Word vector representations

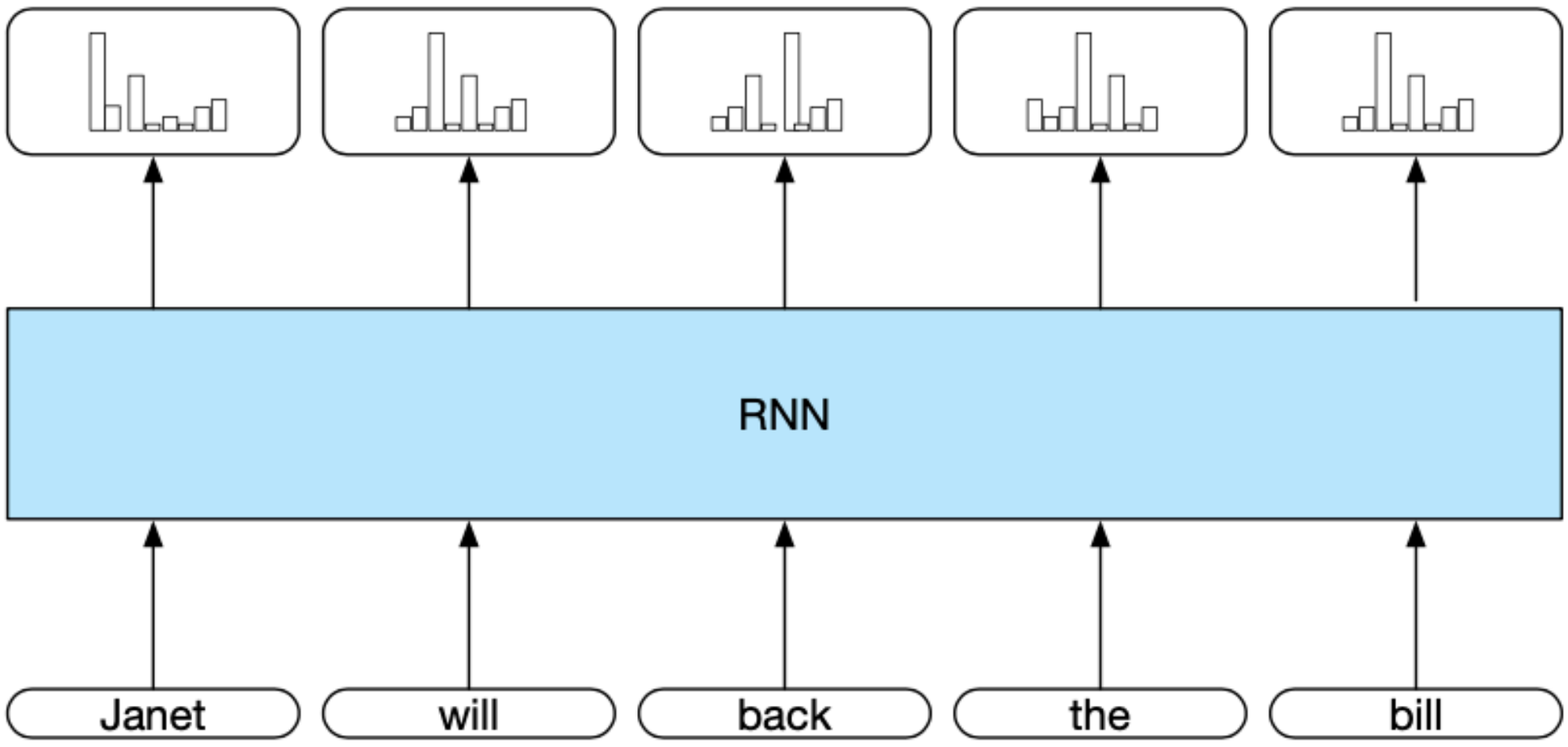
- Before, we had a single representation for each word type:
  - The snow cushioned the **fall.**
  - My favorite season is **fall**.
- Can we construct a representation of a specific token that captures the effect of context on its meaning?

Use case example: named entity recognition

Italy GPE 's business world was rocked by the announcement last Thursday DATE that Mr. Verdi PERSON would leave his job as vice president of Music Masters of Milan, Inc. ORG to become operations director of Arthur Andersen ORG .

- Seattle is in **Washington**.
- **Washington** was the first president of the United States.

Supervised sequence tagging approach using a corpus annotated for named entities



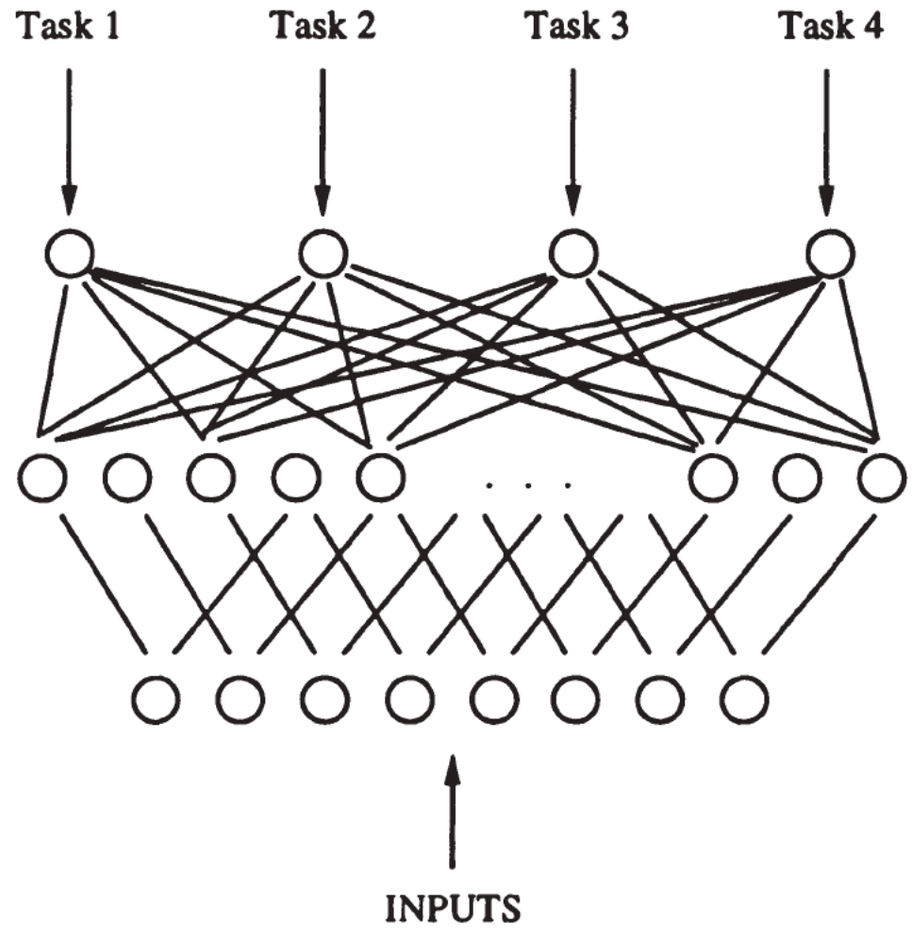
NER corpora are small - hard to learn to produce good token representations!

