. (2014)

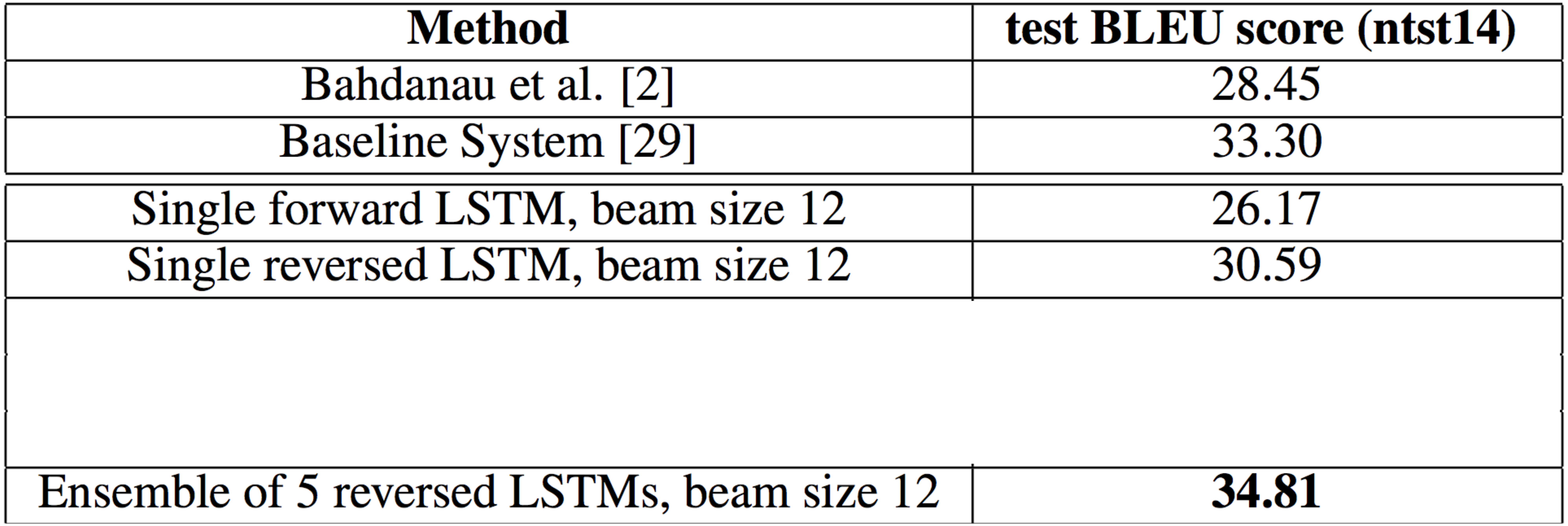
Encoder-Decoder MT

- Sutskever seq2seq paper: first major application of LSTMs to NLP
- Basic encoder-decoder with beam search

Sutskever et al. (2014)

Encoder-Decoder MT

- Sutskever seq2seq paper: first major application of LSTMs to NLP
- Basic encoder-decoder with beam search

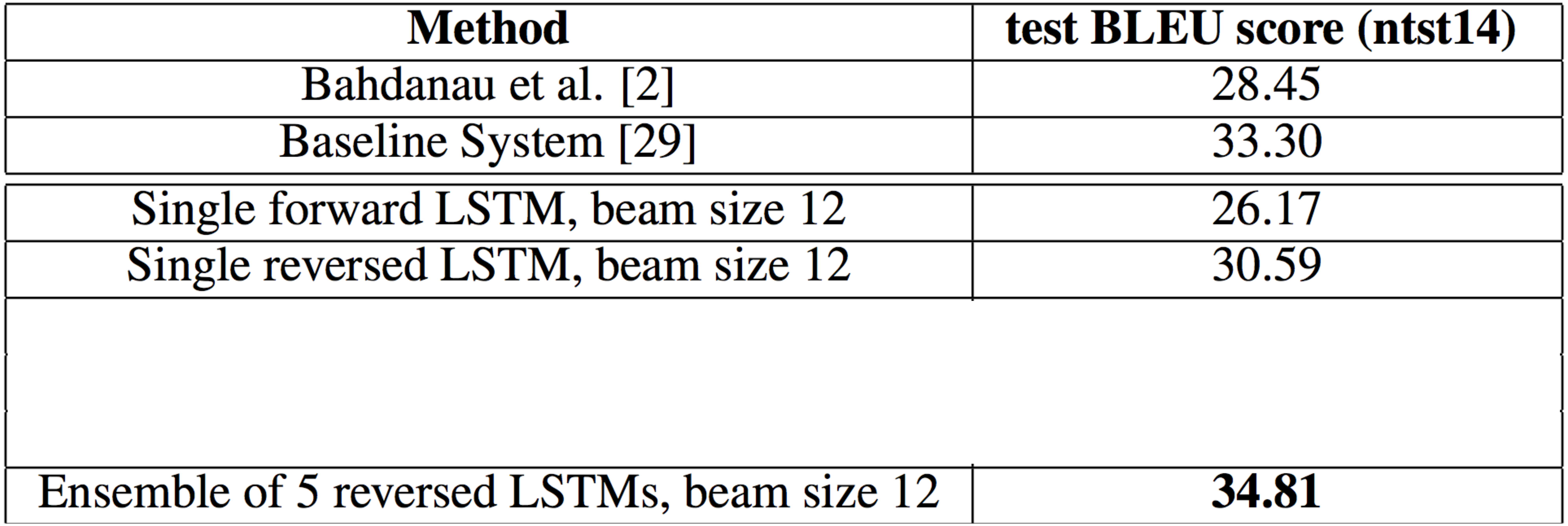


Method	**test BLEU score (ntst14)**
Bahdanau et al. [2]	28.45
Baseline System [29]	33.30
Single forward LSTM, beam size 12	26.17
Single reversed LSTM, beam size 12	30.59
Ensemble of 5 reversed LSTMs, beam size 12	**34.81**

Sutskever et al. (2014)

Encoder-Decoder MT

- Sutskever seq2seq paper: first major application of LSTMs to NLP
- Basic encoder-decoder with beam search



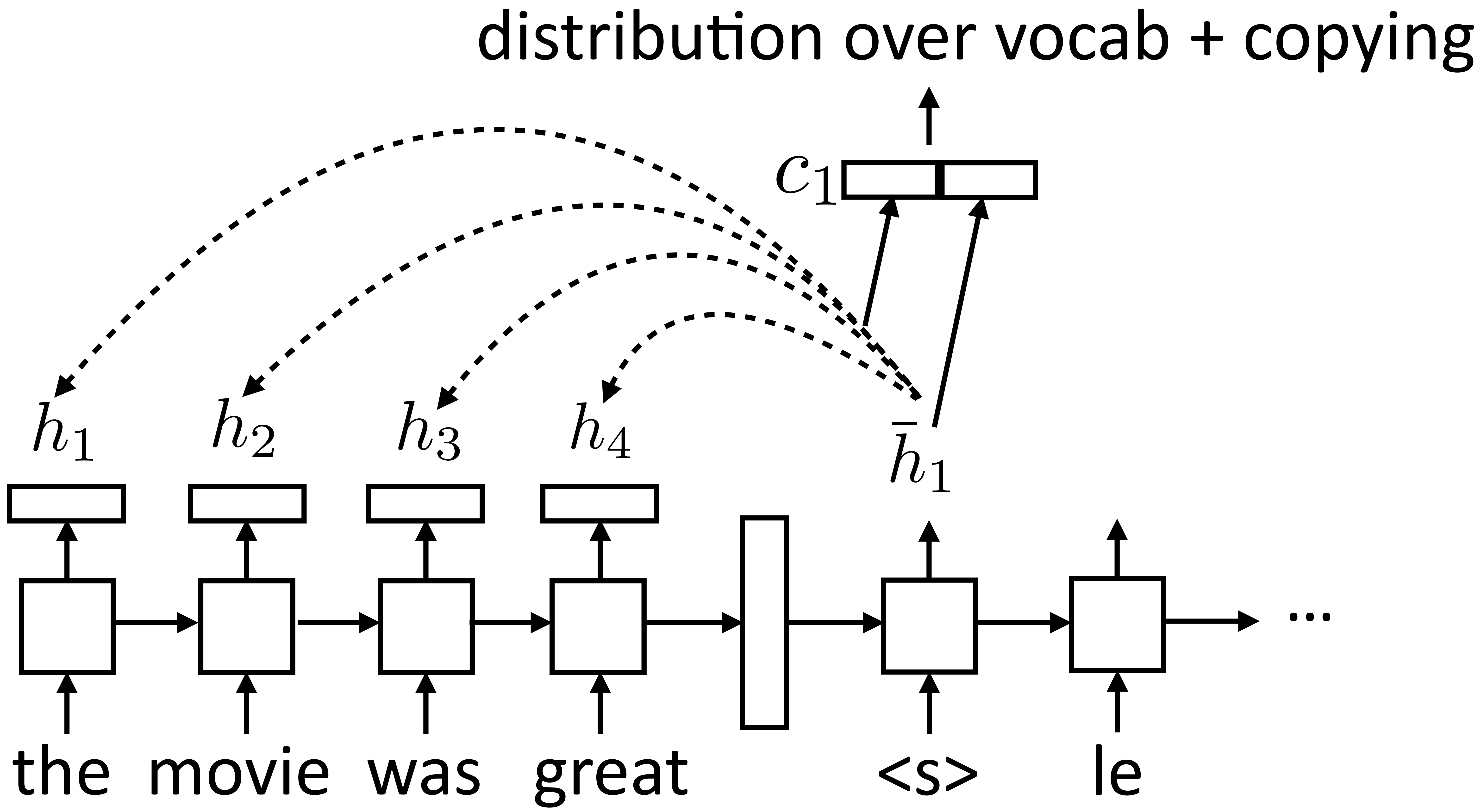
Method	test BLEU score (ntst14)
Bahdanau et al. [2]	28.45
Baseline System [29]	33.30
Single forward LSTM, beam size 12	26.17
Single reversed LSTM, beam size 12	30.59
Ensemble of 5 reversed LSTMs, beam size 12	**34.81**

- SOTA = 37.0 — not all that competitive…

Sutskever et al. (2014)

Encoder-Decoder MT

- Better model from seq2seq lectures: encoder-decoder with attention and copying for rare words



Results: WMT English-French

Results: WMT English-French

- 12M sentence pairs

Results: WMT English-French

- 12M sentence pairs

Classic phrase-based system: ~**33** BLEU, uses additional target-language data

Results: WMT English-French

- 12M sentence pairs

Classic phrase-based system: ~**33** BLEU, uses additional target-language data

Rerank with LSTMs: **36.5** BLEU (long line of work here; Devlin+ 2014)

Results: WMT English-French

- 12M sentence pairs

Classic phrase-based system: ~**33** BLEU, uses additional target-language data

Rerank with LSTMs: **36.5** BLEU (long line of work here; Devlin+ 2014)

Sutskever+ (2014) seq2seq single: **30.6** BLEU

Results: WMT English-French

- 12M sentence pairs

Classic phrase-based system: ~**33** BLEU, uses additional target-language data

Rerank with LSTMs: **36.5** BLEU (long line of work here; Devlin+ 2014)

Sutskever+ (2014) seq2seq single: **30.6** BLEU

Sutskever+ (2014) seq2seq ensemble: **34.8** BLEU

Results: WMT English-French

- 12M sentence pairs

Classic phrase-based system: ~**33** BLEU, uses additional target-language data

Rerank with LSTMs: **36.5** BLEU (long line of work here; Devlin+ 2014)

Sutskever+ (2014) seq2seq single: **30.6** BLEU

Sutskever+ (2014) seq2seq ensemble: **34.8** BLEU

Luong+ (2015) seq2seq ensemble with attention and rare word handling: **37.5** BLEU

- But English-French is a really easy language pair and there's *tons* of data for it! Does this approach work for anything harder?

Results: WMT English-German

- 4.5M sentence pairs

Classic phrase-based system: **20.7** BLEU

Luong+ (2014) seq2seq: **14** BLEU

Luong+ (2015) seq2seq ensemble with rare word handling: **23.0** BLEU

- Not nearly as good in absolute BLEU, but not really comparable across languages

Results: WMT English-German

- 4.5M sentence pairs

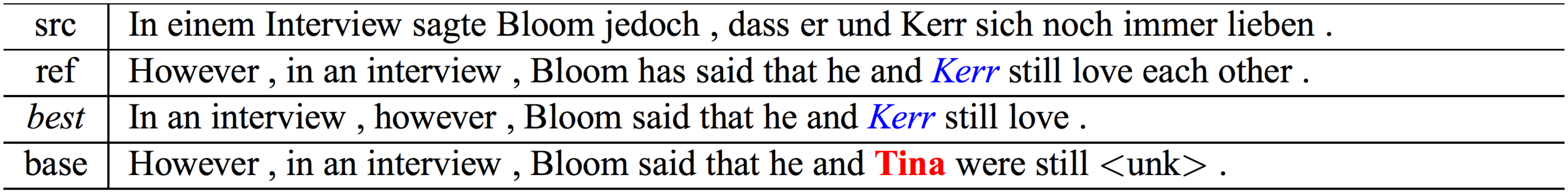
Classic phrase-based system: **20.7** BLEU

Luong+ (2014) seq2seq: **14** BLEU

Luong+ (2015) seq2seq ensemble with rare word handling: **23.0** BLEU

- Not nearly as good in absolute BLEU, but not really comparable across languages
- French, Spanish = easiest
  German, Czech = harder
  Japanese, Russian = hard (grammatically different, lots of morphology…)

MT Examples

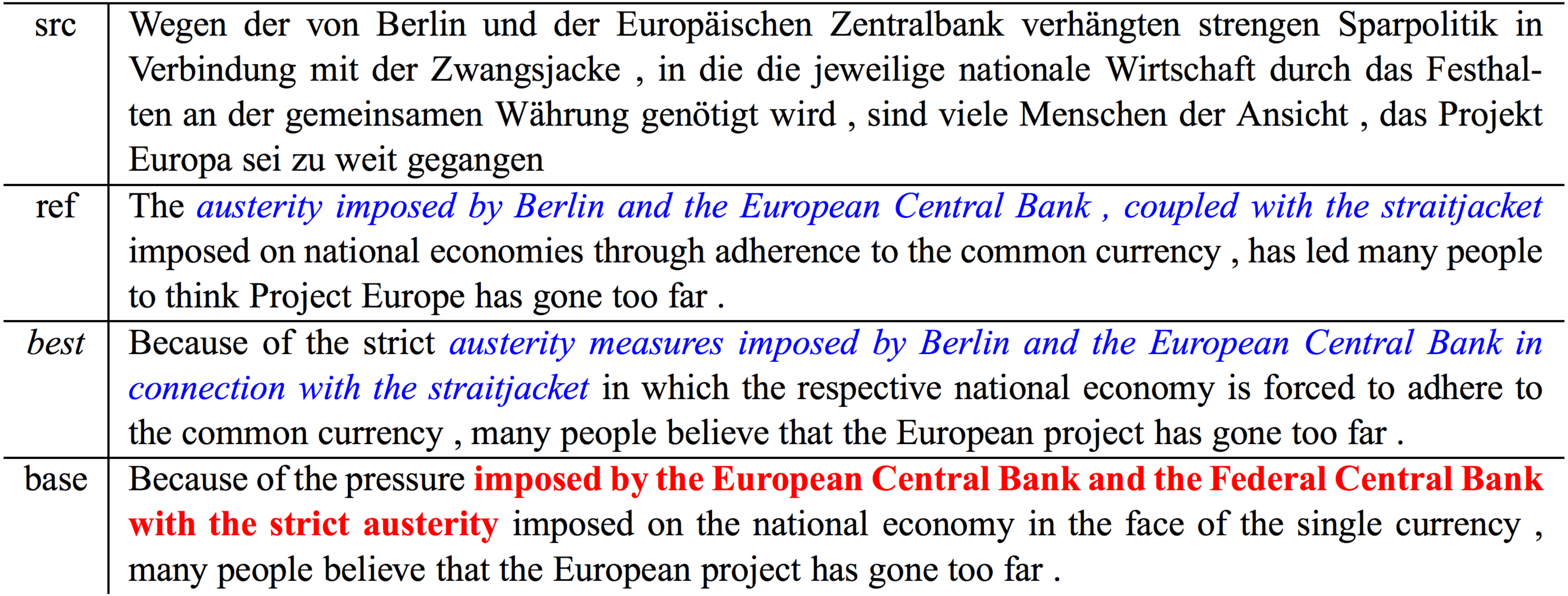


src	In einem Interview sagte Bloom jedoch , dass er und Kerr sich noch immer lieben .
ref	However , in an interview , Bloom has said that he and *Kerr* still love each other .
best	In an interview , however , Bloom said that he and *Kerr* still love .
base	However , in an interview , Bloom said that he and **Tina** were still <unk> .

- best = with attention, base = no attention
- NMT systems can hallucinate words, especially when not using attention — phrase-based doesn't do this

Luong et al. (2015)

MT Examples



src	Wegen der von Berlin und der Europäischen Zentralbank verhängten strengen Sparpolitik in Verbindung mit der Zwangsjacke , in die die jeweilige nationale Wirtschaft durch das Festhalten an der gemeinsamen Währung genötigt wird , sind viele Menschen der Ansicht , das Projekt Europa sei zu weit gegangen
ref	The *austerity imposed by Berlin and the European Central Bank , coupled with the straitjacket* imposed on national economies through adherence to the common currency , has led many people to think Project Europe has gone too far .
best	Because of the strict *austerity measures imposed by Berlin and the European Central Bank in connection with the straitjacket* in which the respective national economy is forced to adhere to the common currency , many people believe that the European project has gone too far .
base	Because of the pressure **imposed by the European Central Bank and the Federal Central Bank with the strict austerity** imposed on the national economy in the face of the single currency , many people believe that the European project has gone too far .

- best = with attention, base = no attention

Luong et al. (2015)

Backtranslation

- Classical MT methods used a bilingual corpus of sentences B = (S, T) and a large monolingual corpus T’ to train a language model. Can neural MT do the same?

Sennrich et al. (2015)

Backtranslation

- Classical MT methods used a bilingual corpus of sentences B = (S, T) and a large monolingual corpus T' to train a language model. Can neural MT do the same?
- Approach 1: force the system to generate T' as targets from null inputs

Sennrich et al. (2015)

Backtranslation

- Classical MT methods used a bilingual corpus of sentences B = (S, T) and a large monolingual corpus T' to train a language model. Can neural MT do the same?
- Approach 1: force the system to generate T' as targets from null inputs

s_1, t_1

s_2, t_2

…

[null], t'_1

[null], t'_2

…

Sennrich et al. (2015)

Backtranslation

- Classical MT methods used a bilingual corpus of sentences B = (S, T) and a large monolingual corpus T' to train a language model. Can neural MT do the same?

- Approach 1: force the system to generate T' as targets from null inputs

- Approach 2: generate synthetic sources with a T->S machine translation system (backtranslation)

s_1, t_1

s_2, t_2

…

[null], t'_1

[null], t'_2

…

Sennrich et al. (2015)

Backtranslation

- Classical MT methods used a bilingual corpus of sentences B = (S, T) and a large monolingual corpus T' to train a language model. Can neural MT do the same?