Getting different machine learning tasks to help each other

- Transfer learning: using representations from one (data-rich) task to bootstrap learning for a data-poor task

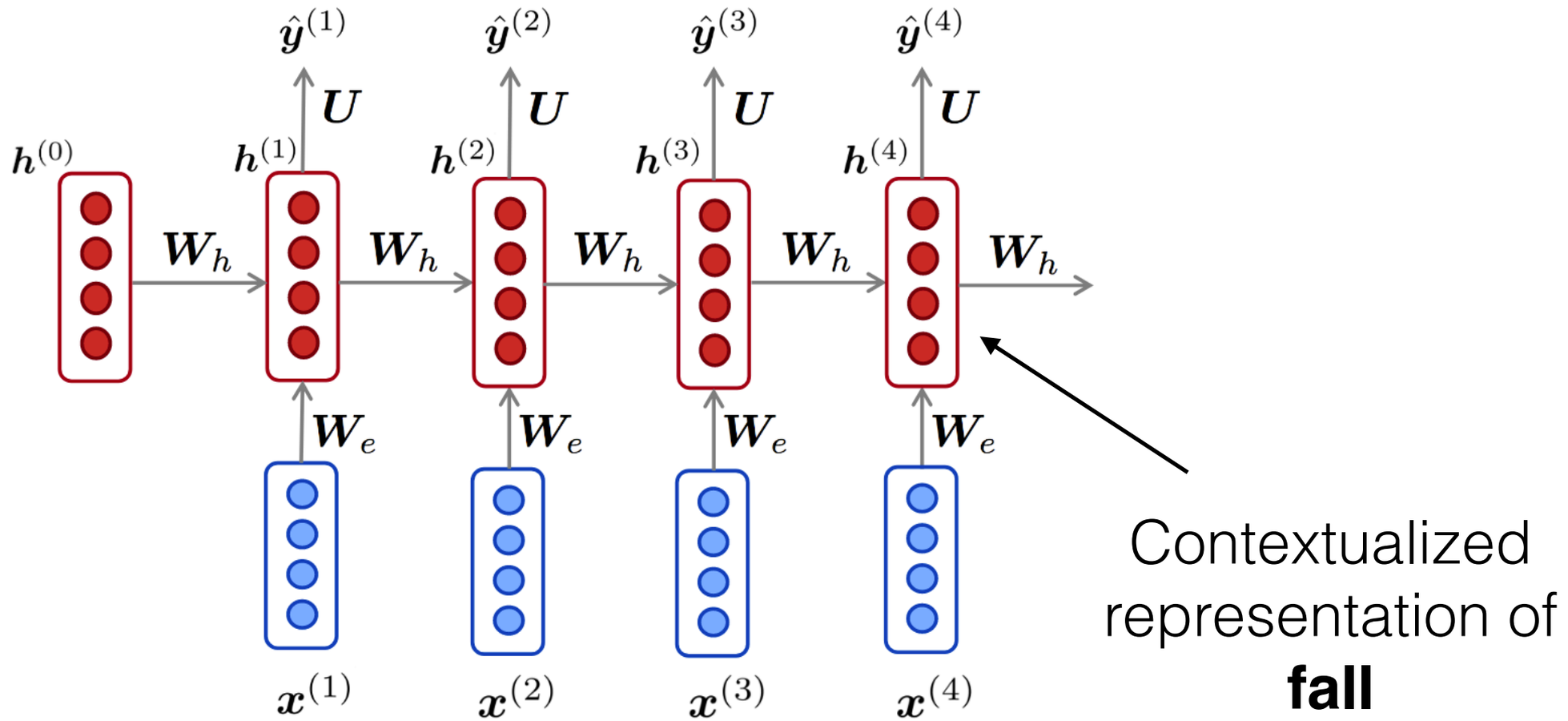
- Multi-task learning: representations that are useful for multiple tasks at once

- Meta-learning: based on our experience learning to perform past tasks, how can we quickly learn future tasks?



Multitask learning (Caruana, 1998)

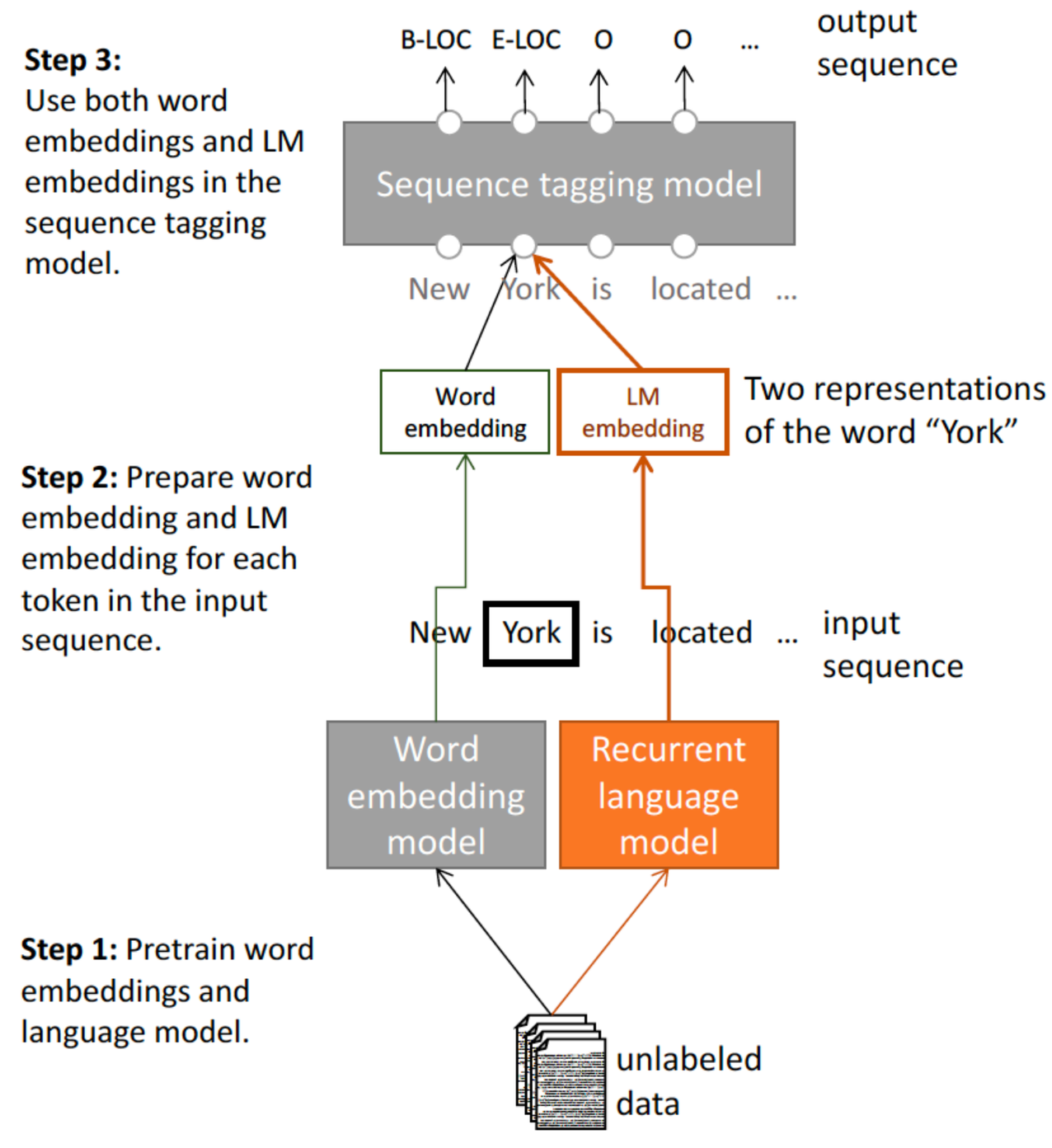
RNN language model states as token representations



We can train RNN LMs on billions of words

Transfer learning

Step 3:
Use both word embeddings and LM embeddings in the sequence tagging model.



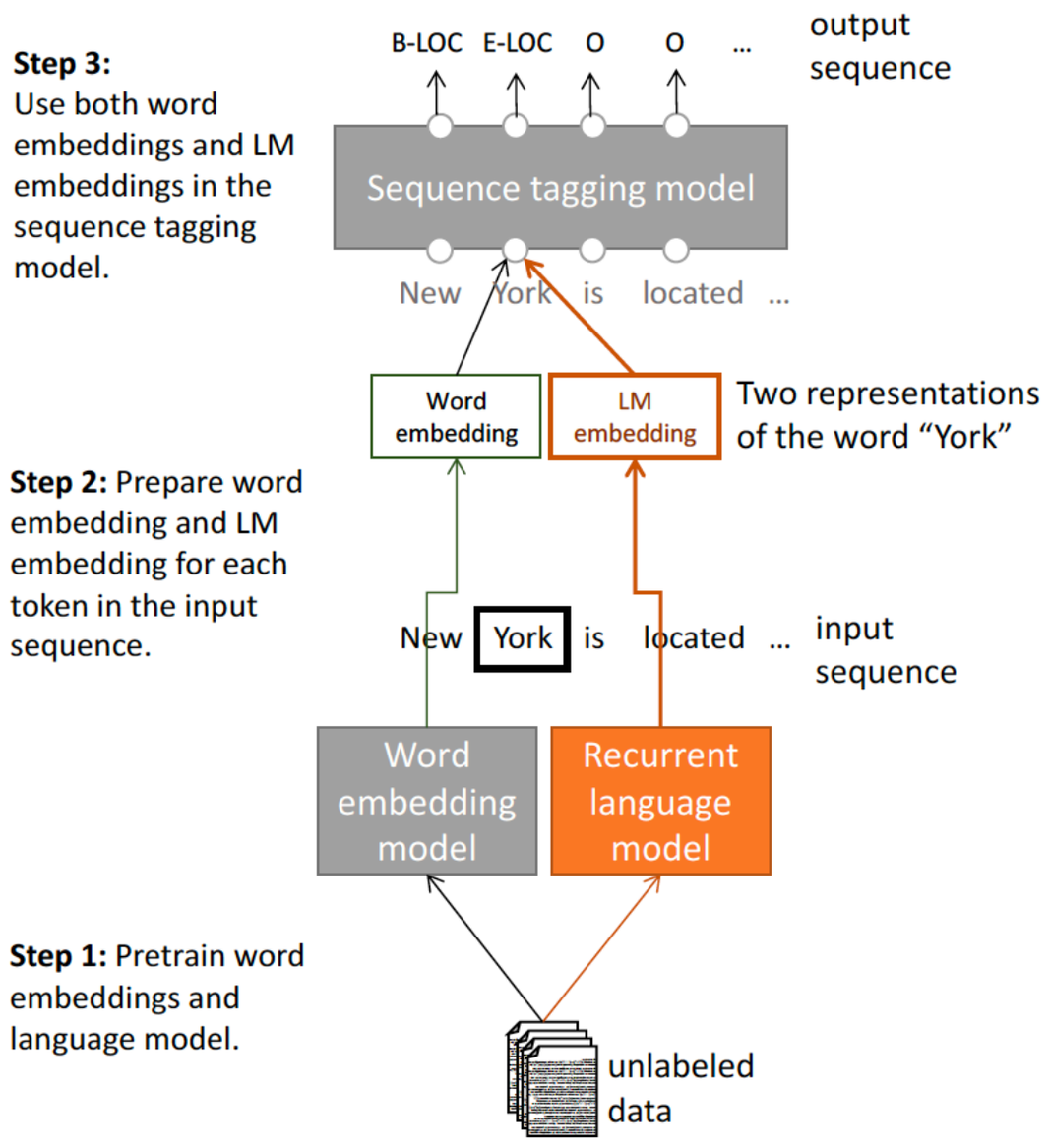
Step 2: Prepare word embedding and LM embedding for each token in the input sequence.

Step 1: Pretrain word embeddings and language model.

(Peters et al. 2017)

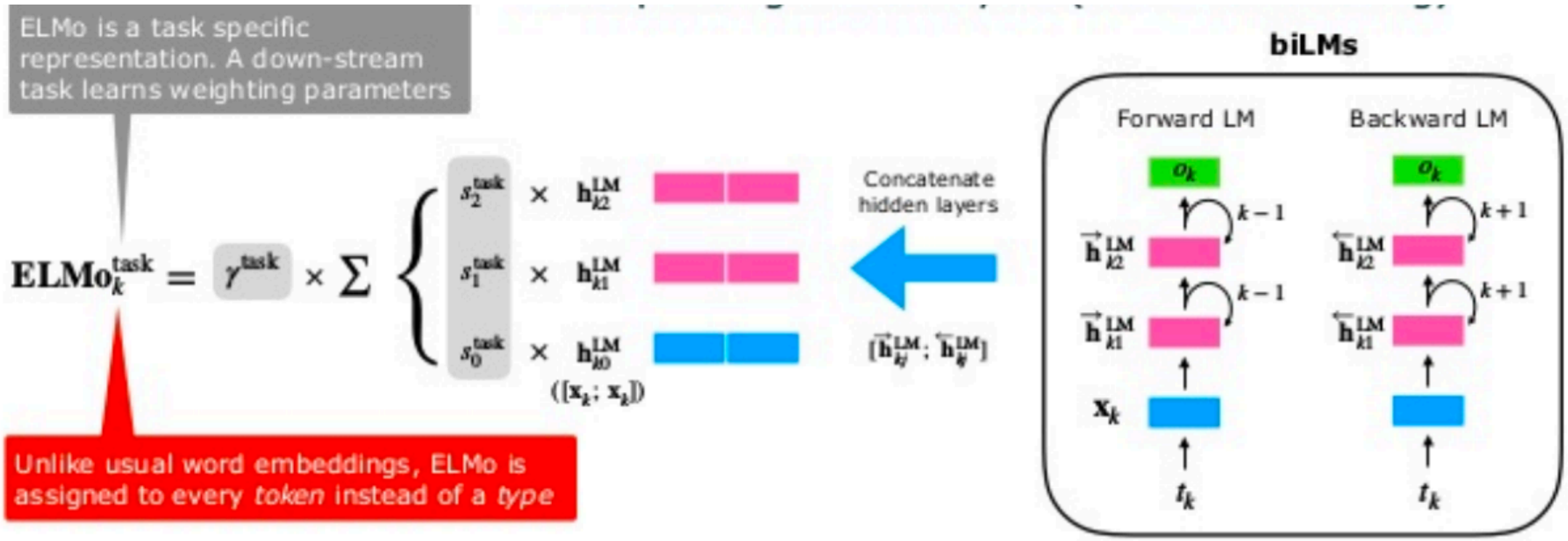
Transfer learning terminology

- Pre-training
- Fine-tuning a classifier
- Continue to train LM weights (backpropagate loss from classification)?



ELMo

Different downstream tasks can assign different weights to each of the BiLSTM's layers: fine-tuning is just learning a linear combination



(Peters et al, 2018)

Type vs. token embeddings

	Source	Nearest Neighbors
GloVe	play	playing, game, games, played, players, plays, player, Play, football, multiplayer
biLM	Chico Ruiz made a spectacular play on Alusik 's grounder {...}	Kieffer , the only junior in the group , was commended for his ability to hit in the clutch , as well as his all-round excellent play .
	Olivia De Havilland signed to do a Broadway play for Garson {...}	{...} they were actors who had been handed fat roles in a successful play , and had talent enough to fill the roles competently , with nice understatement .

(Peters et al, 2018)