- Approach 1: force the system to generate T' as targets from null inputs

s_1, t_1

s_2, t_2

…

[null], t'_1

[null], t'_2

…

- Approach 2: generate synthetic sources with a T->S machine translation system (backtranslation)

s_1, t_1

s_2, t_2

…

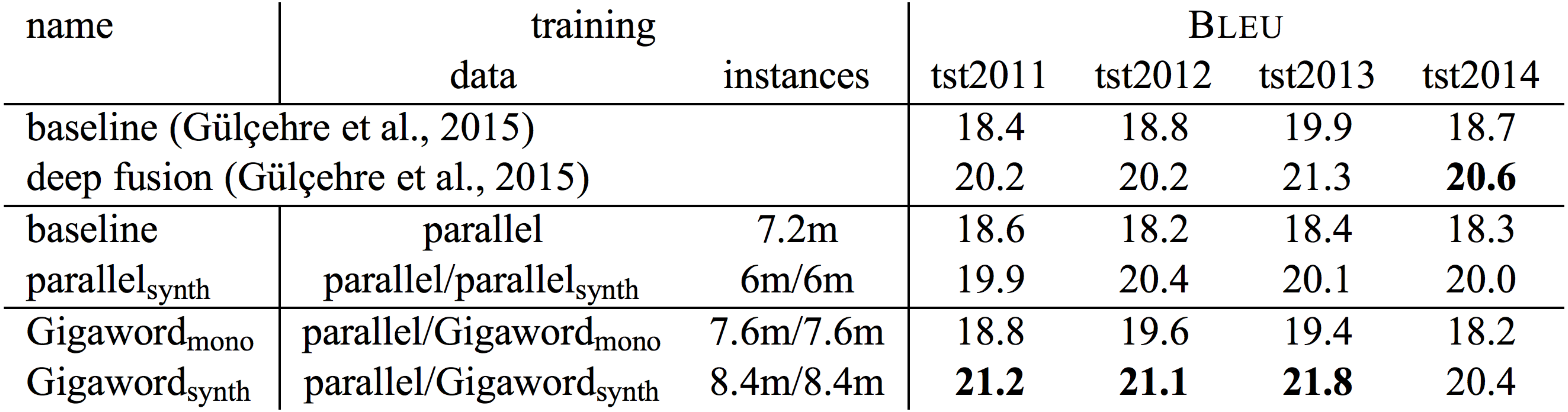
$MT(t'_1), t'_1$

$MT(t'_2), t'_2$

…

Sennrich et al. (2015)

Backtranslation



name	training		BLEU			
	data	instances	tst2011	tst2012	tst2013	tst2014
baseline (Gülçehre et al., 2015)			18.4	18.8	19.9	18.7
deep fusion (Gülçehre et al., 2015)			20.2	20.2	21.3	**20.6**
baseline	parallel	7.2m	18.6	18.2	18.4	18.3
parallel$_{synth}$	parallel/parallel$_{synth}$	6m/6m	19.9	20.4	20.1	20.0
Gigaword$_{mono}$	parallel/Gigaword$_{mono}$	7.6m/7.6m	18.8	19.6	19.4	18.2
Gigaword$_{synth}$	parallel/Gigaword$_{synth}$	8.4m/8.4m	**21.2**	**21.1**	**21.8**	20.4

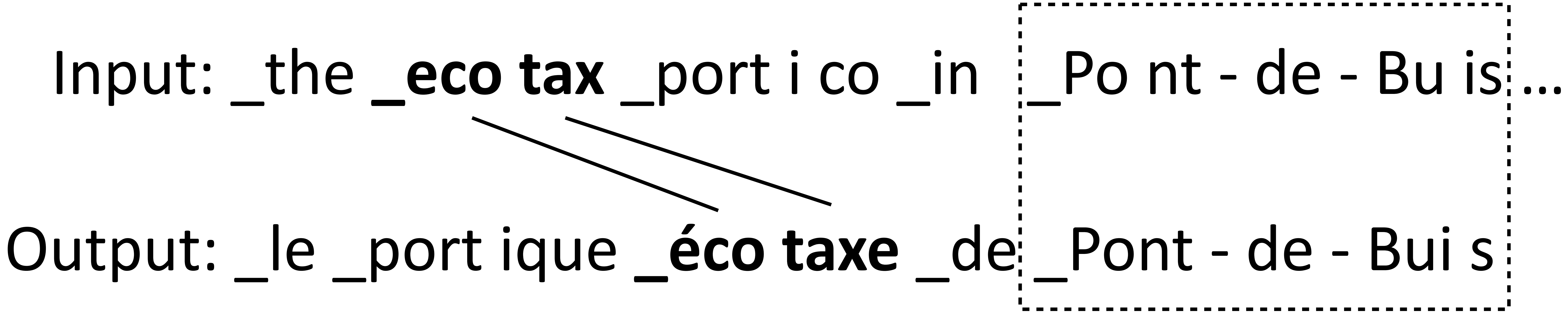
- Gigaword: large monolingual English corpus
- parallel$_{synth}$: backtranslate training data; makes additional noisy source sentences which could be useful

Sennrich et al. (2015)

Tokenization

Handling Rare Words

- Words are a difficult unit to work with: copying can be cumbersome, word vocabularies get very large
- Character-level models don't work well
- Compromise solution: use thousands of "word pieces" (which may be full words but may also be parts of words)



Input: _the **_eco tax** _port i co _in _Po nt - de - Bu is ...

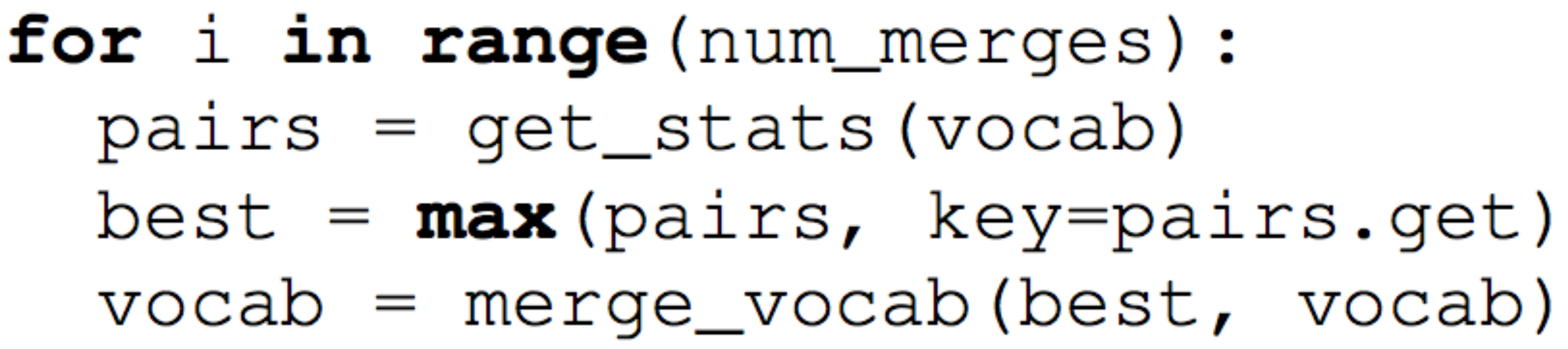
Output: _le _port ique **_éco taxe** _de _Pont - de - Bui s

- Can achieve transliteration with this, subword structure makes some translations easier to achieve

Sennrich et al. (2016)

Byte Pair Encoding (BPE)

- Start with every individual byte (basically character) as its own symbol



```
for i in range(num_merges):
  pairs = get_stats(vocab)
  best = max(pairs, key=pairs.get)
  vocab = merge_vocab(best, vocab)
```

- Count bigram character cooccurrences in dictionary
- Merge the most frequent pair of adjacent characters

- Vocabulary stats are weighted over a large corpus
- Doing 30k merges => vocabulary of around 30,000 word pieces. Includes many whole words

  and there were no re_ fueling stations anywhere

  one of the city ’s more un_ princi_ pled real estate agents

Sennrich et al. (2016)

Word Pieces

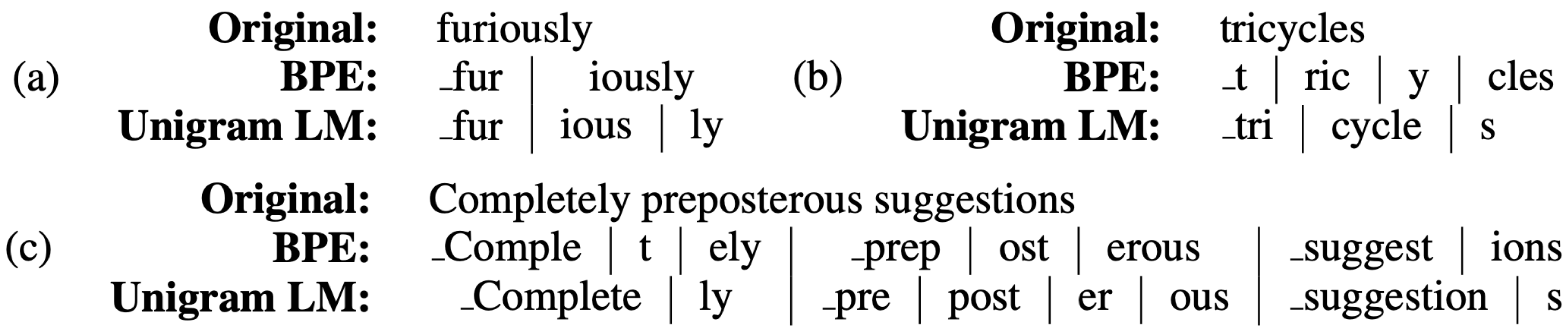
- Alternative to BPE

```
while voc size < target voc size:
    Build a language model over your corpus
    Merge pieces that lead to highest improvement in language model perplexity
```

- Issues: what LM to use? How to make this tractable?
- SentencePiece library from Google: unigram LM
- Result: way of segmenting input appropriate for translation

Schuster and Nakajima (2012), Wu et al. (2016), Kudo and Richardson (2018)

Comparison



(a)
Original: furiously
BPE: _fur | iously
Unigram LM: _fur | ious | ly

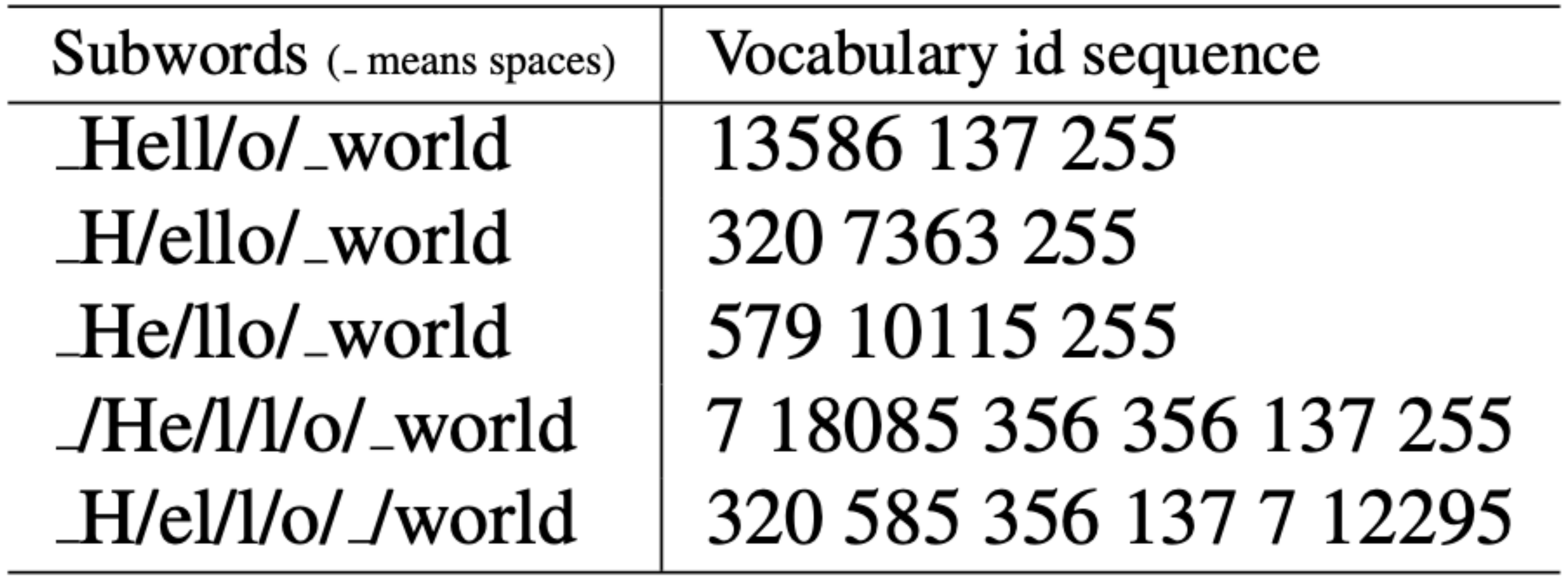
(b)
Original: tricycles
BPE: _t | ric | y | cles
Unigram LM: _tri | cycle | s

(c)
Original: Completely preposterous suggestions
BPE: _Comple | t | ely | _prep | ost | erous | _suggest | ions
Unigram LM: _Complete | ly | _pre | post | er | ous | _suggestion | s

- BPE produces less linguistically plausible units than word pieces (unigram LM)
- Some evidence that unigram LM works better in pre-trained transformer models

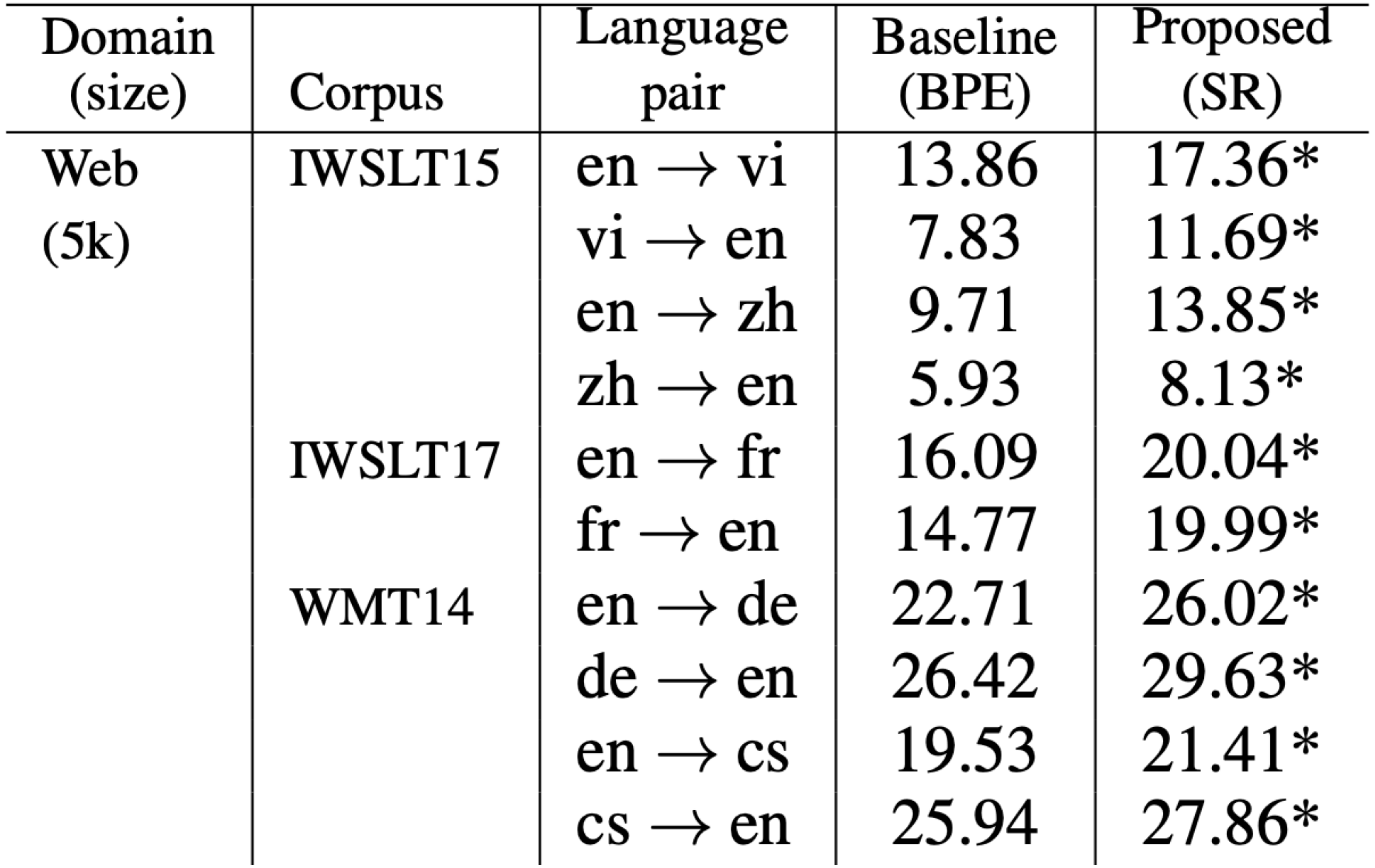
Bostrom and Durrett (2020)

Subword Regularization



Subwords (_ means spaces)	Vocabulary id sequence
_Hell/o/_world	13586 137 255
_H/ello/_world	320 7363 255
_He/llo/_world	579 10115 255
_/He/l/l/o/_world	7 18085 356 356 137 255
_H/el/l/o/_/world	320 585 356 137 7 12295

- Change subword sampling on-the-fly during training



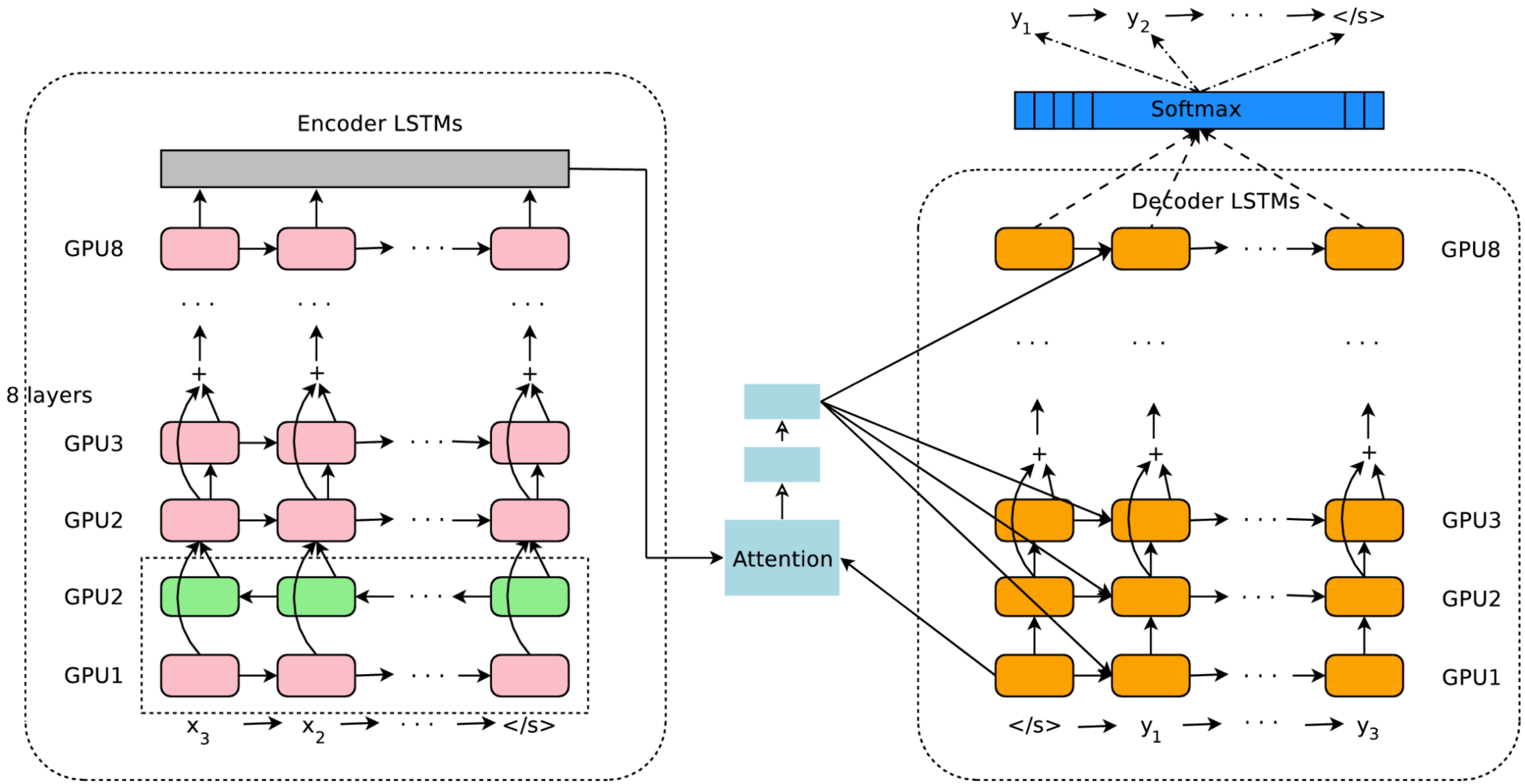
Domain (size)	Corpus	Language pair	Baseline (BPE)	Proposed (SR)
Web (5k)	IWSLT15	en → vi	13.86	17.36*
		vi → en	7.83	11.69*
		en → zh	9.71	13.85*
		zh → en	5.93	8.13*
	IWSLT17	en → fr	16.09	20.04*
		fr → en	14.77	19.99*
	WMT14	en → de	22.71	26.02*
		de → en	26.42	29.63*
		en → cs	19.53	21.41*
		cs → en	25.94	27.86*

- Subword regularization (SR) improves results over a static scheme (BPE)

Kudo (2018)

Google NMT

Google's NMT System



- 8-layer LSTM encoder-decoder with attention, word piece vocabulary of 8k-32k

Wu et al. (2016)

Google’s NMT System

Wu et al. (2016)

Google’s NMT System

English-French:

Google’s phrase-based system: 37.0 BLEU

Luong+ (2015) seq2seq ensemble with rare word handling: 37.5 BLEU

Google’s 32k word pieces: 38.95 BLEU

Wu et al. (2016)

Google’s NMT System

English-French:

Google’s phrase-based system: 37.0 BLEU

Luong+ (2015) seq2seq ensemble with rare word handling: 37.5 BLEU

Google’s 32k word pieces: 38.95 BLEU

English-German:

Google’s phrase-based system: 20.7 BLEU

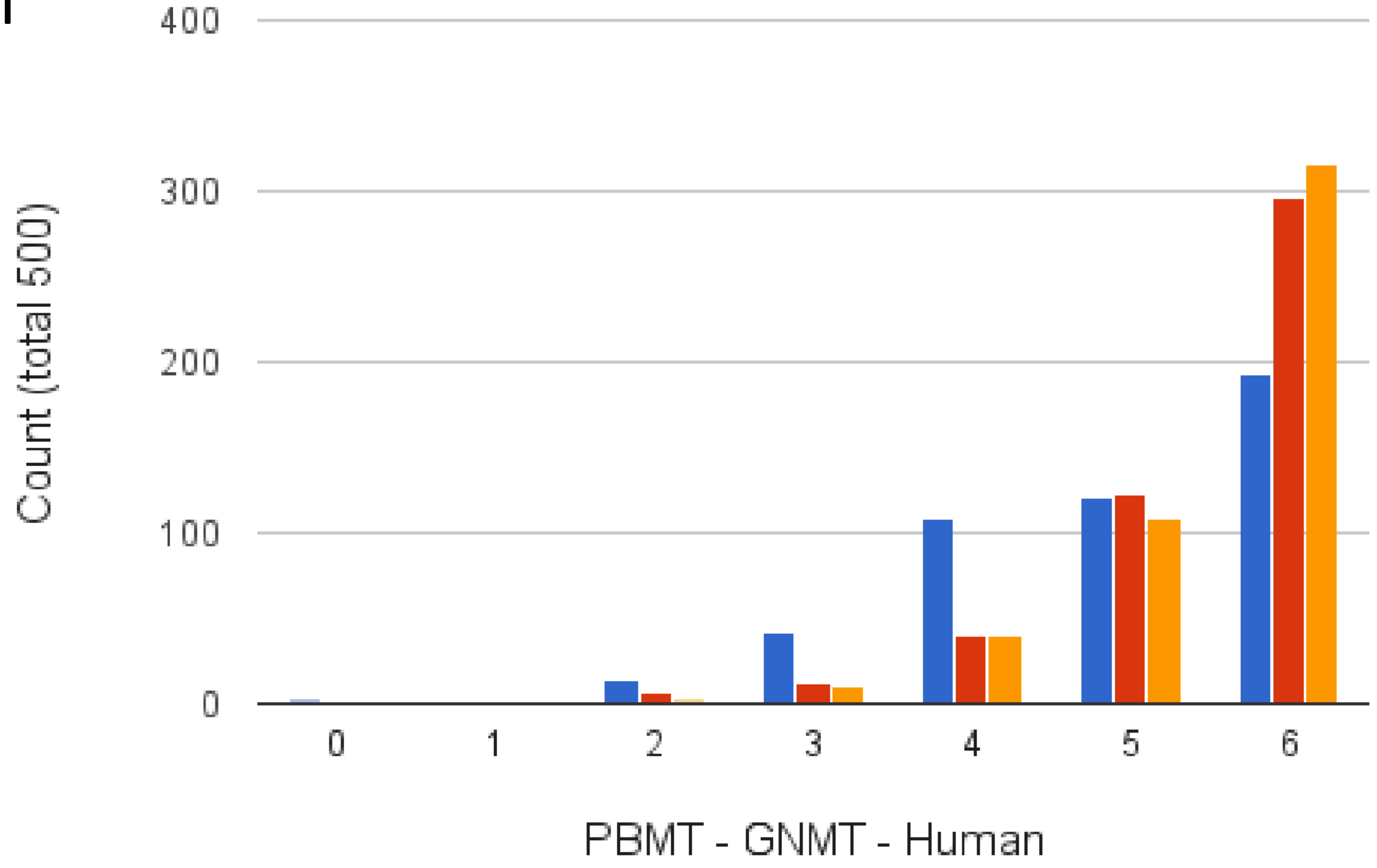
Luong+ (2015) seq2seq ensemble with rare word handling: 23.0 BLEU

Google’s 32k word pieces: 24.2 BLEU

Wu et al. (2016)

Human Evaluation (En-Es)

- Similar to human-level performance *on English-Spanish*



Wu et al. (2016)

Google’s NMT System

Source	She was spotted three days later by a dog walker trapped in the quarry	
PBMT	Elle a été repéré trois jours plus tard par un promeneur de chien piégé dans la carrière	6.0
GNMT	Elle a été repérée trois jours plus tard par un traîneau à chiens piégé dans la carrière.	2.0
Human	Elle a été repérée trois jours plus tard par une personne qui promenait son chien coincée dans la carrière	5.0

Wu et al. (2016)

Google’s NMT System

Source	She was spotted three days later by a dog walker trapped in the quarry	
PBMT	Elle a été repéré trois jours plus tard par un promeneur de chien piégé dans la carrière	6.0
GNMT	Elle a été repérée trois jours plus tard par un traîneau à chiens piégé dans la carrière.	2.0
Human	Elle a été repérée trois jours plus tard par une personne qui promenait son chien coincée dans la carrière	5.0

Gender is correct in GNMT
but not in PBMT

Wu et al. (2016)

Google’s NMT System

Source	She was spotted three days later by a dog walker trapped in the quarry	
PBMT	Elle a été repéré trois jours plus tard par un promeneur de chien piégé dans la carrière	6.0
GNMT	Elle a été repérée trois jours plus tard par un traîneau à chiens piégé dans la carrière.	2.0
Human	Elle a été repérée trois jours plus tard par une personne qui promenait son chien coincée dans la carrière	5.0

Gender is correct in GNMT but not in PBMT

“sled”

Wu et al. (2016)

Google’s NMT System

Source	She was spotted three days later by a dog walker trapped in the quarry	
PBMT	Elle a été repéré trois jours plus tard par un promeneur de chien piégé dans la carrière	6.0
GNMT	Elle a été repérée trois jours plus tard par un traîneau à chiens piégé dans la carrière.	2.0
Human	Elle a été repérée trois jours plus tard par une personne qui promenait son chien coincée dans la carrière	5.0

Gender is correct in GNMT but not in PBMT

“sled”

Wu et al. (2016)

Google’s NMT System

Source	She was spotted three days later by a dog walker trapped in the quarry	
PBMT	Elle a été repéré trois jours plus tard par un promeneur de chien piégé dans la carrière	6.0
GNMT	Elle a été repérée trois jours plus tard par un traîneau à chiens piégé dans la carrière.	2.0
Human	Elle a été repérée trois jours plus tard par une personne qui promenait son chien coincée dans la carrière	5.0

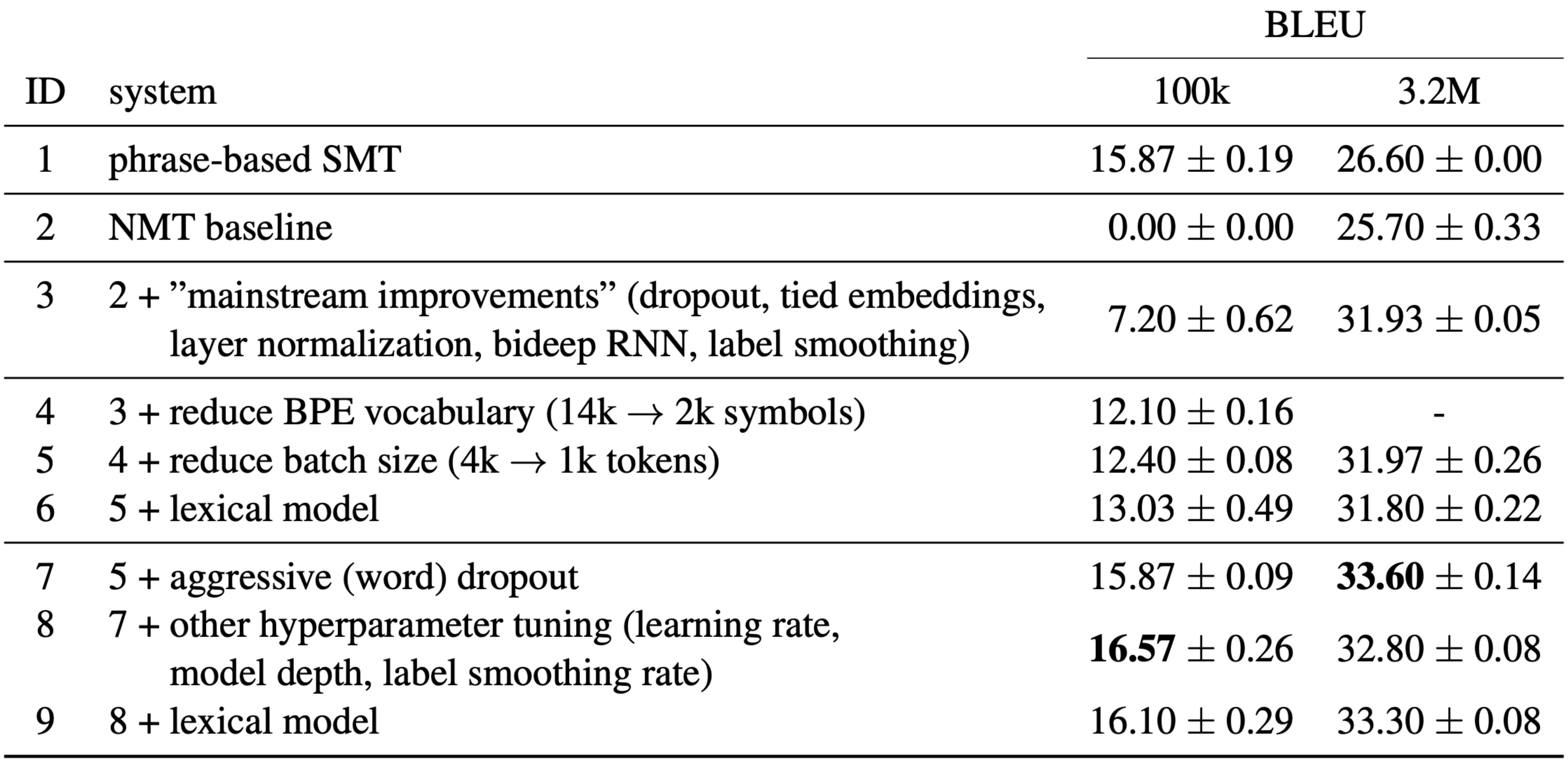
Gender is correct in GNMT but not in PBMT

“sled”

“walker”

Wu et al. (2016)

Frontiers in MT: Small Data



		BLEU	
ID	system	100k	3.2M
1	phrase-based SMT	15.87 ± 0.19	26.60 ± 0.00
2	NMT baseline	0.00 ± 0.00	25.70 ± 0.33
3	2 + ”mainstream improvements” (dropout, tied embeddings, layer normalization, bideep RNN, label smoothing)	7.20 ± 0.62	31.93 ± 0.05
4	3 + reduce BPE vocabulary (14k → 2k symbols)	12.10 ± 0.16	-
5	4 + reduce batch size (4k → 1k tokens)	12.40 ± 0.08	31.97 ± 0.26
6	5 + lexical model	13.03 ± 0.49	31.80 ± 0.22
7	5 + aggressive (word) dropout	15.87 ± 0.09	**33.60** ± 0.14
8	7 + other hyperparameter tuning (learning rate, model depth, label smoothing rate)	**16.57** ± 0.26	32.80 ± 0.08
9	8 + lexical model	16.10 ± 0.29	33.30 ± 0.08

- Synthetic small data setting: German -> English

Sennrich and Zhang (2019)

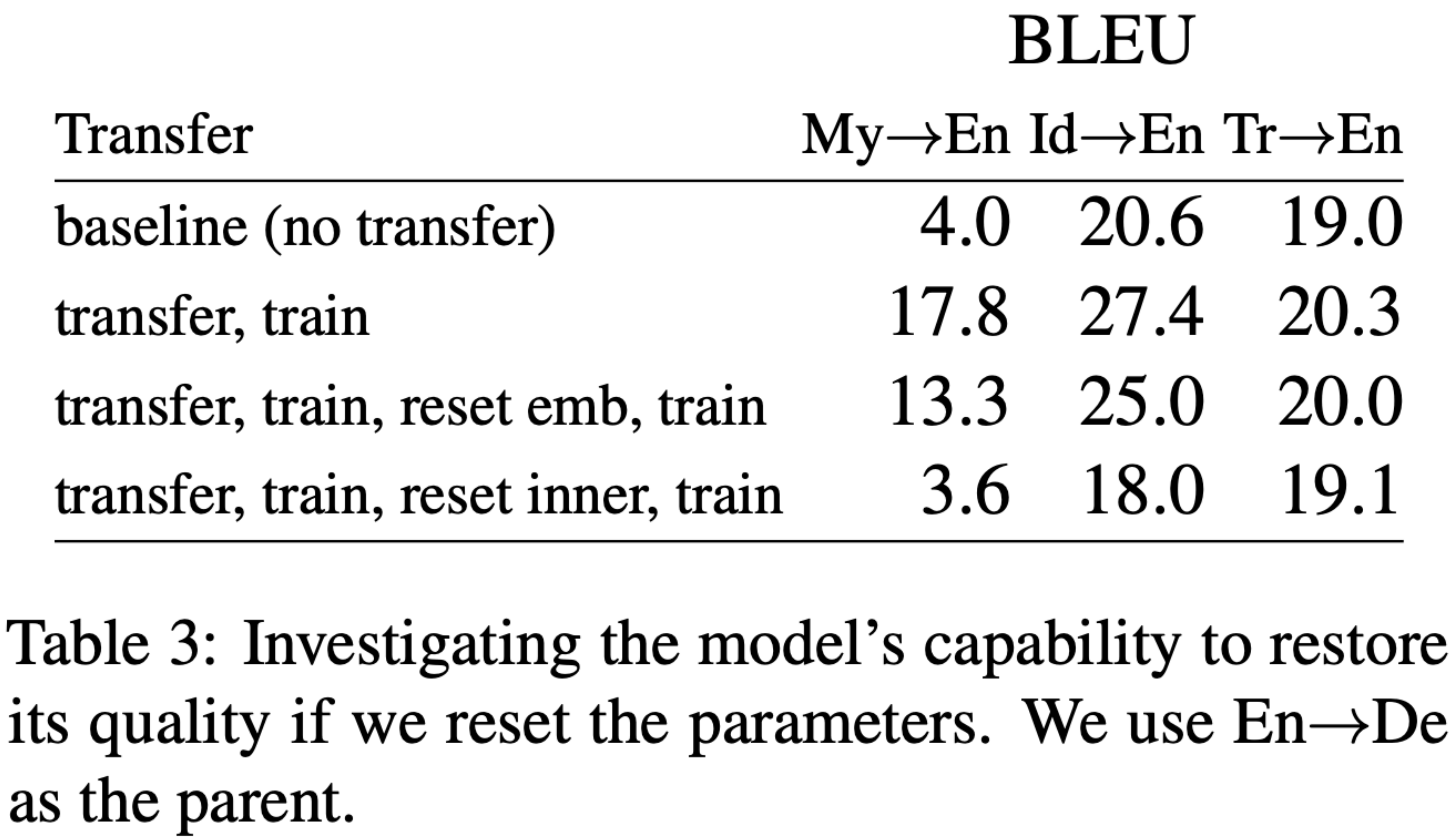
Frontiers in MT: Low-Resource

- Particular interest in deploying MT systems for languages with little or no parallel data

- BPE allows us to transfer models even without training on a specific language

- Pre-trained models can help further

Burmese, Indonesian, Turkish



	BLEU		
Transfer	My→En	Id→En	Tr→En
baseline (no transfer)	4.0	20.6	19.0
transfer, train	17.8	27.4	20.3
transfer, train, reset emb, train	13.3	25.0	20.0
transfer, train, reset inner, train	3.6	18.0	19.1

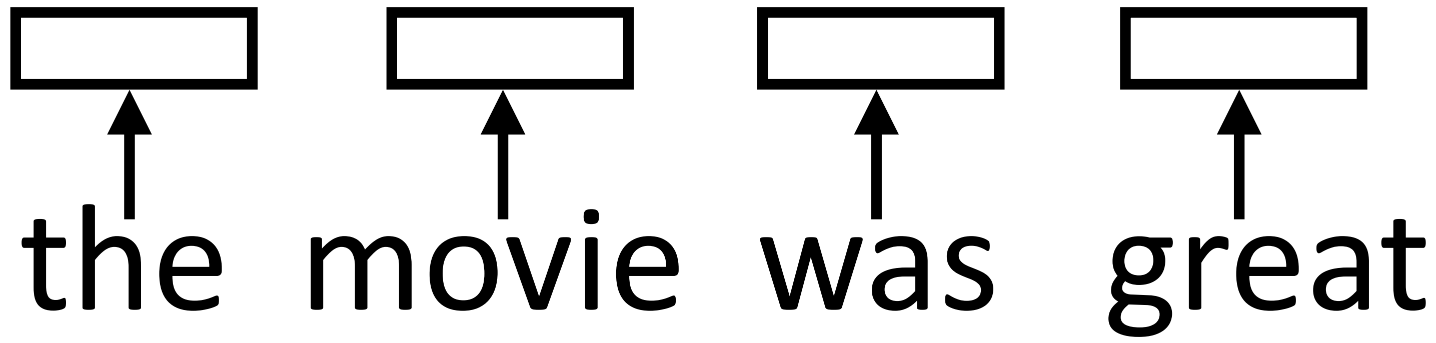
Table 3: Investigating the model's capability to restore its quality if we reset the parameters. We use En→De as the parent.