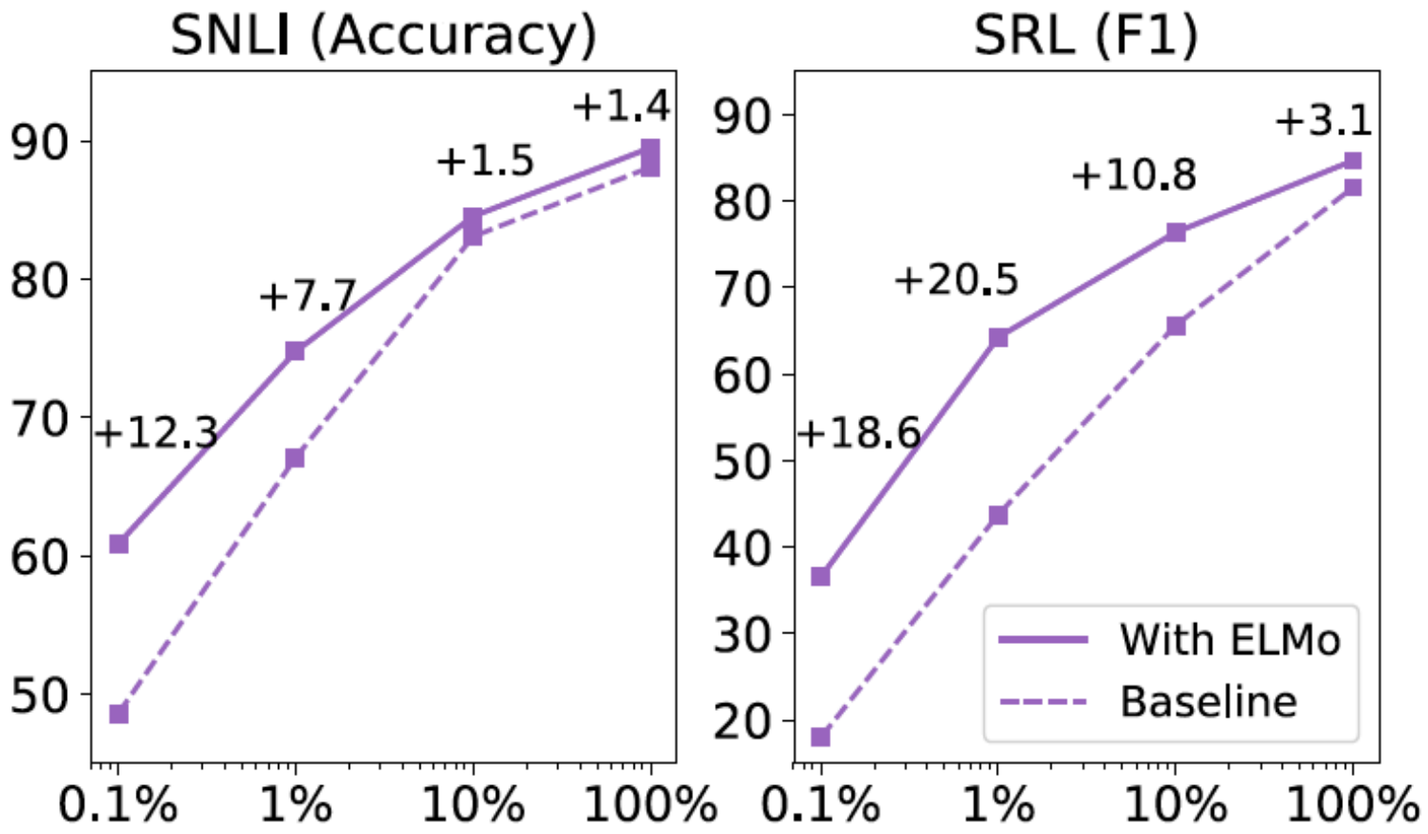
ELMo

Adding ELMo token embeddings to existing neural systems improves performance across a range of tasks (and neural architectures)

Task	Previous SOTA		Our baseline	ELMo + baseline	Increase (absolute/ relative)
SQuAD	Liu et al. (2017)	84.4	81.1	85.8	4.7 / 24.9%
SNLI	Chen et al. (2017)	88.6	88.0	88.7 ± 0.17	0.7 / 5.8%
SRL	He et al. (2017)	81.7	81.4	84.6	3.2 / 17.2%
Coref	Lee et al. (2017)	67.2	67.2	70.4	3.2 / 9.8%
NER	Peters et al. (2017)	91.93 ± 0.19	90.15	92.22 ± 0.10	2.06 / 21%
SST-5	McCann et al. (2017)	53.7	51.4	54.7 ± 0.5	3.3 / 6.8%

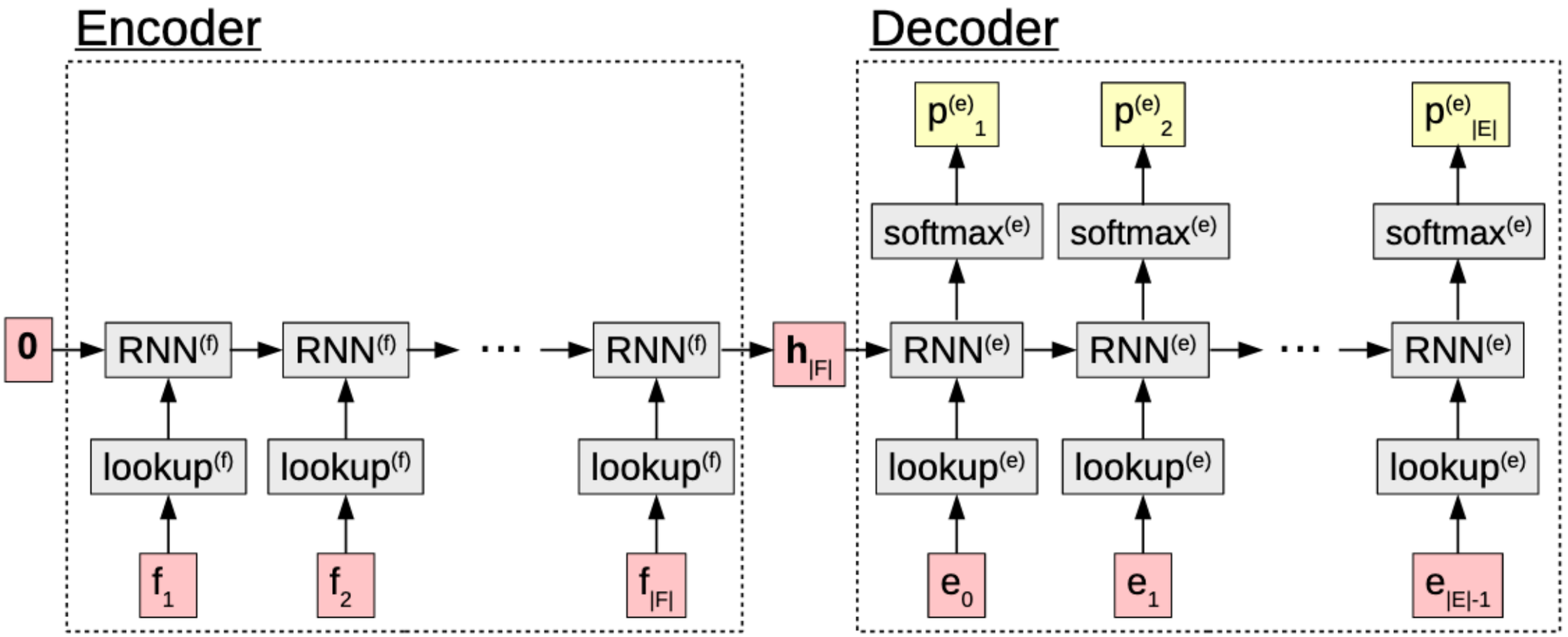
- First layer (slightly) better than second for POS tagging
- Second layer (slightly) better than first for word sense disambiguation

Pretraining improves sample efficiency



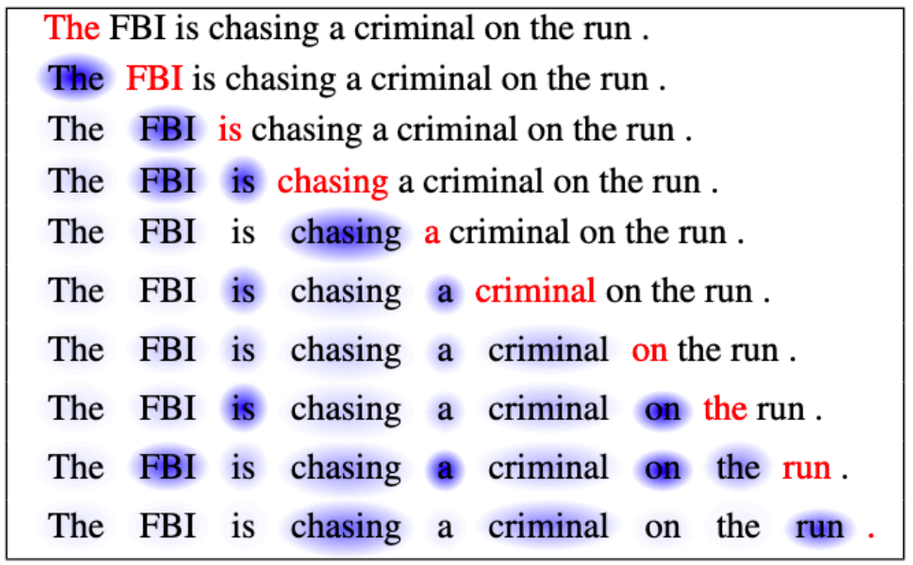
(Peters et al, 2018)

Encoder/decoder RNN



- Attention removes the bottleneck: the decoder uses a weighted average of the encoder's hidden states, instead of just the last one

LSTM with self-attention

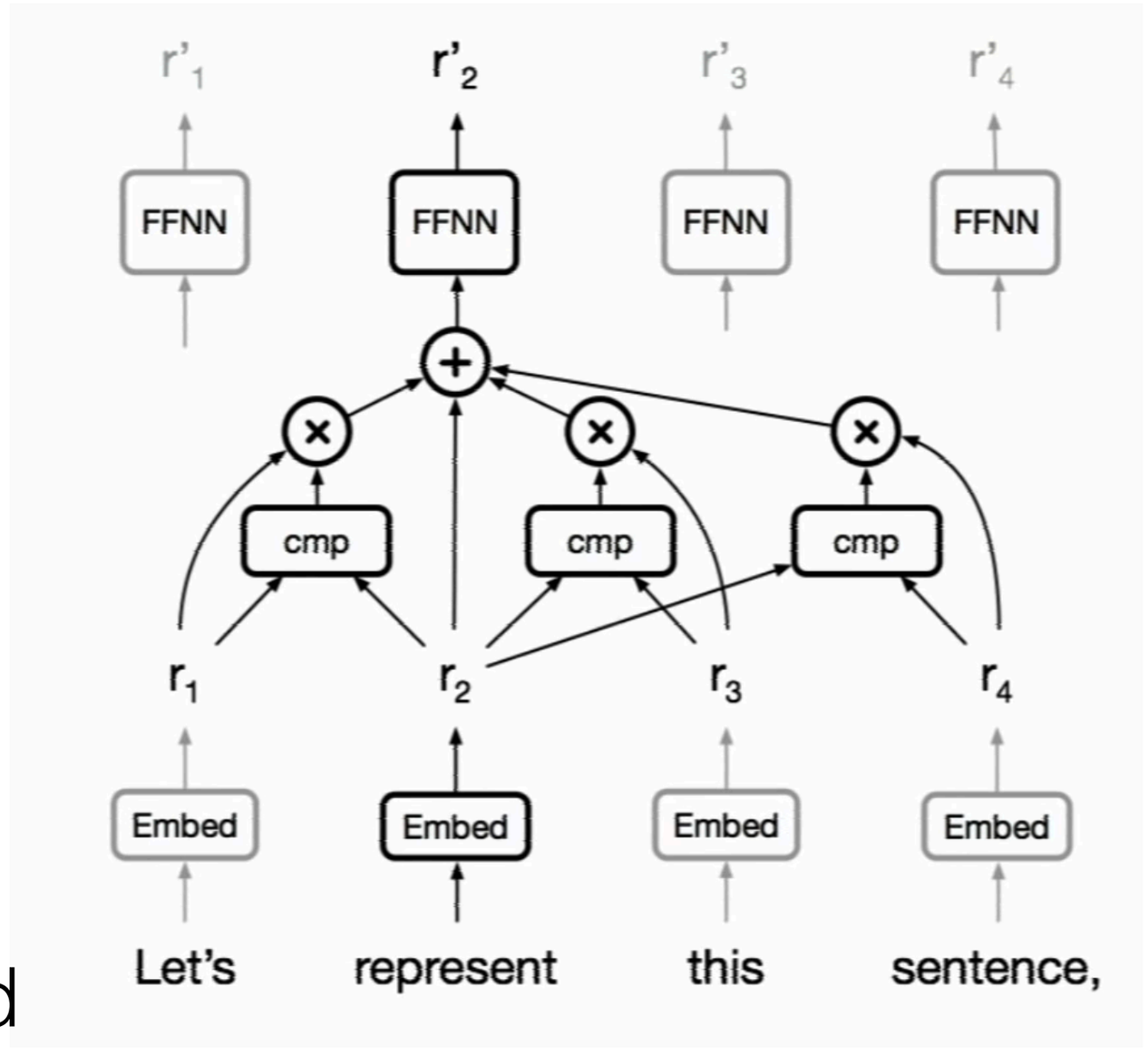


(Cheng et al. 2016)

- Can we can do away with the LSTM (recurrence) and just operate over contextualized word representations?

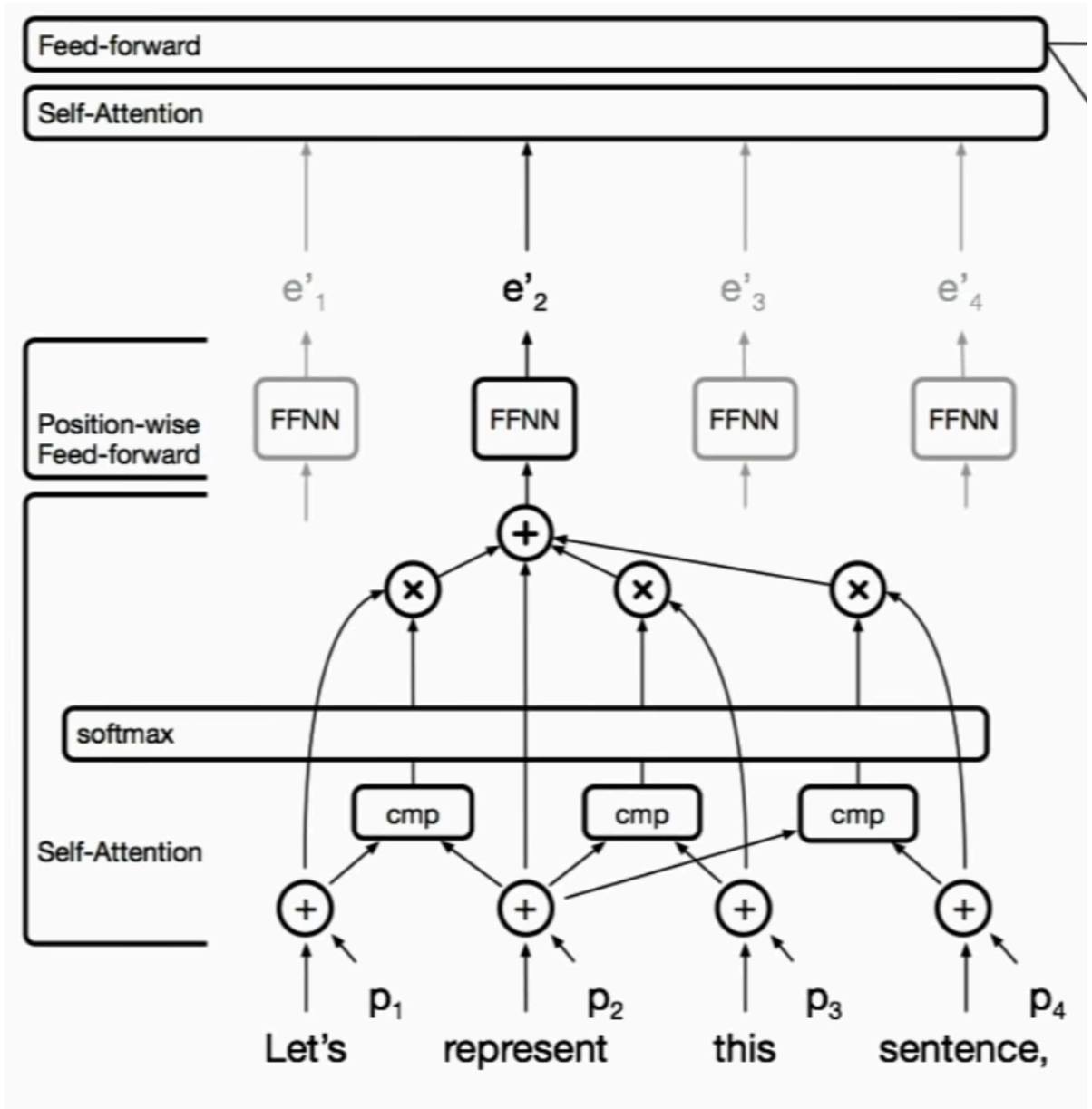
Transformer self-attention layer

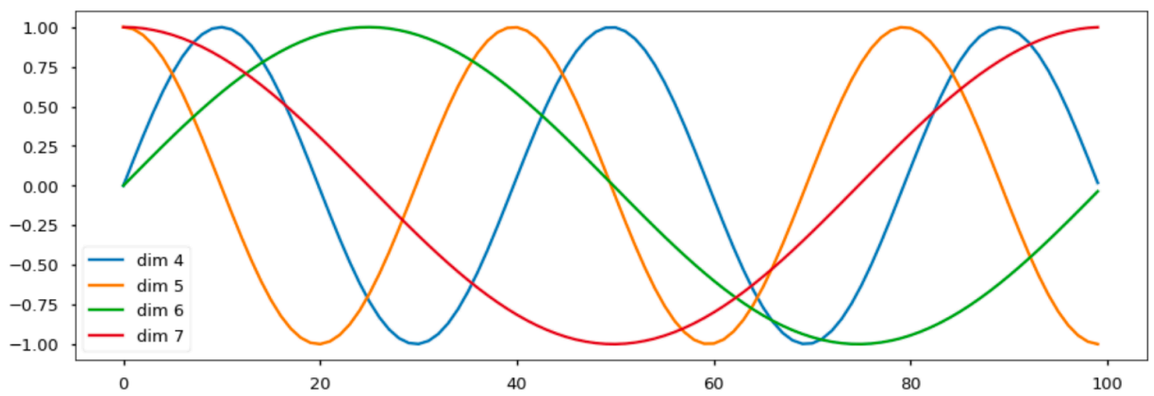
- Attend to both earlier and later words
- No hidden state: only contextual word representations
- Unlike RNNs, no recency bias: most recent words and word 20 steps earlier can affect the current