Aji et al. (2020)

Transformers for MT

Recall: Self-Attention

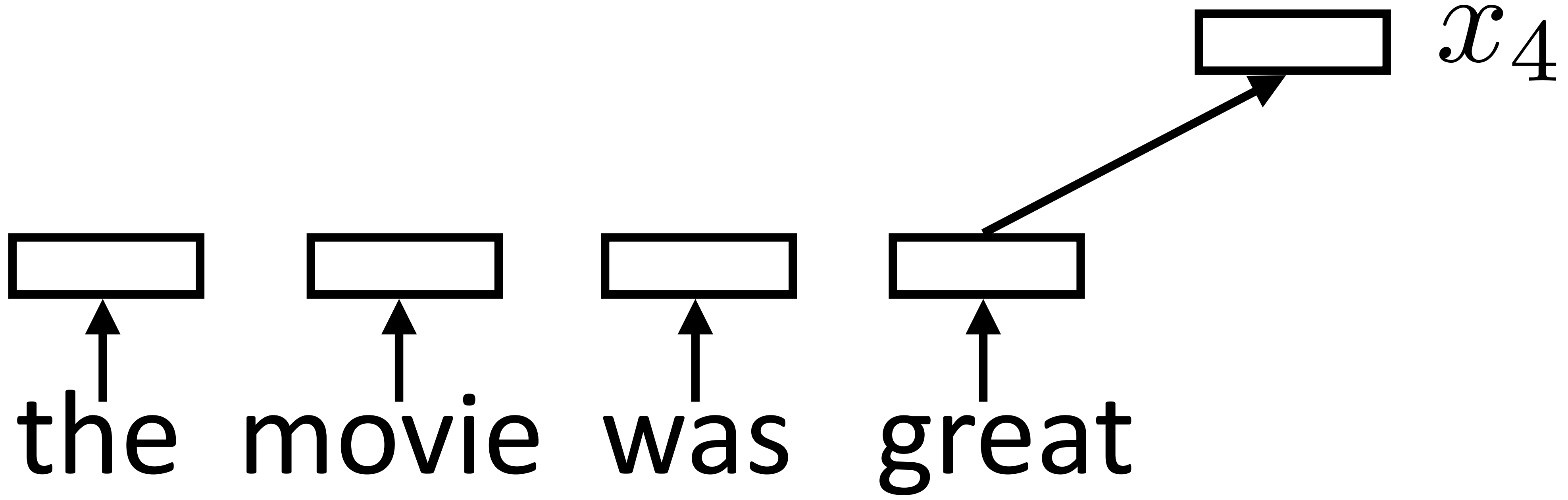
- Each word forms a “query” which then computes attention over each word



Vaswani et al. (2017)

Recall: Self-Attention

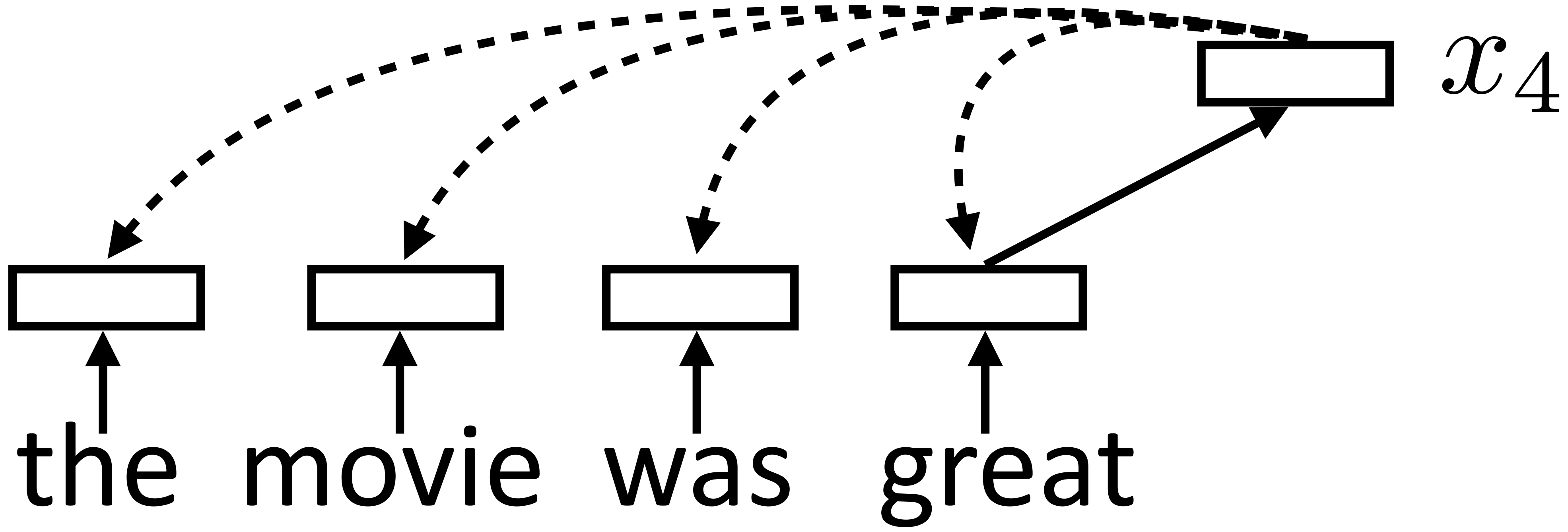
- Each word forms a “query” which then computes attention over each word



Vaswani et al. (2017)

Recall: Self-Attention

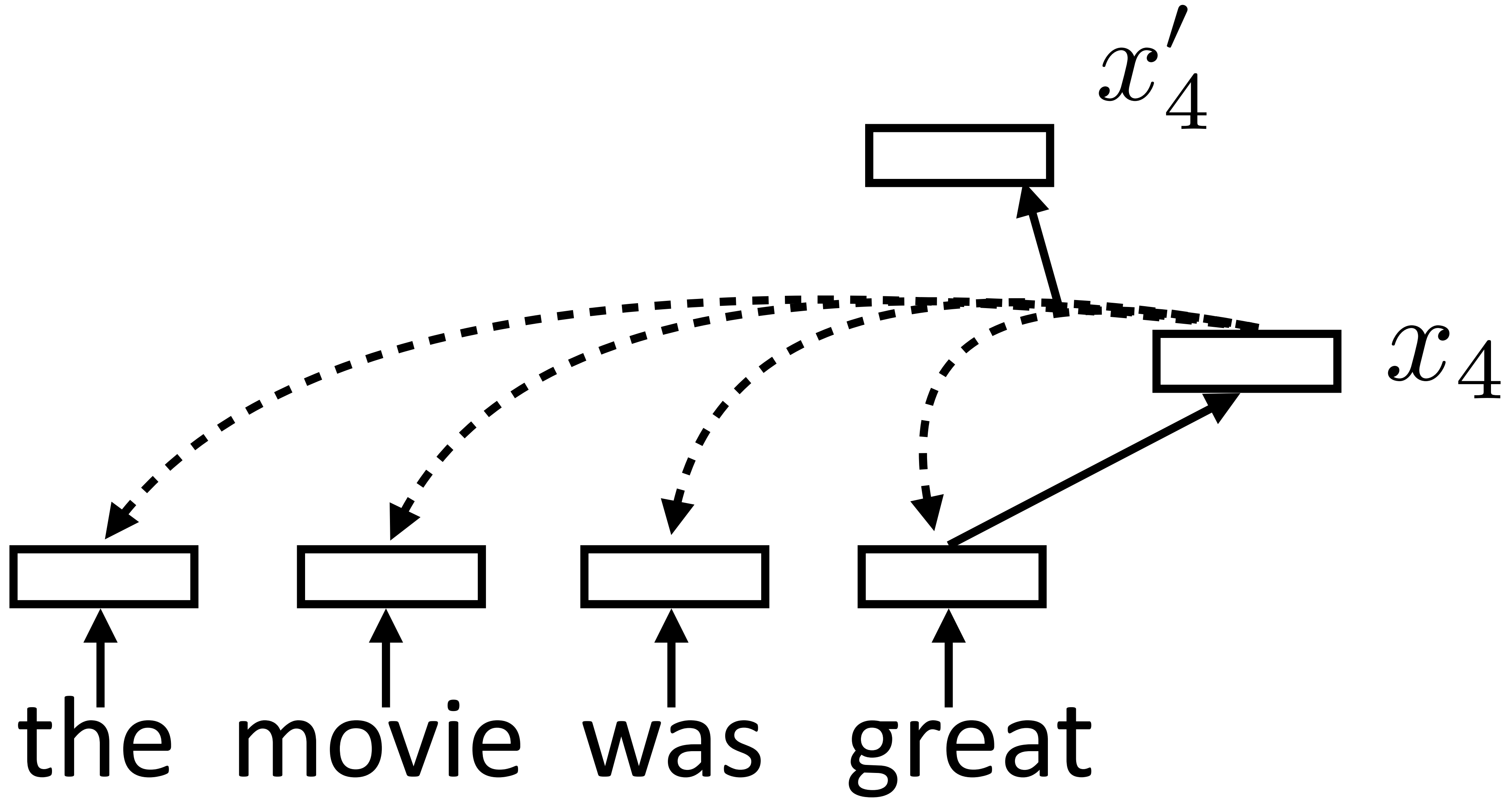
- Each word forms a "query" which then computes attention over each word



Vaswani et al. (2017)

Recall: Self-Attention

- Each word forms a "query" which then computes attention over each word

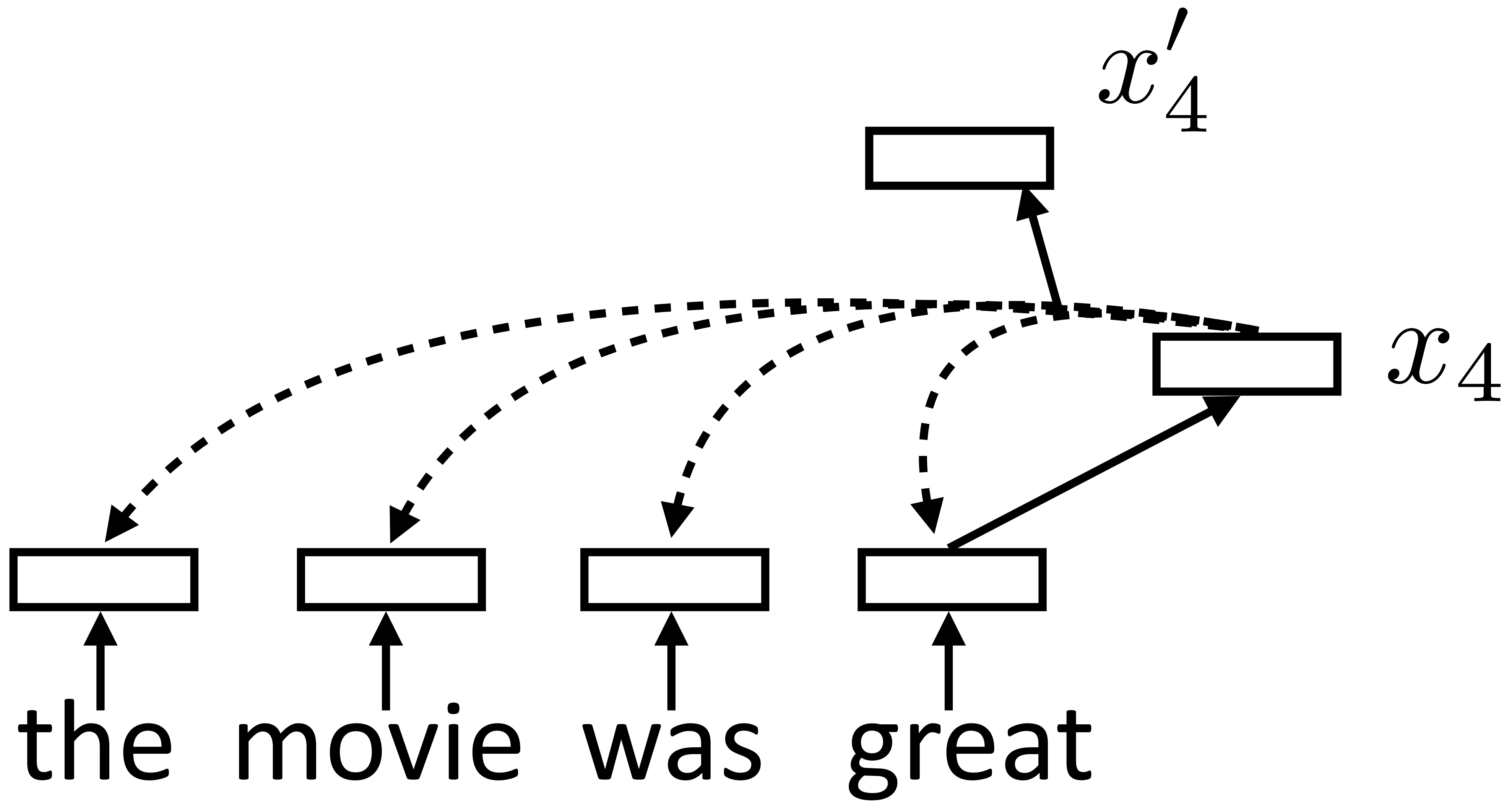


Vaswani et al. (2017)

Recall: Self-Attention

- Each word forms a “query” which then computes attention over each word

$\alpha_{i,j} = \mathrm{softmax}(x_i^\top x_j)$ scalar



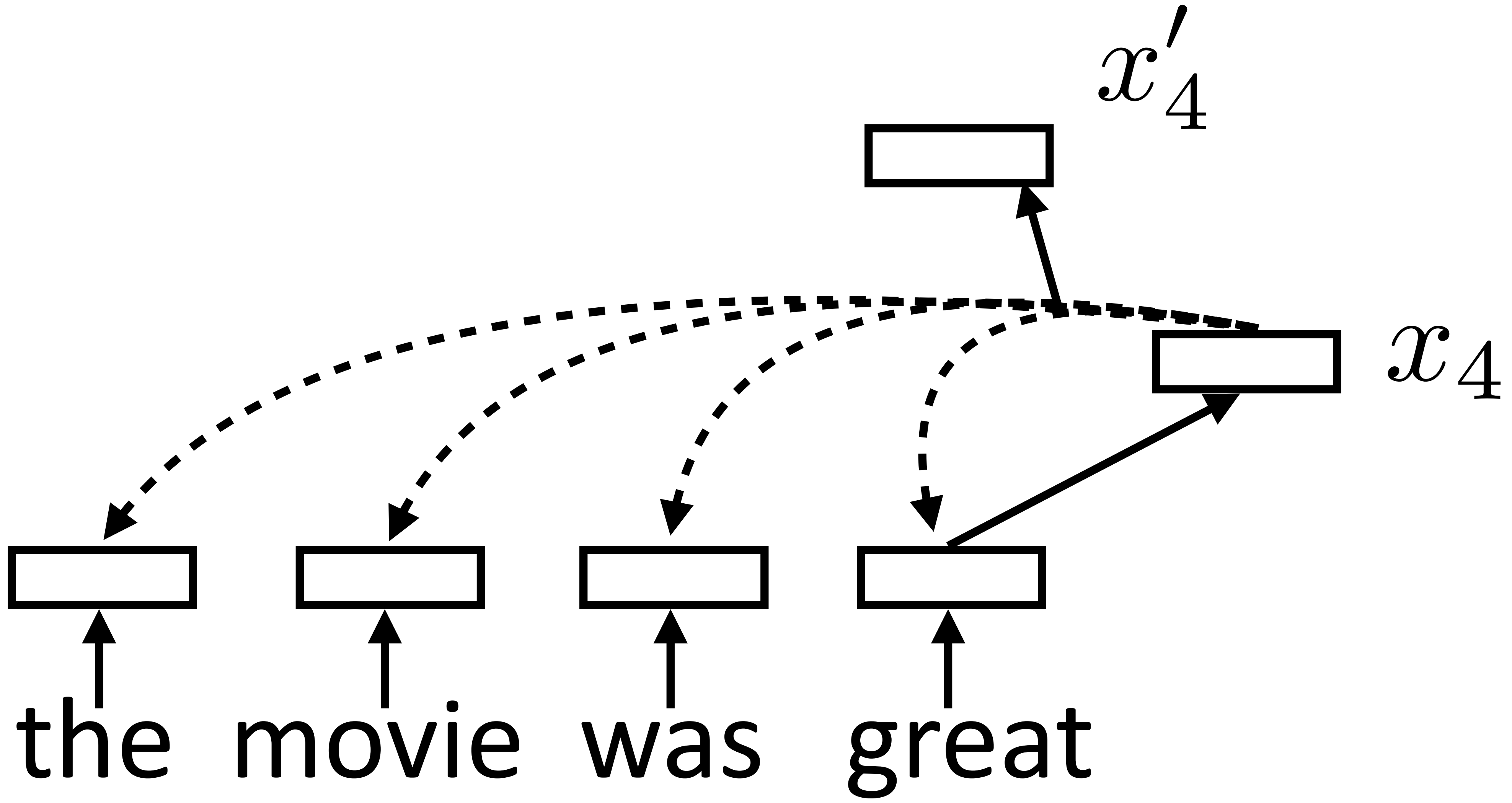
Vaswani et al. (2017)