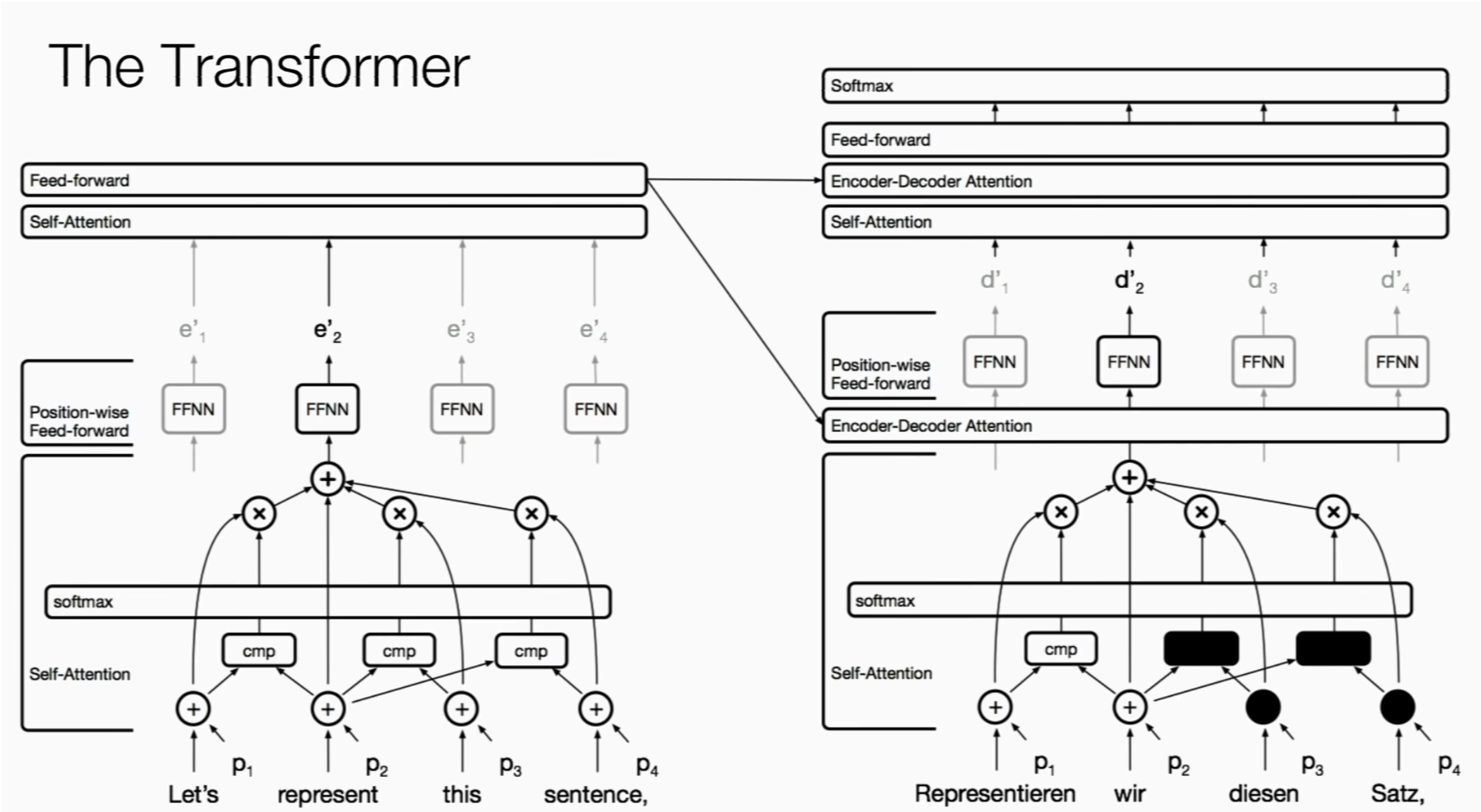
(Credit: Ashish Vaswani)

Positional embeddings





Transformer encoder-decoder



Dot-product attention in an RNN (reminder)