Recall: Self-Attention

- Each word forms a “query” which then computes attention over each word

$$\alpha_{i,j} = \mathrm{softmax}(x_i^\top x_j)$$ scalar

$$x_i' = \sum_{j=1}^{n} \alpha_{i,j} x_j$$ vector = sum of scalar * vector



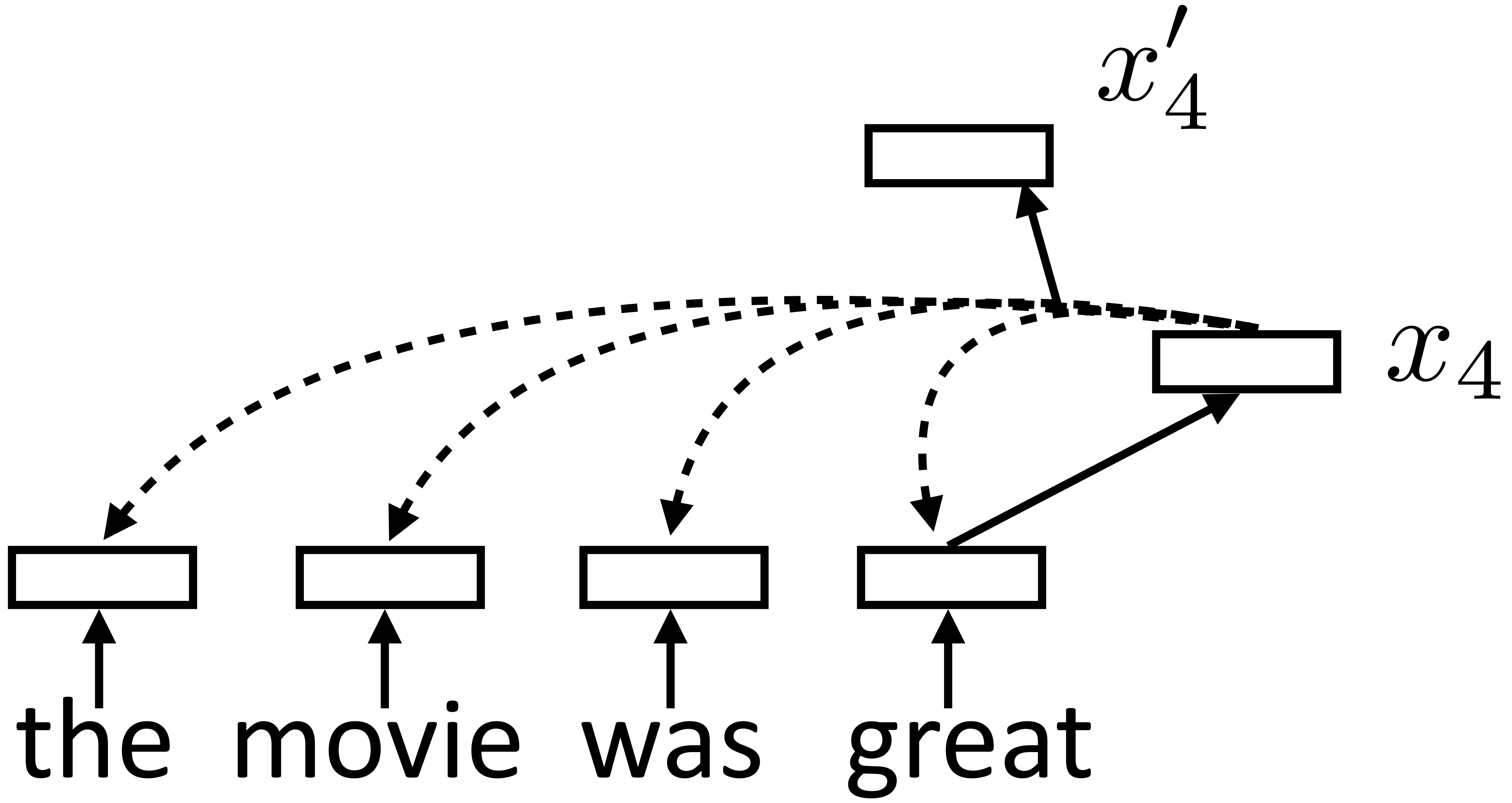
Vaswani et al. (2017)

Recall: Self-Attention

- Each word forms a “query” which then computes attention over each word

$$\alpha_{i,j} = \mathrm{softmax}(x_i^\top x_j)$$ scalar

$$x'_i = \sum_{j=1}^{n} \alpha_{i,j} x_j$$ vector = sum of scalar * vector



- Multiple “heads” analogous to different convolutional filters. Use parameters W_k and V_k to get different attention values + transform vectors

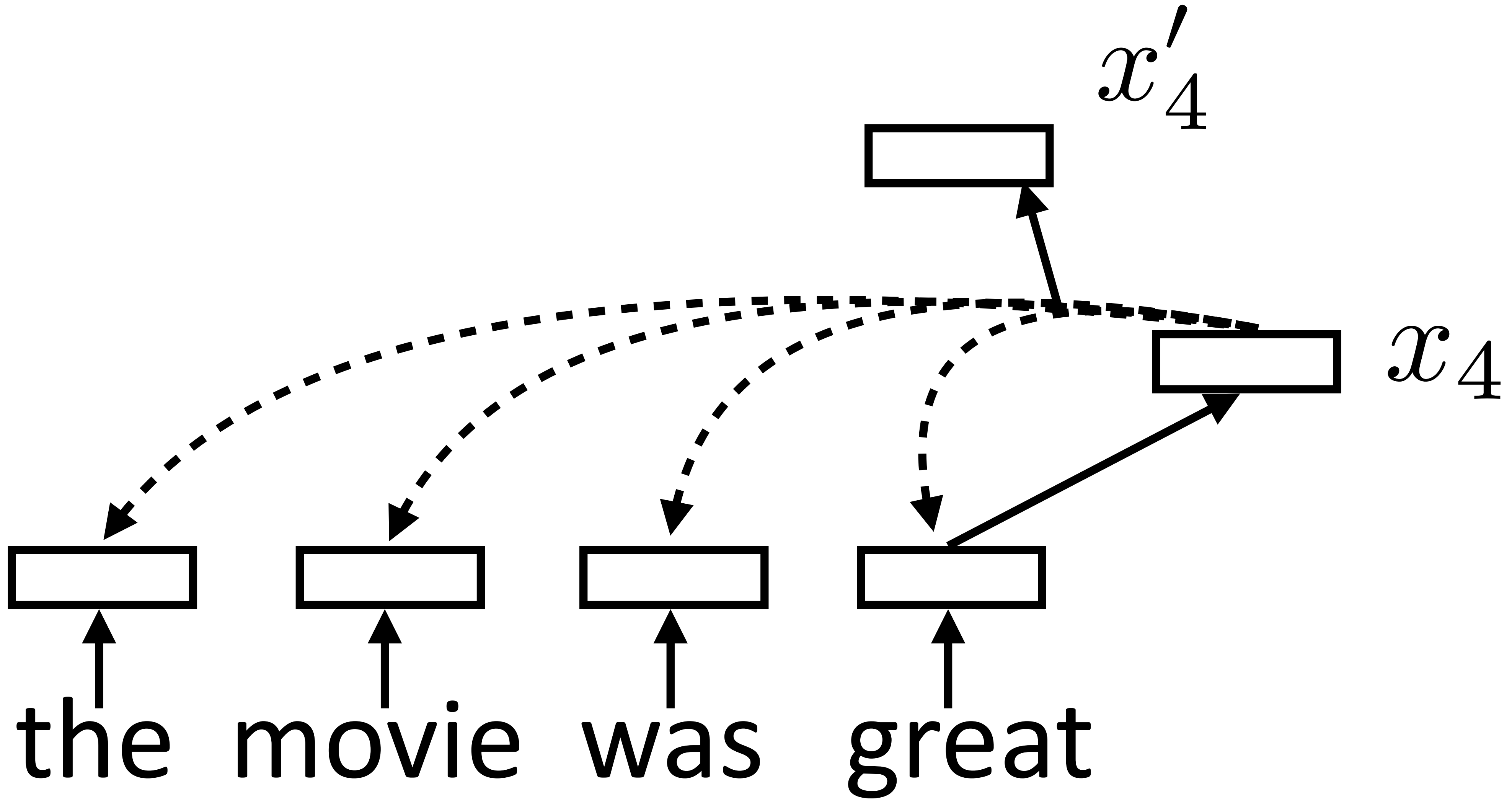
Vaswani et al. (2017)

Recall: Self-Attention

- Each word forms a "query" which then computes attention over each word

$$\alpha_{i,j} = \text{softmax}(x_i^\top x_j)$$ scalar

$$x_i' = \sum_{j=1}^{n} \alpha_{i,j} x_j$$ vector = sum of scalar * vector



- Multiple "heads" analogous to different convolutional filters. Use parameters *W_k* and *V_k* to get different attention values + transform vectors

$$\alpha_{k,i,j} = \text{softmax}(x_i^\top W_k x_j)$$

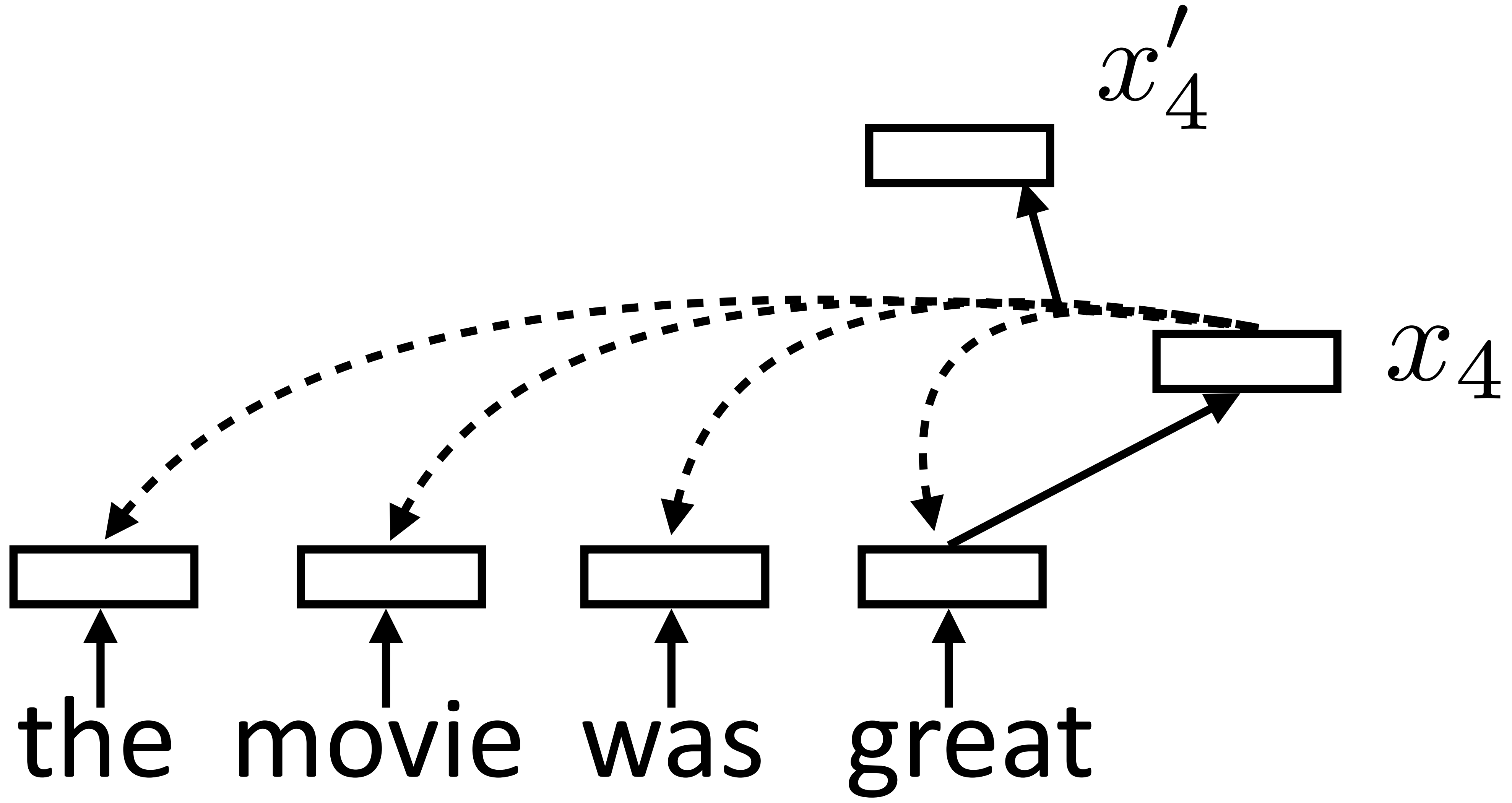
Vaswani et al. (2017)

Recall: Self-Attention

- Each word forms a "query" which then computes attention over each word

$$\alpha_{i,j} = \mathrm{softmax}(x_i^\top x_j)$$ scalar

$$x'_i = \sum_{j=1}^{n} \alpha_{i,j} x_j$$ vector = sum of scalar * vector



- Multiple "heads" analogous to different convolutional filters. Use parameters *W*

$$\alpha_{k,i,j} = \mathrm{softmax}(x_i^\top W_k x_j) \quad x'_{k,i} = \sum_{j=1}^{n} \alpha_{k,i,j} V_k x_j$$

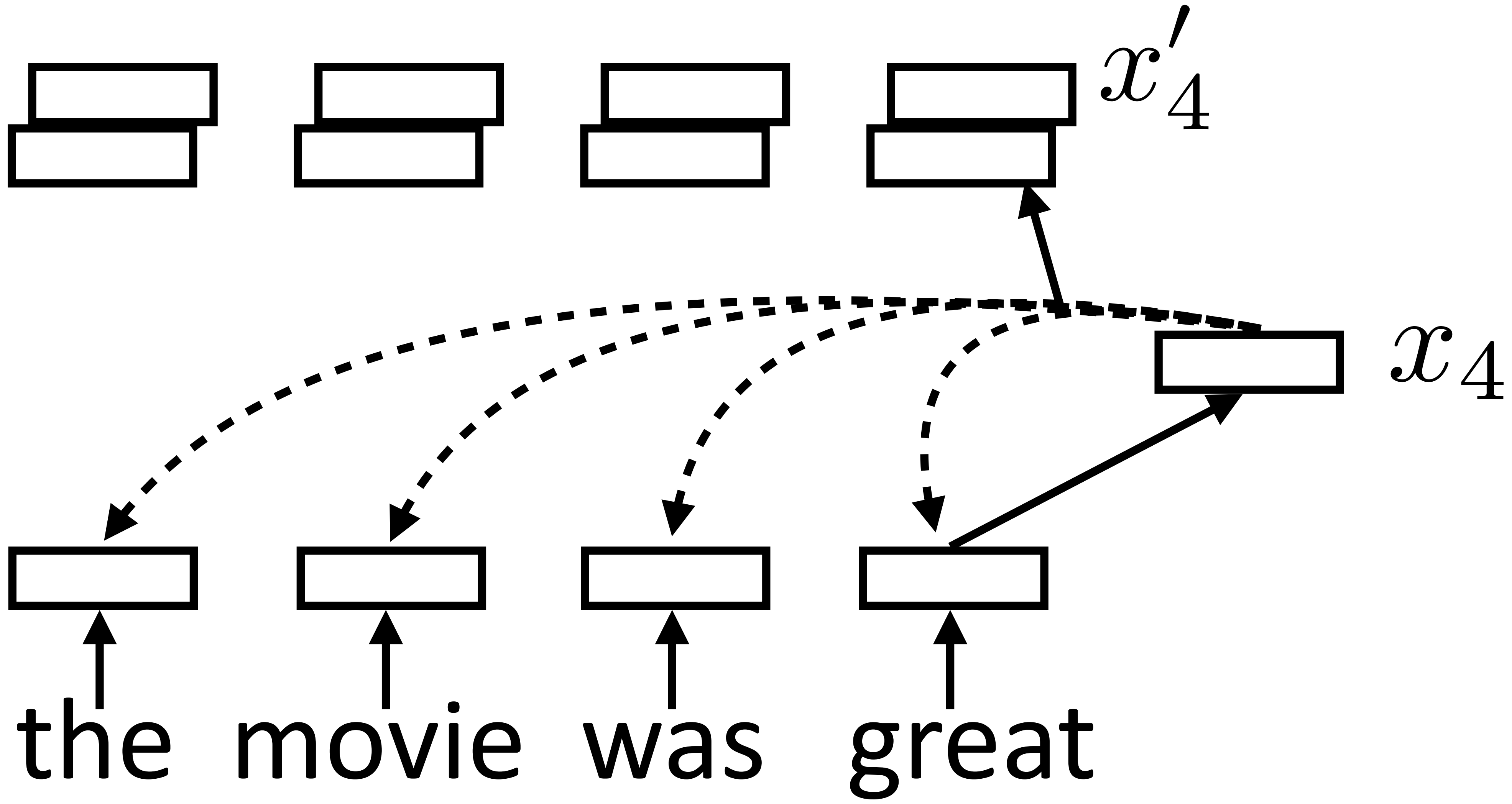
Vaswani et al. (2017)

Recall: Self-Attention

- Each word forms a "query" which then computes attention over each word

$$\alpha_{i,j} = \text{softmax}(x_i^\top x_j)$$ scalar

$$x'_i = \sum_{j=1}^{n} \alpha_{i,j} x_j$$ vector = sum of scalar * vector



- Multiple "heads" analogous to different convolutional filters. Use parameters *W_k* and *V_k* to get different attention values + transform vectors

$$\alpha_{k,i,j} = \text{softmax}(x_i^\top W_k x_j) \qquad x'_{k,i} = \sum_{j=1}^{n} \alpha_{k,i,j} V_k x_j$$

Vaswani et al. (2017)

Multi-Head Self Attention

- Multiple "heads" analogous to different convolutional filters
- Let X = [sent len, embedding dim] be the input sentence
- Query $Q = W^Q X$: these are like the **decoder hidden state** in attention
- Keys $K = W^K X$: these control what gets attended to, along with the query
- Values $V = W^V X$: these vectors get summed up to form the output

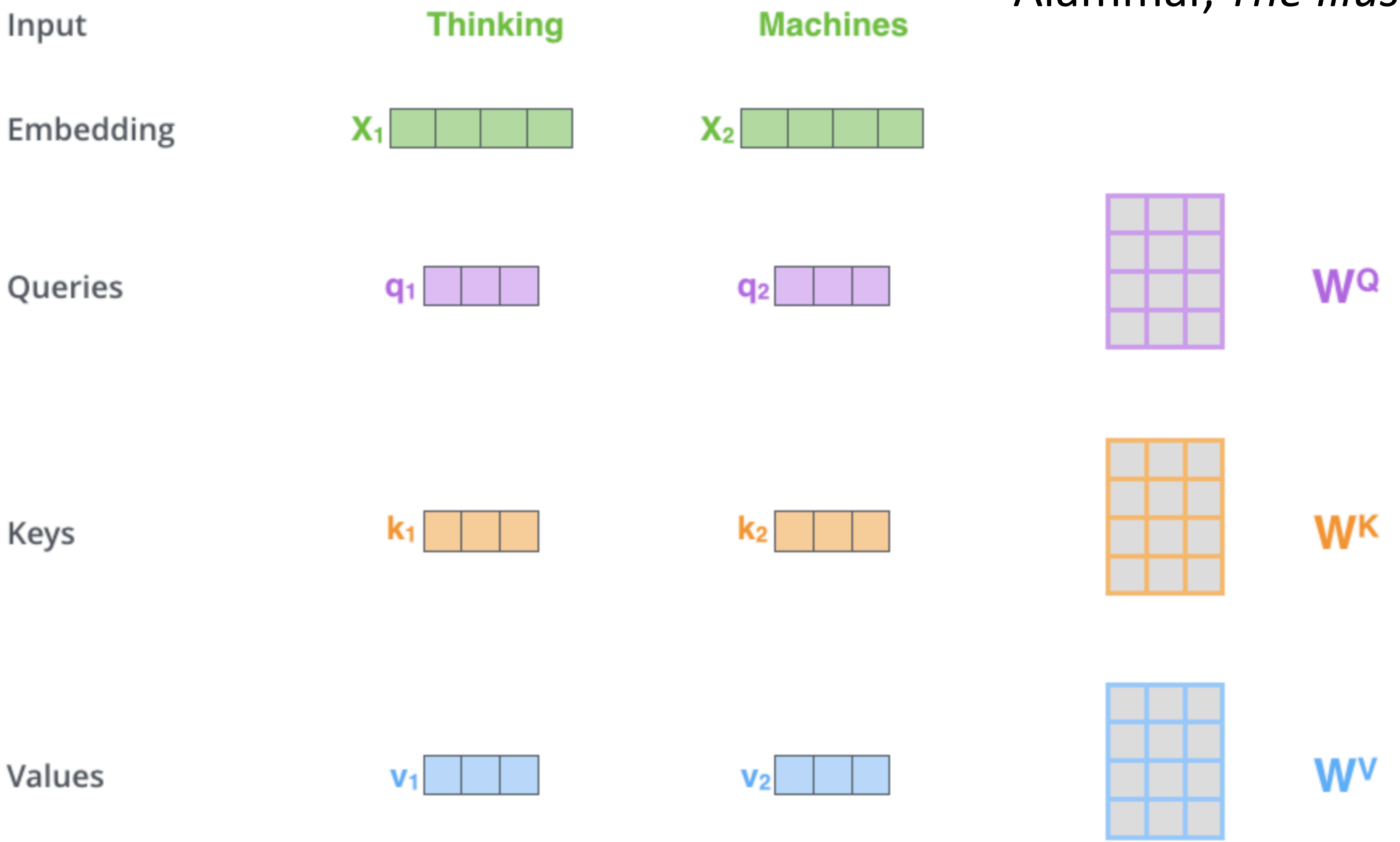
$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

dim of keys

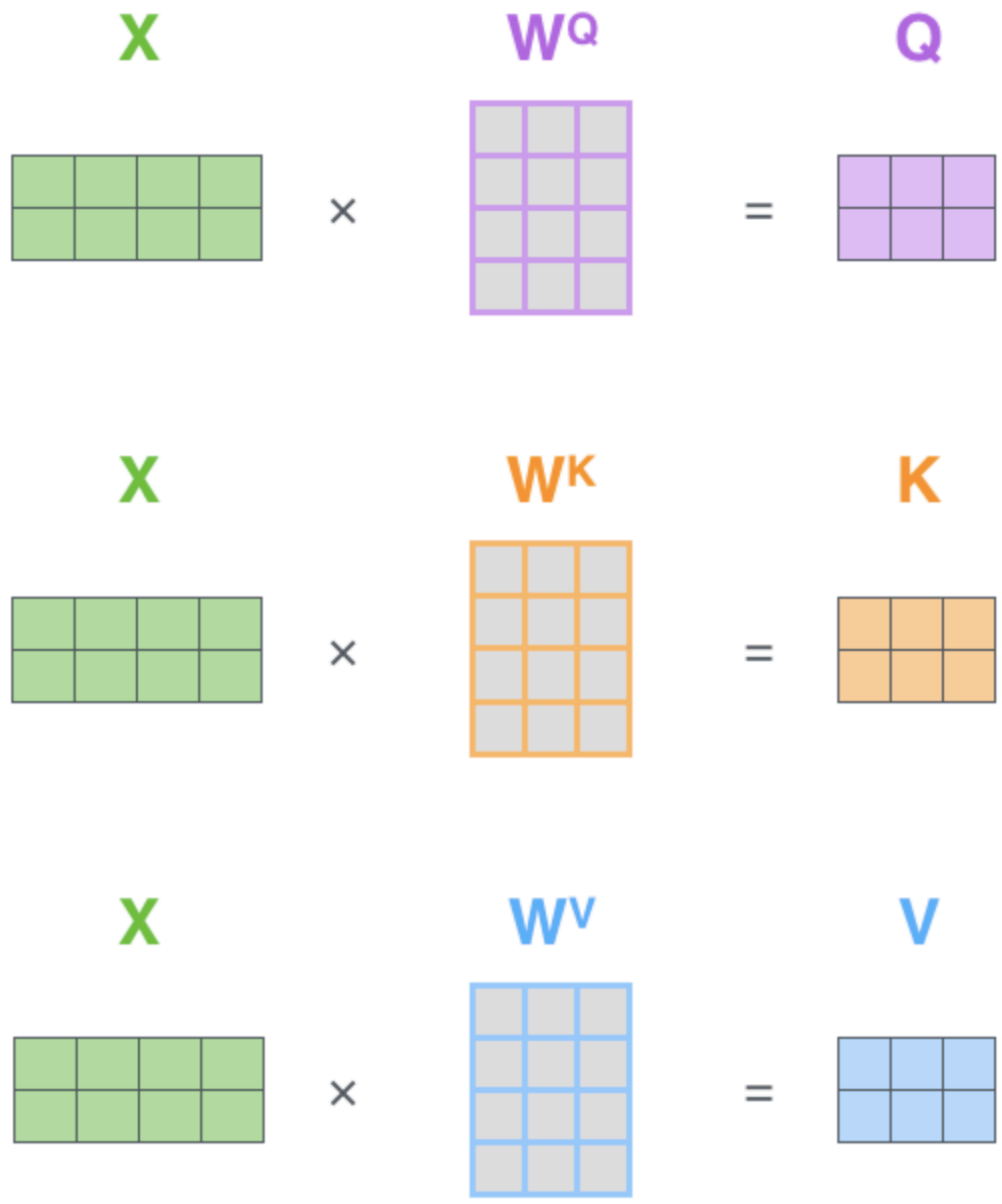
Vaswani et al. (2017)

Multi-Head Self Attention

Alammar, *The Illustrated Transformer*

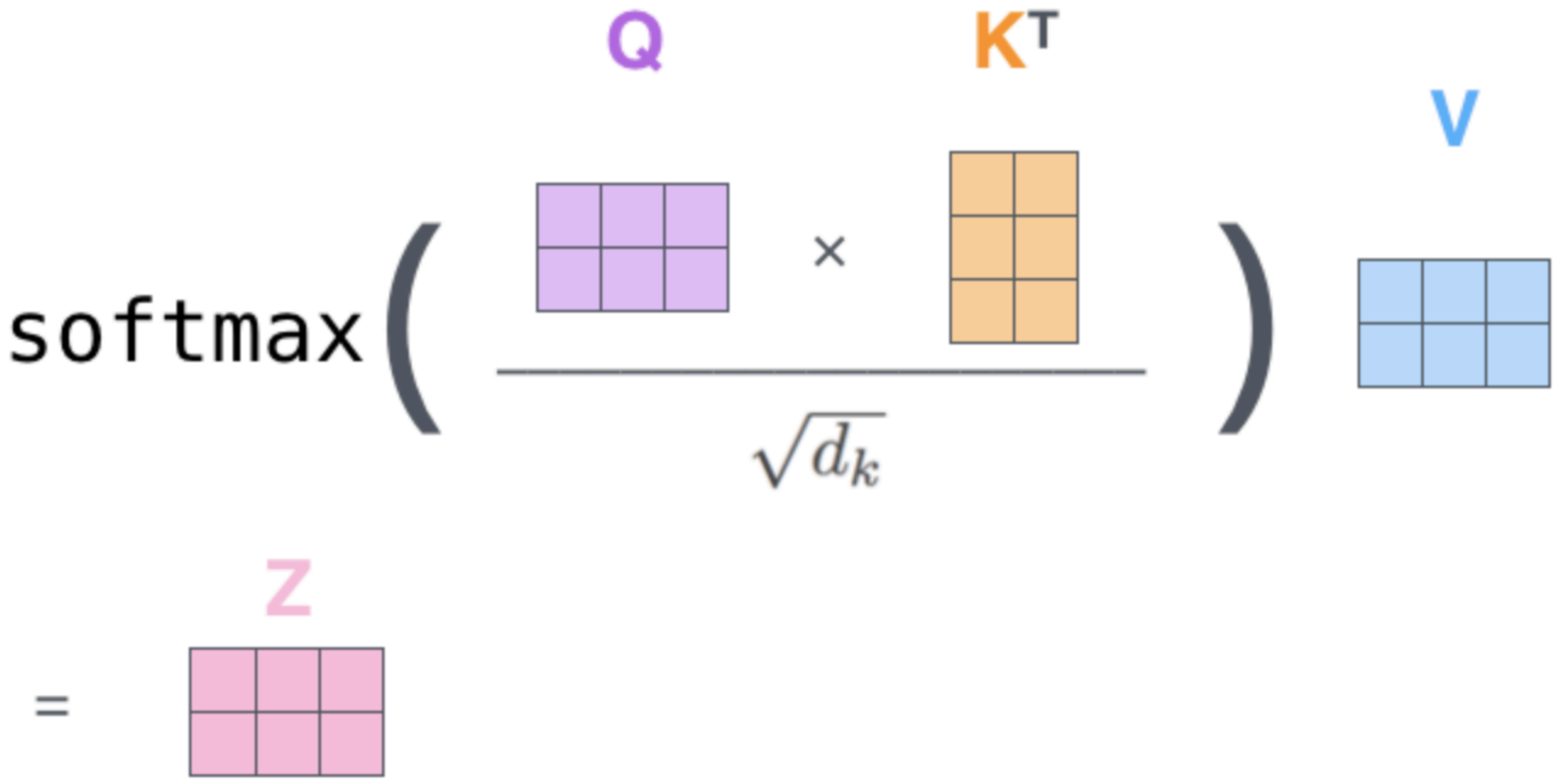


Multi-Head Self Attention



Alammar, *The Illustrated Transformer*

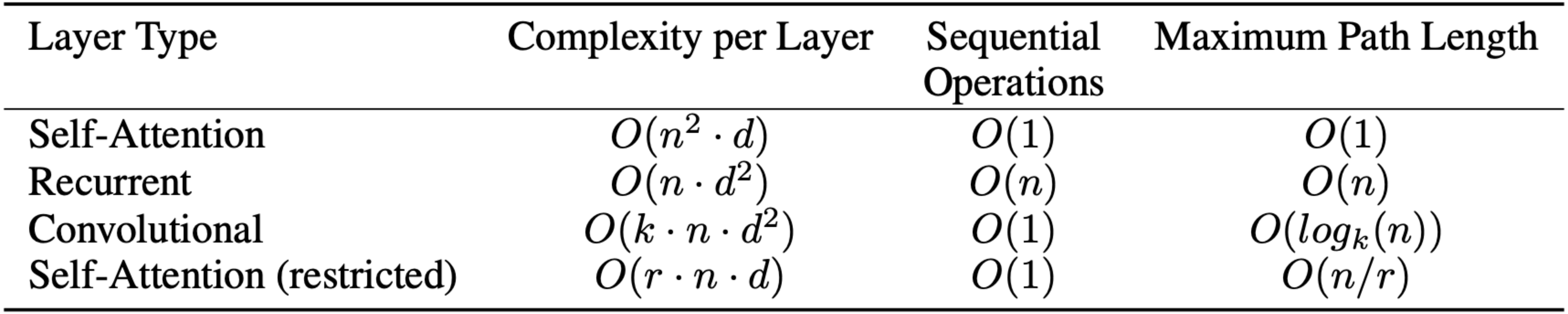
sent len x sent len (attn for each word to each other)



sent len x hidden dim

Z is a weighted combination of V rows

Properties of Self-Attention



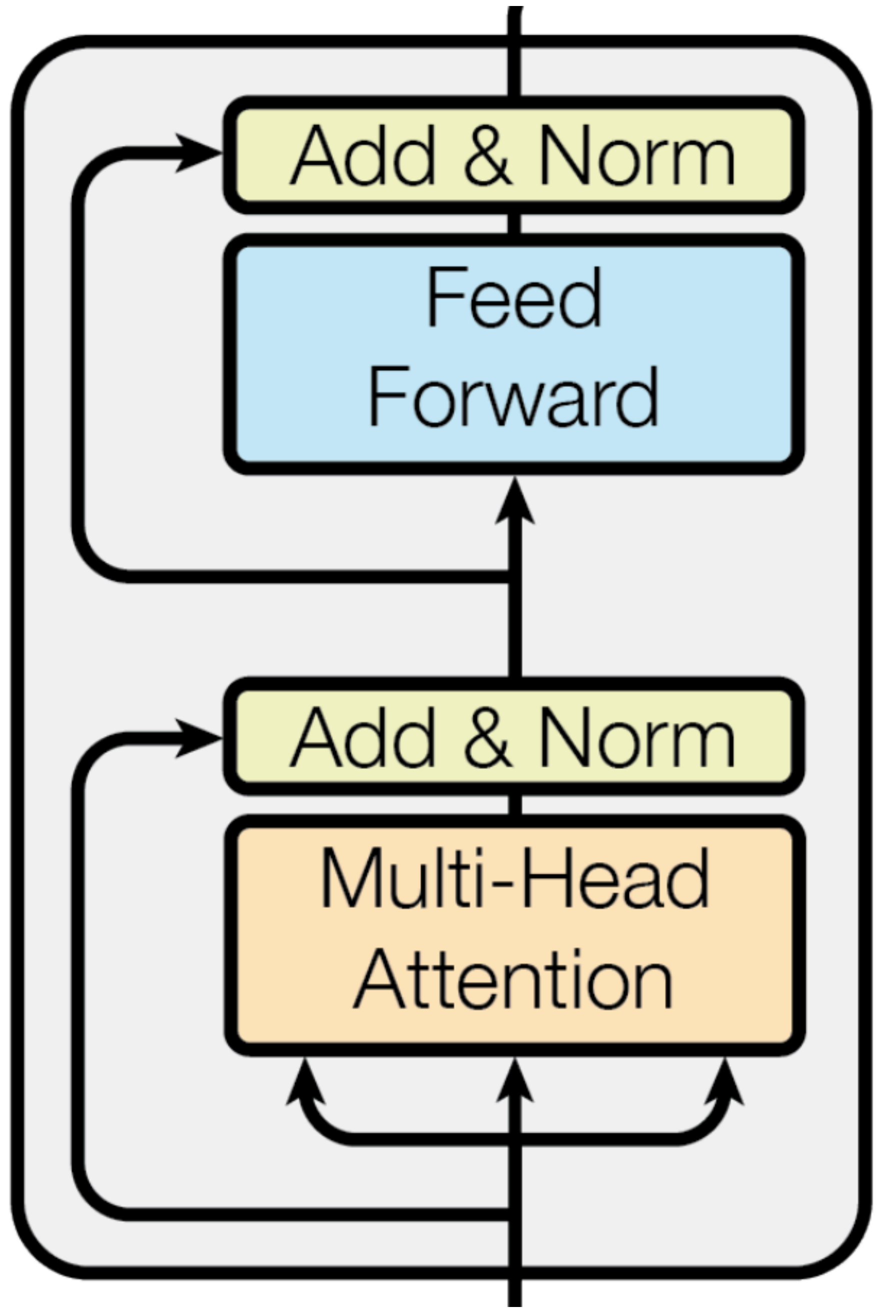
Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

- *n* = sentence length, *d* = hidden dim, *k* = kernel size, *r* = restricted neighborhood size

- **Quadratic complexity**, but O(1) sequential operations (not linear like in RNNs) and O(1) "path" for words to inform each other

Vaswani et al. (2017)

Transformers

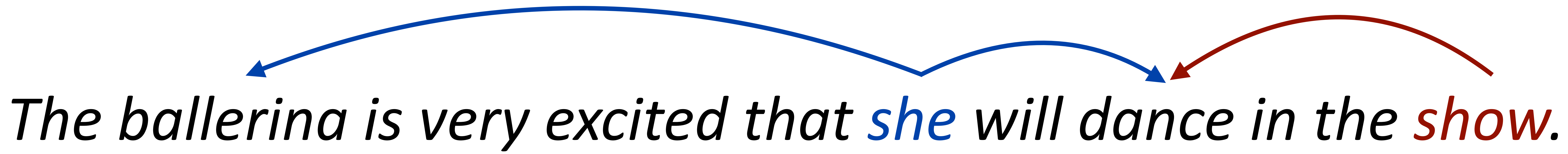


- Alternate multi-head self-attention layers and feedforward layers

- Residual connections let the model “skip” each layer — these are particularly useful for training deep networks

Vaswani et al. (2017)