$$s_i = f(s_{i-1}, y_{i-1}, c_i)$$

$$c_t = \sum_{j=1}^{T(x)} \alpha_{ij} h_j$$

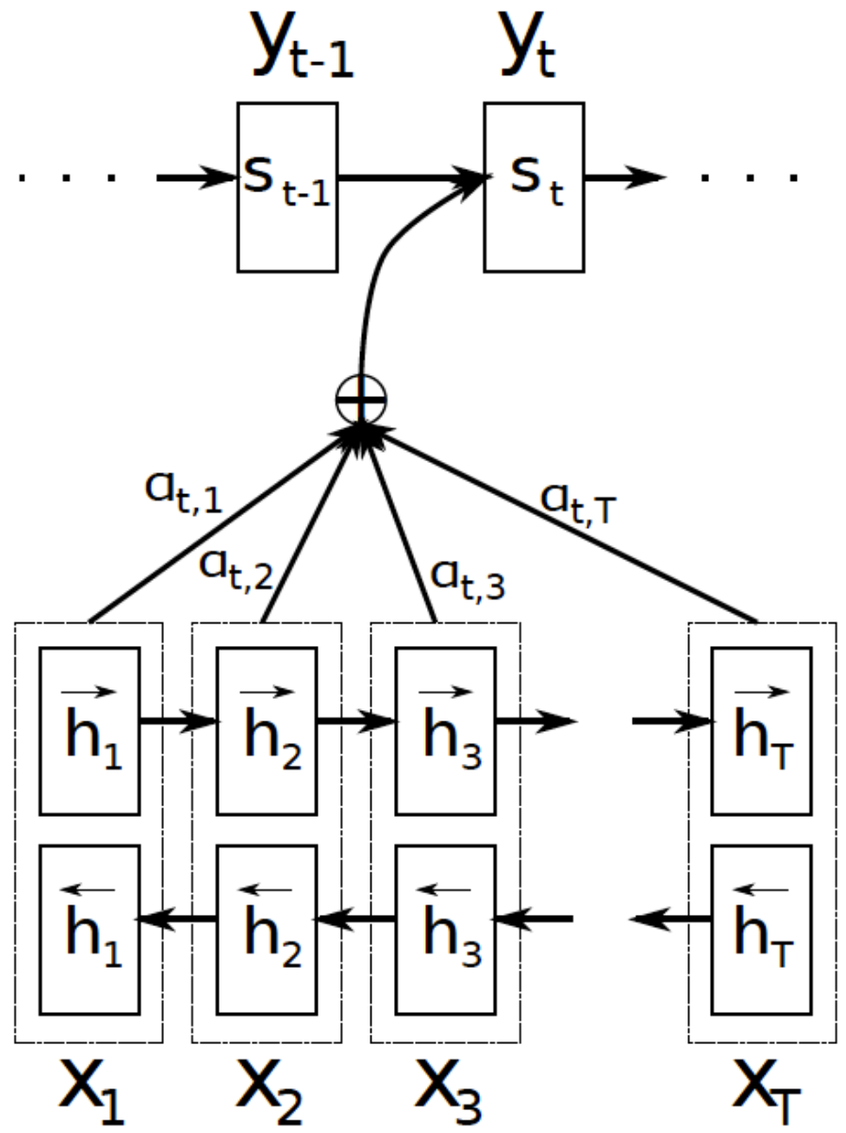
Value

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_j \exp(e_{ij})}$$

$$e_{ij} = s_{i-1}^T h_j$$

Query

Key



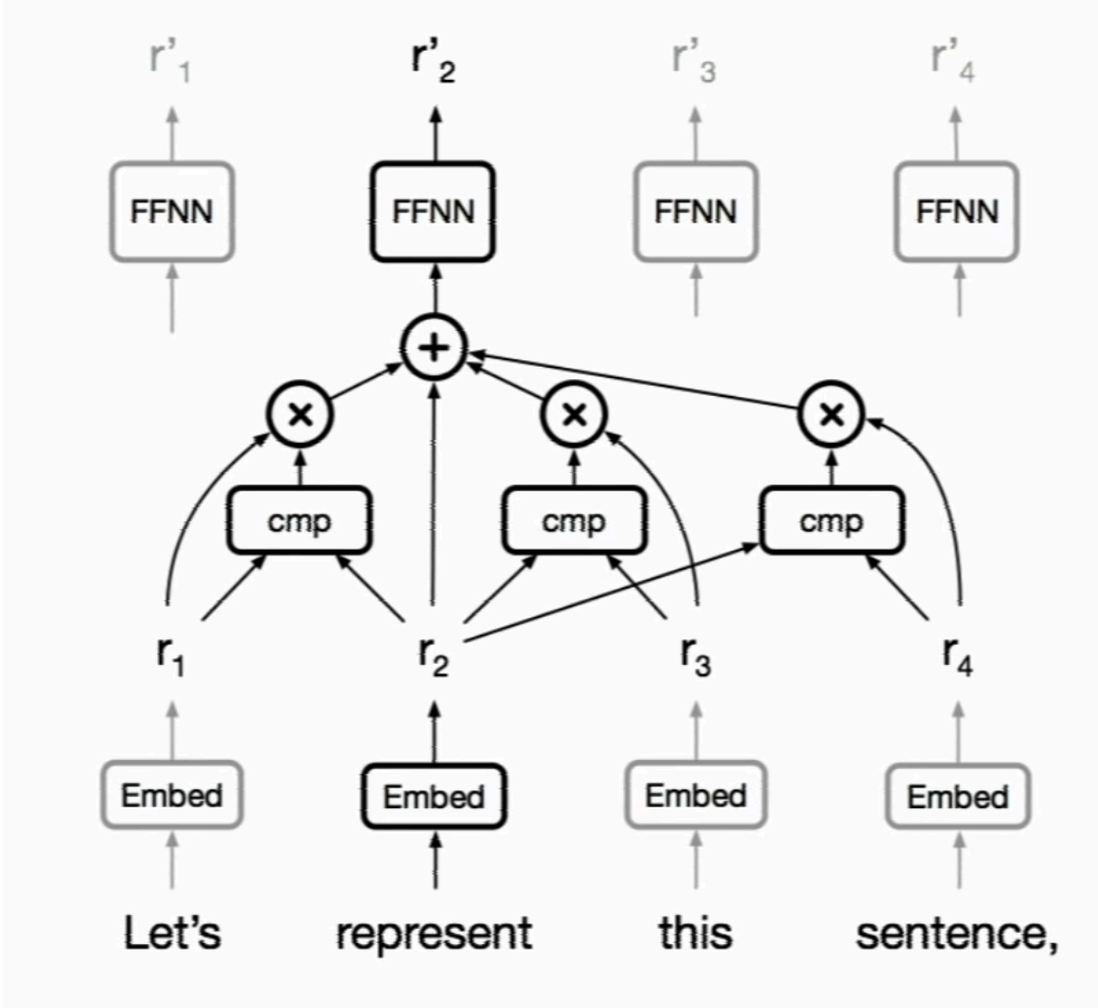
(Bahdanau et al., 2015)

Encoder self-attention

- The word that we're currently computing the next representation for is the **query**

- All other words are **keys** (for computing attention weights) and **values** (for the weighted average)

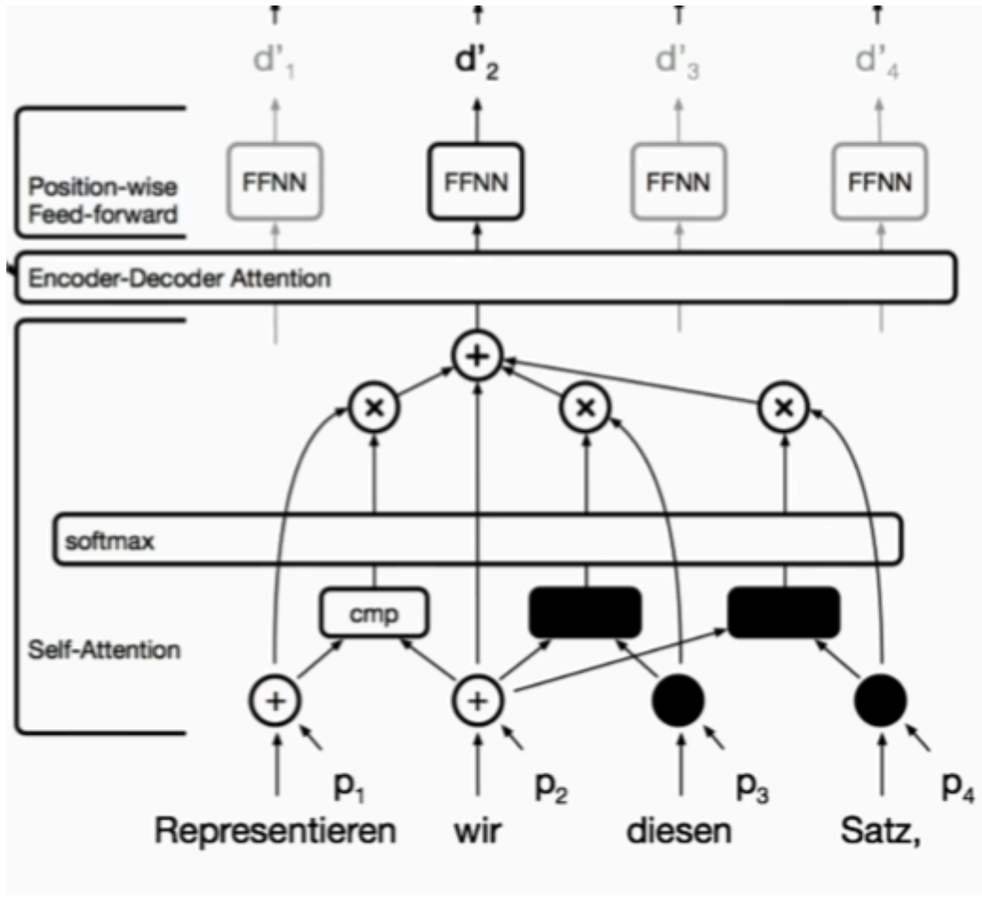
- Efficiently parallelized using matrix multiplication:



$$\text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Decoder self-attention

- The word that we're currently computing the next representation for is the query
- All other words are keys (for computing attention weights) and values (for the weighted average)
- Future words are "masked" so that efficient matrix multiplication is still possible