Transformers: Position Sensitivity

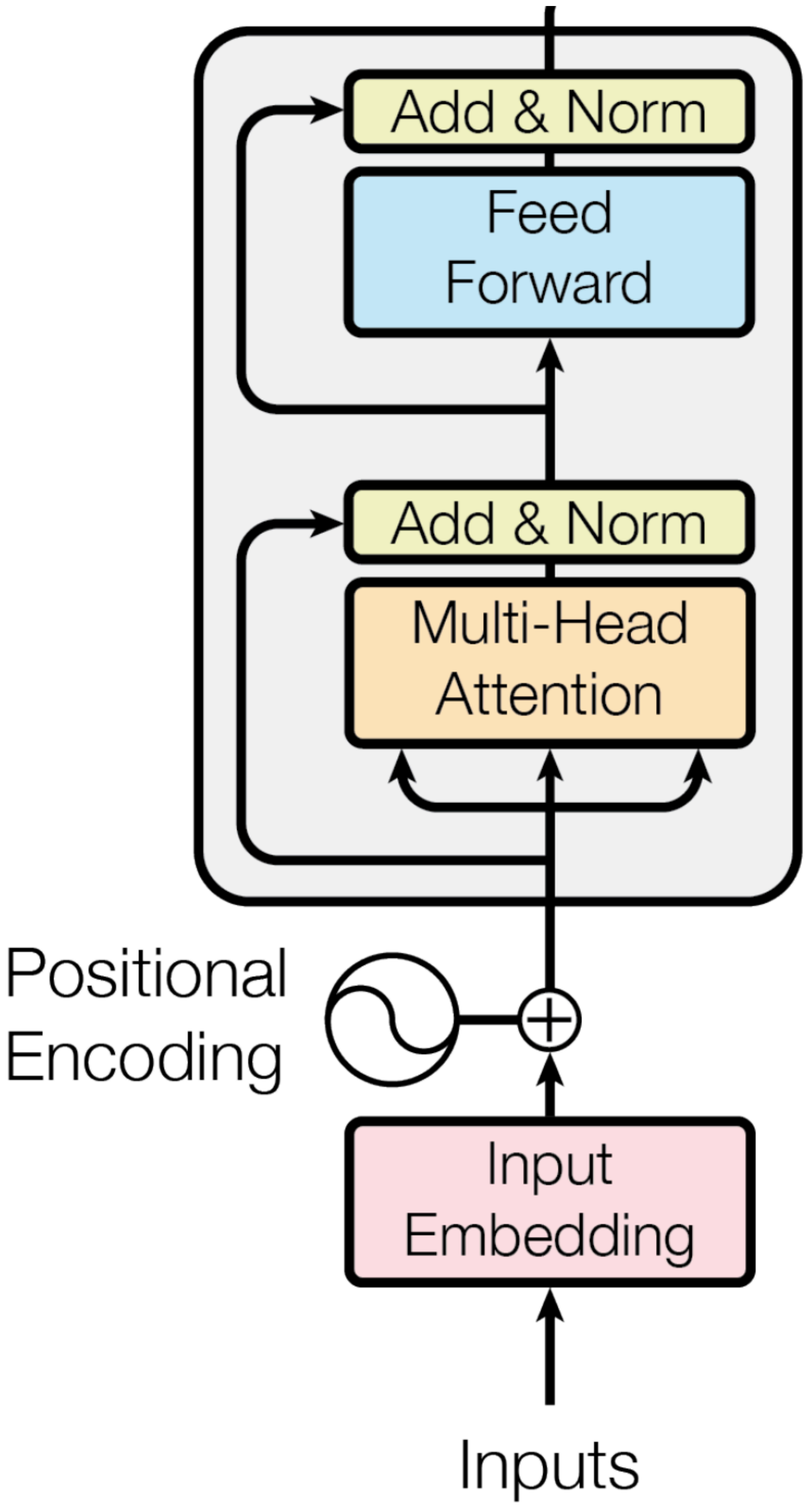


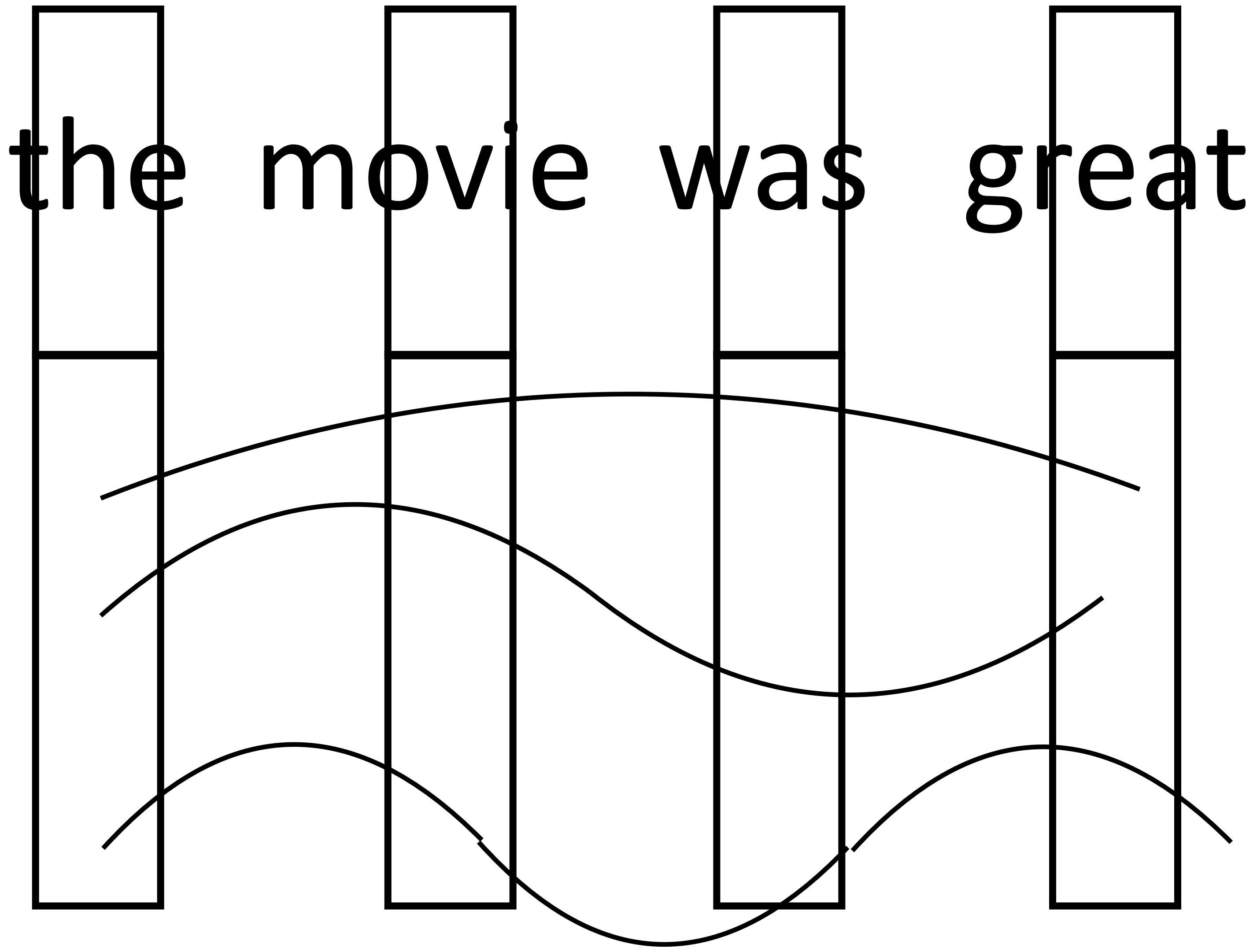
The ballerina is very excited that she will dance in the show.

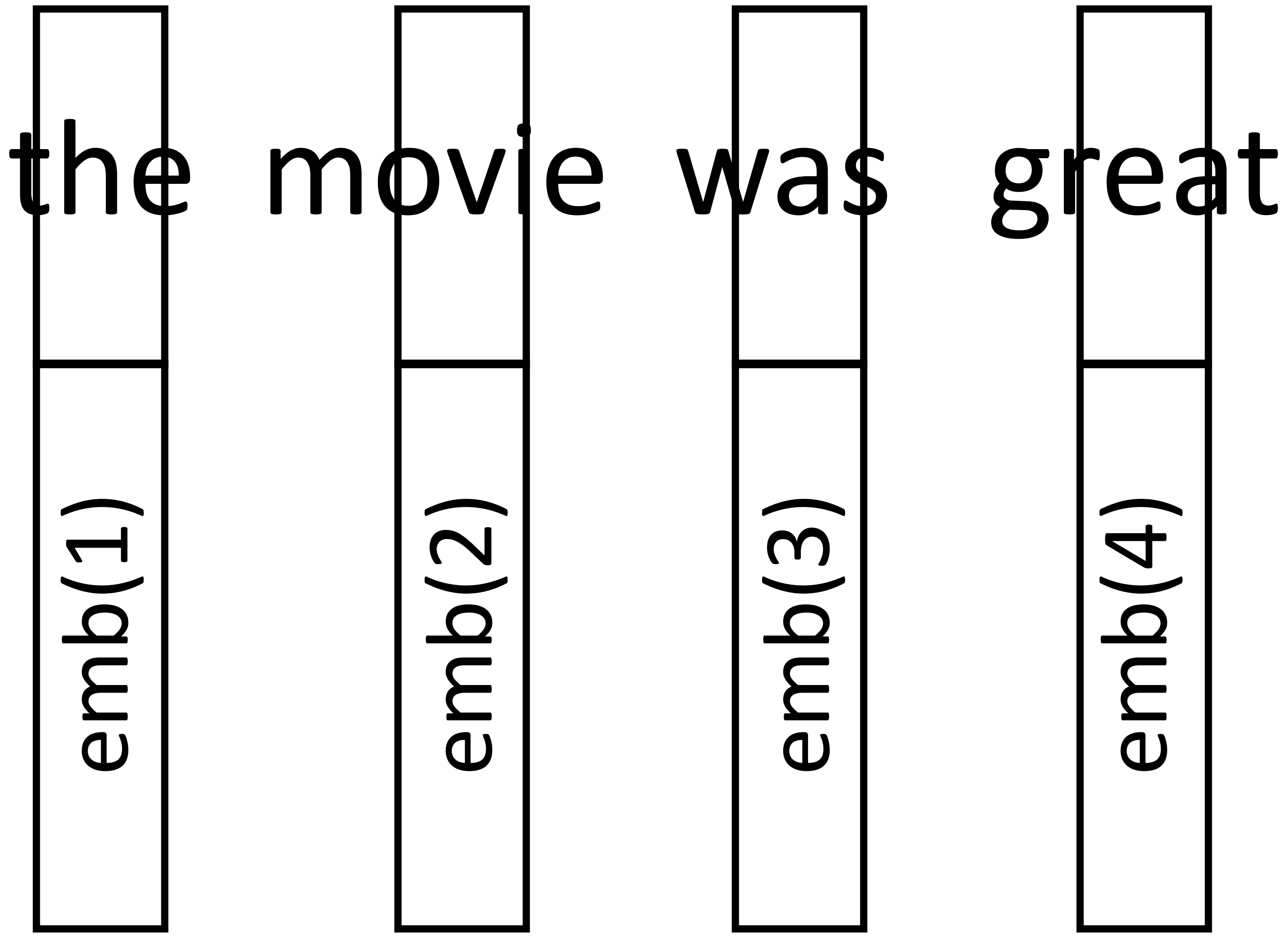
- If this is in a longer context, we want words to attend *locally*
- But transformers have *no notion of position* by default

Vaswani et al. (2017)

Transformers: Position Sensitivity





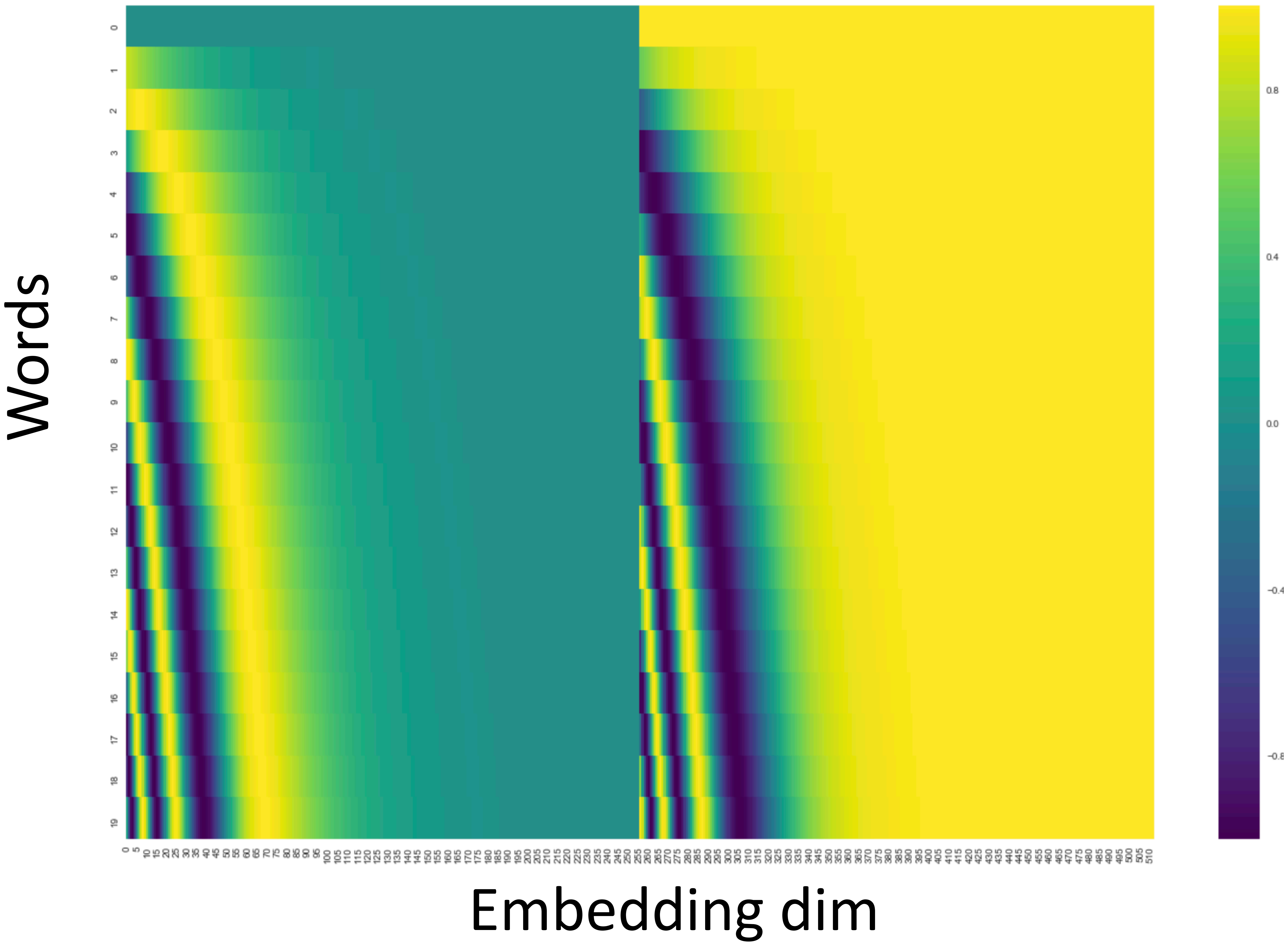


- Augment word embedding with position embeddings, each dim is a sine/cosine wave of a different frequency. Closer points = higher dot products
- Works essentially as well as just encoding position as a one-hot vector

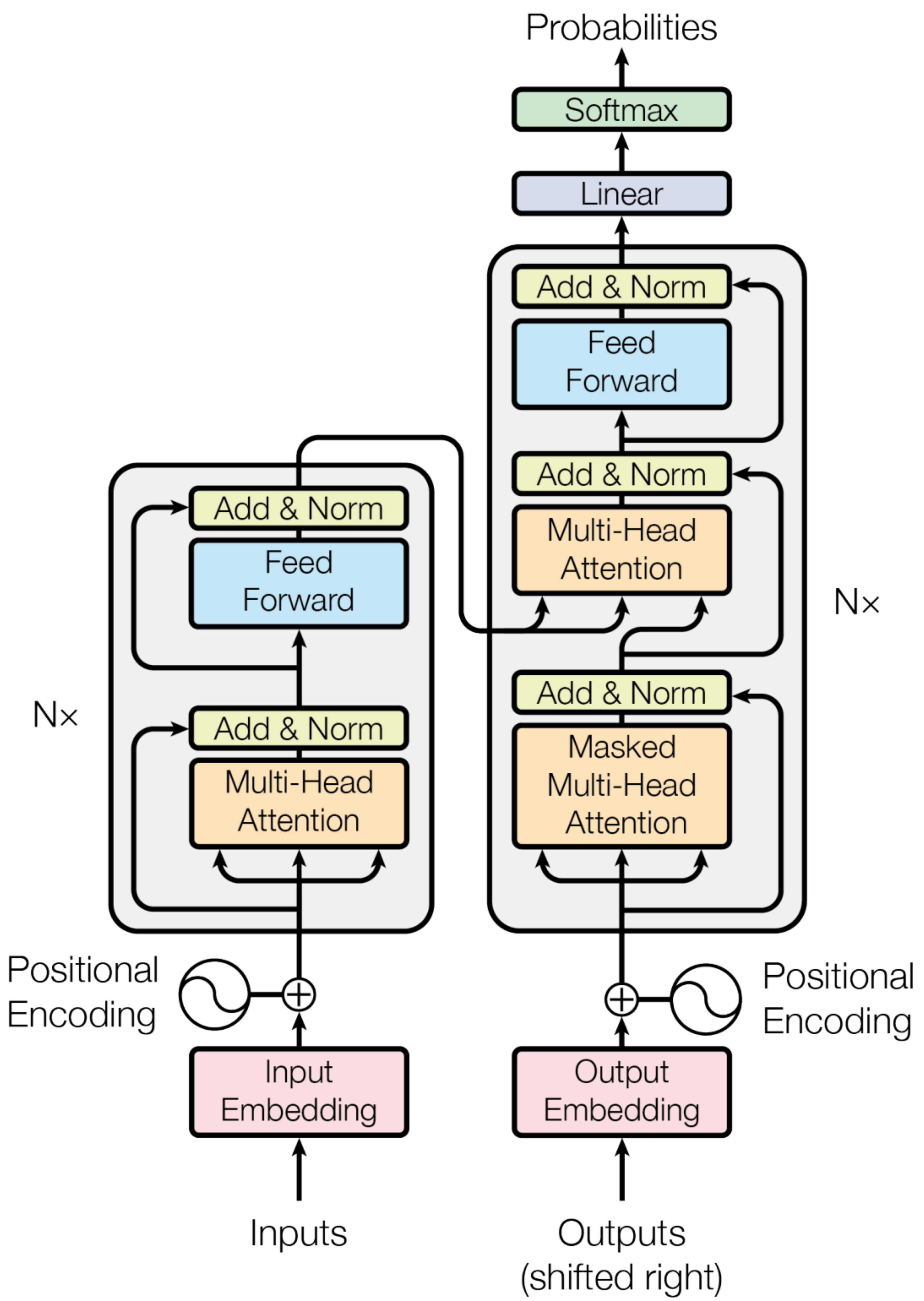
Vaswani et al. (2017)

Transformers

Alammar, *The Illustrated Transformer*



Transformers: Complete Model



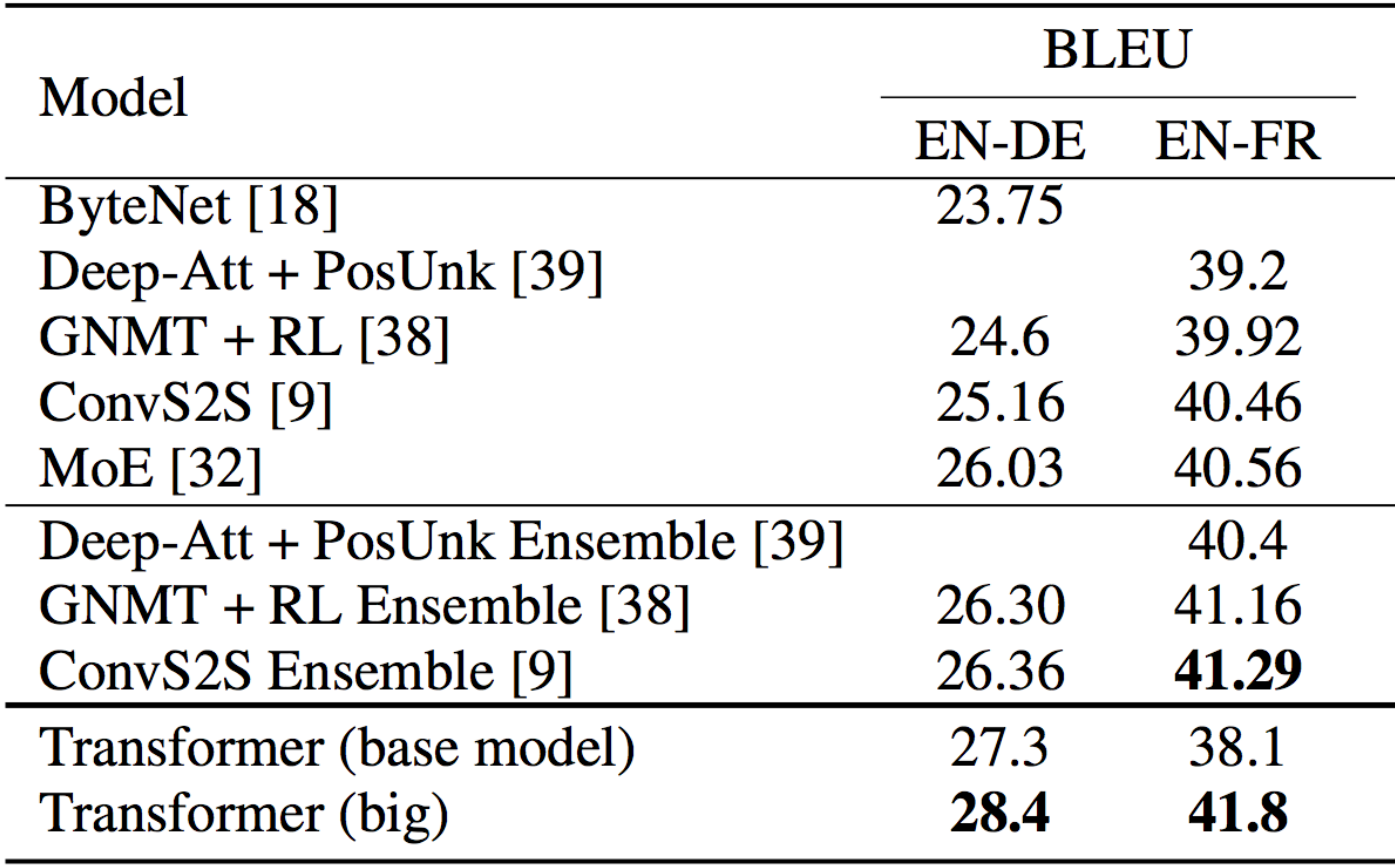
- Encoder and decoder are both transformers

- Decoder alternates attention over the output and attention over the input as well

- Decoder consumes the previous generated tokens but has *no recurrent state*

Vaswani et al. (2017)

Transformers

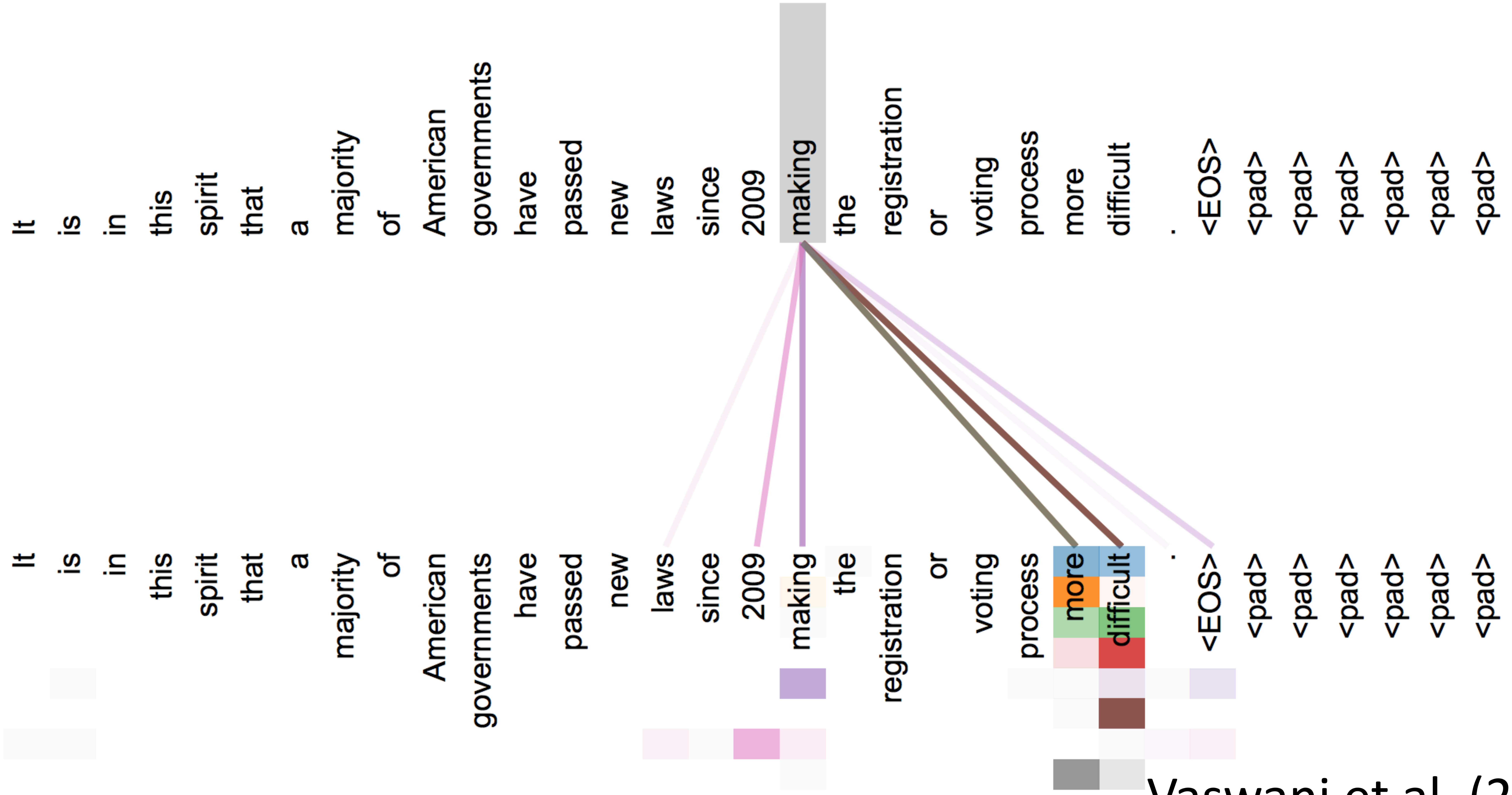


Model	BLEU	
	EN-DE	EN-FR
ByteNet [18]	23.75	
Deep-Att + PosUnk [39]		39.2
GNMT + RL [38]	24.6	39.92
ConvS2S [9]	25.16	40.46
MoE [32]	26.03	40.56
Deep-Att + PosUnk Ensemble [39]		40.4
GNMT + RL Ensemble [38]	26.30	41.16
ConvS2S Ensemble [9]	26.36	**41.29**
Transformer (base model)	27.3	38.1
Transformer (big)	**28.4**	**41.8**

- Big = 6 layers, 1000 dim for each token, 16 heads, base = 6 layers + other params halved

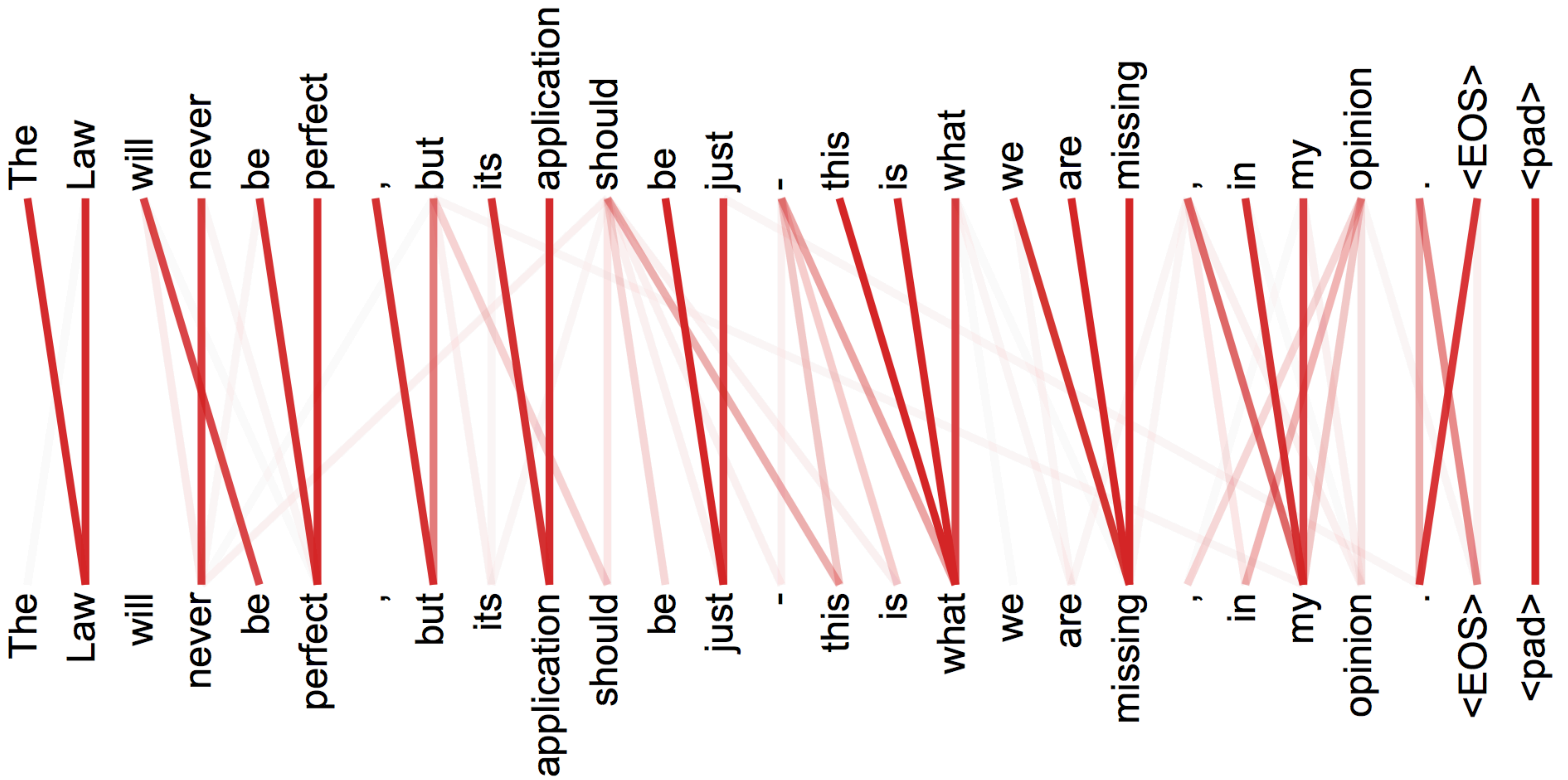
Vaswani et al. (2017)

Visualization



Vaswani et al. (2017)

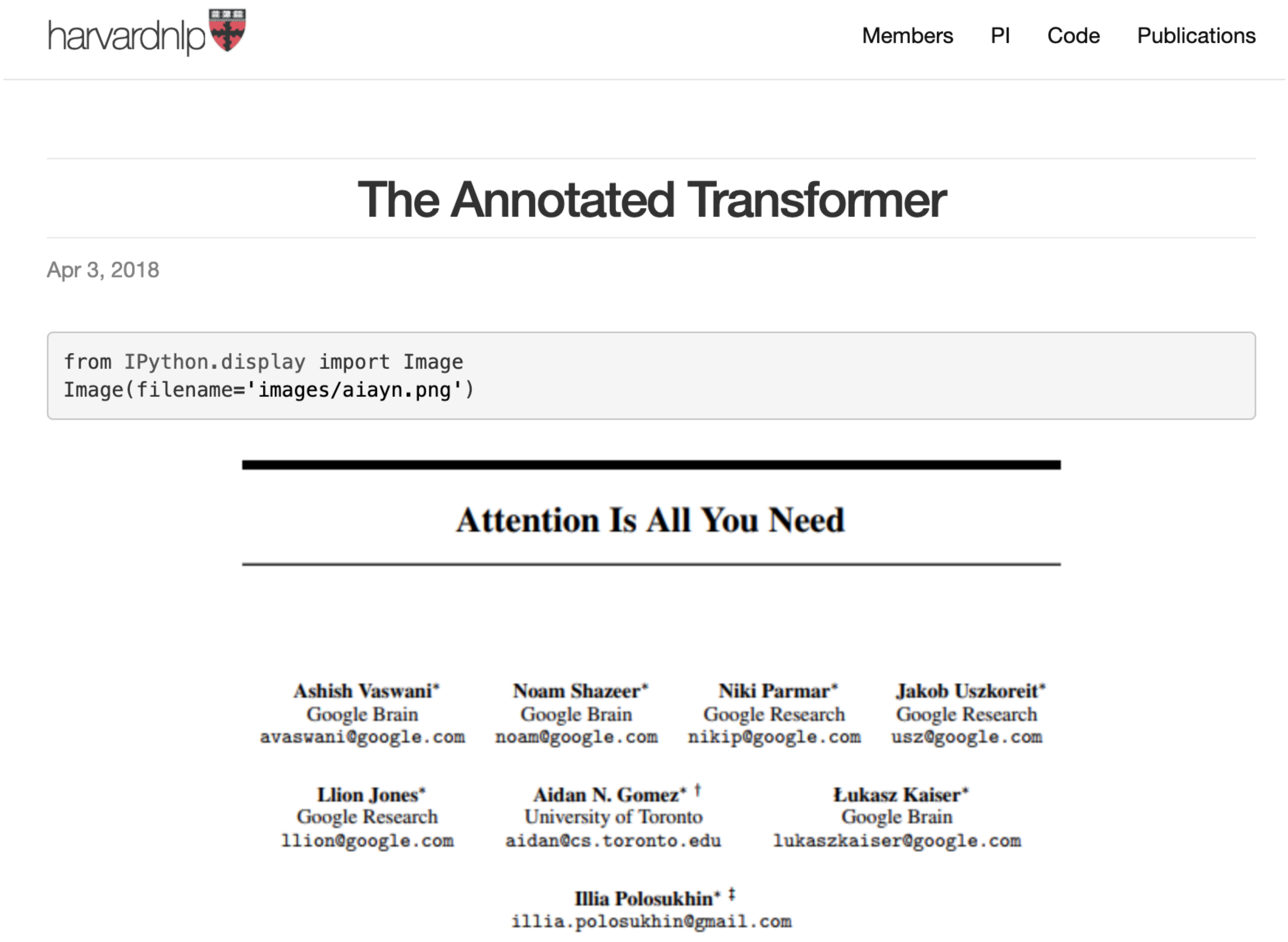
Visualization



Vaswani et al. (2017)

Transformer Implementations

https://nlp.seas.harvard.edu/2018/04/03/attention.html



The Annotated Transformer

Apr 3, 2018

```
from IPython.display import Image
Image(filename='images/aiayn.png')
```

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

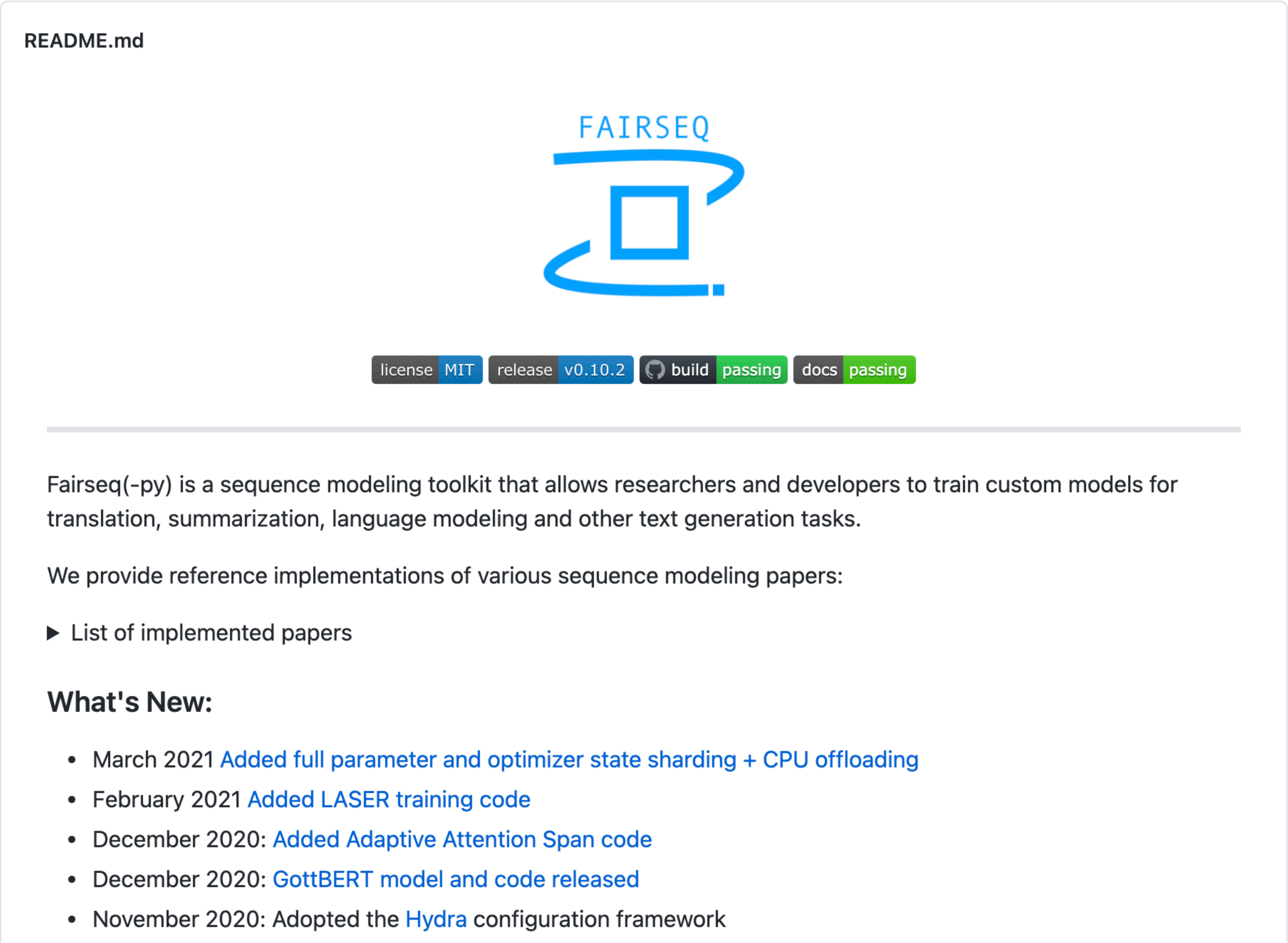
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Transformer Implementations

https://github.com/pytorch/fairseq



README.md

Fairseq(-py) is a sequence modeling toolkit that allows researchers and developers to train custom models for translation, summarization, language modeling and other text generation tasks.

We provide reference implementations of various sequence modeling papers:

▶ List of implemented papers

What's New:

- March 2021 Added full parameter and optimizer state sharding + CPU offloading
- February 2021 Added LASER training code
- December 2020: Added Adaptive Attention Span code
- December 2020: GottBERT model and code released
- November 2020: Adopted the Hydra configuration framework

Takeaways

- Can build MT systems with LSTM encoder-decoders, CNNs, or transformers

- Word piece / byte pair models are really effective and easy to use

- State of the art systems are getting pretty good, but lots of challenges remain, especially for low-resource settings

- Next time: pre-trained transformer models (BERT), applied to other tasks