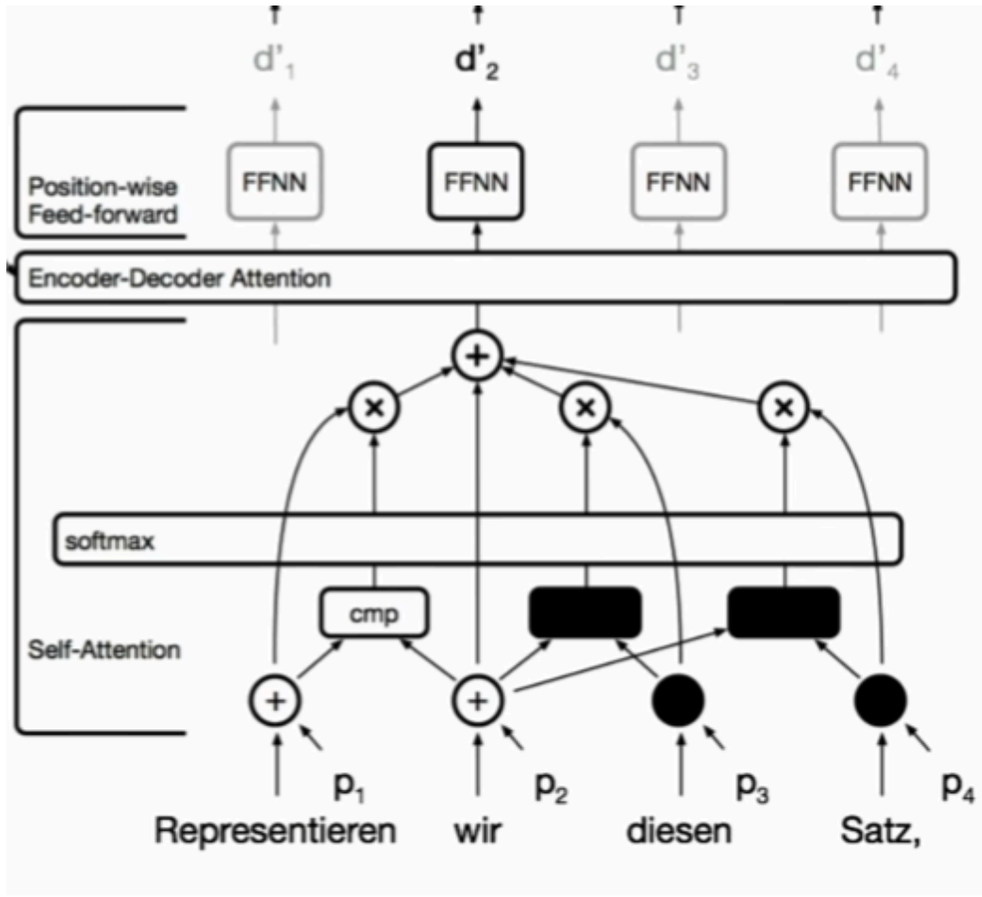


$$\text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

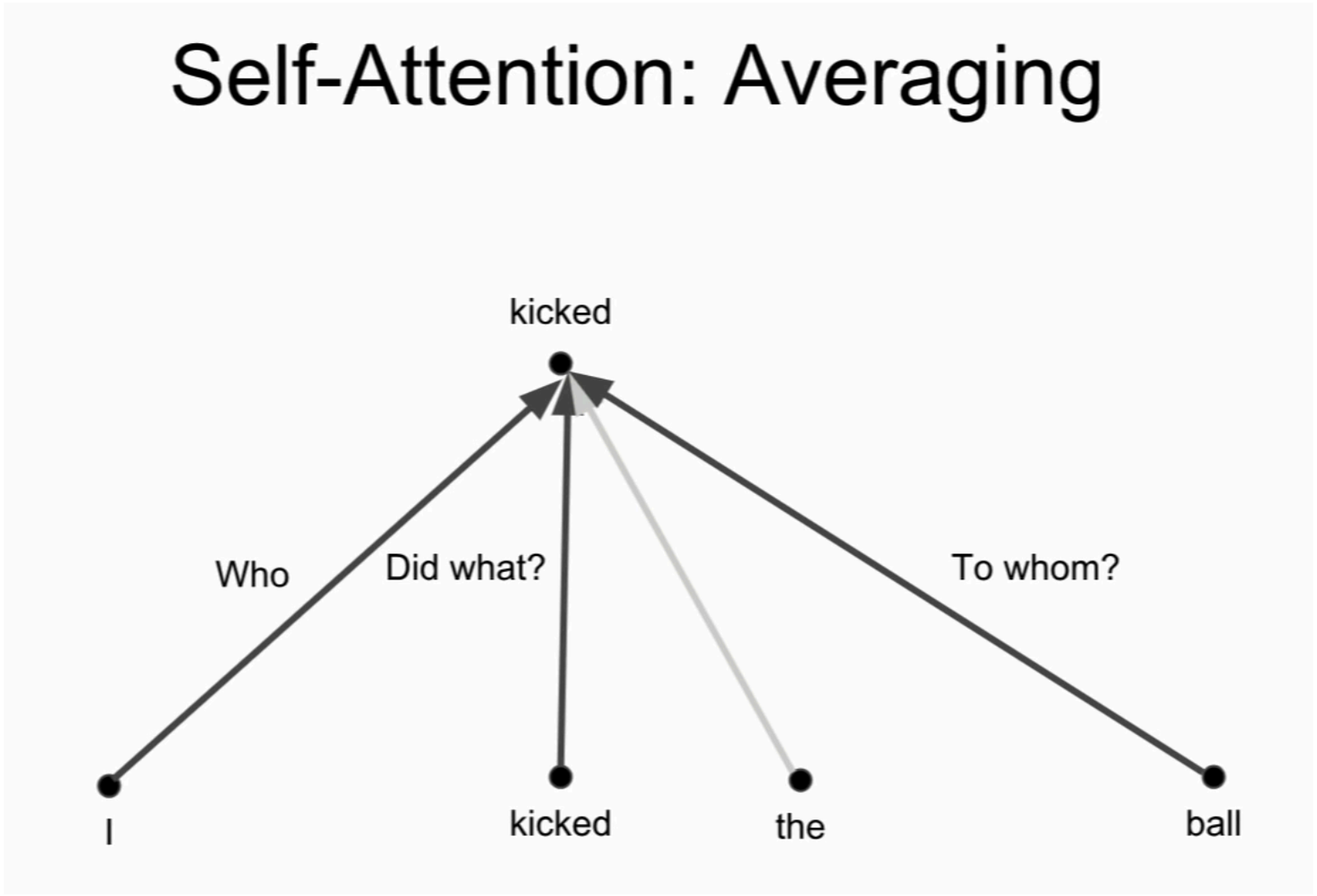
Encoder-decoder attention

- The word that we're currently computing the next representation for (in the decoder) is still the query

- The keys and values are the **encoder's** representations of the words

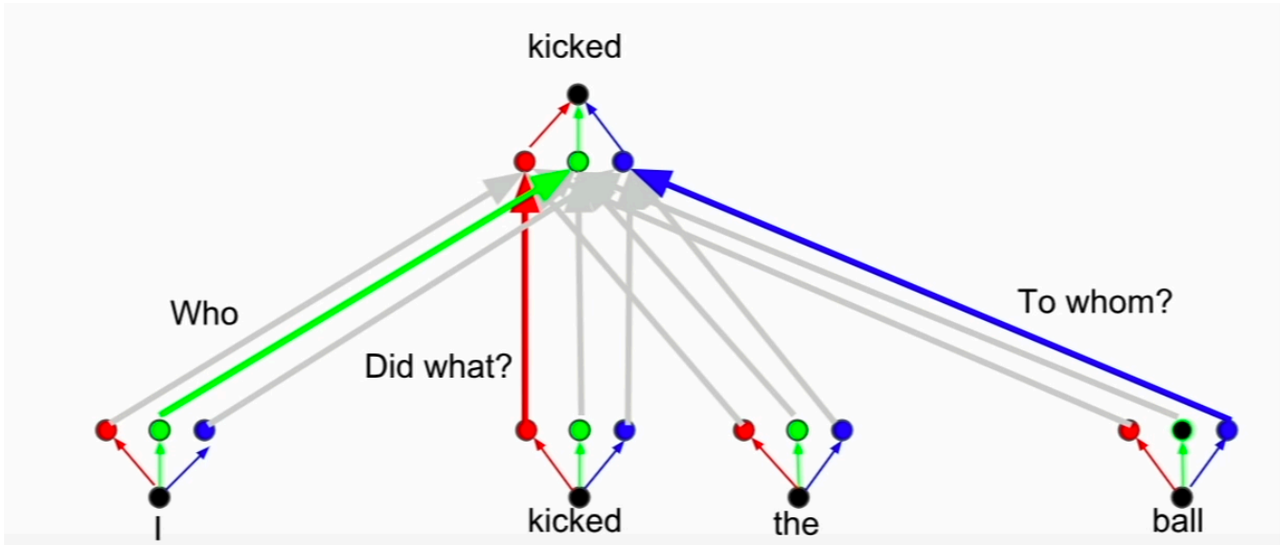


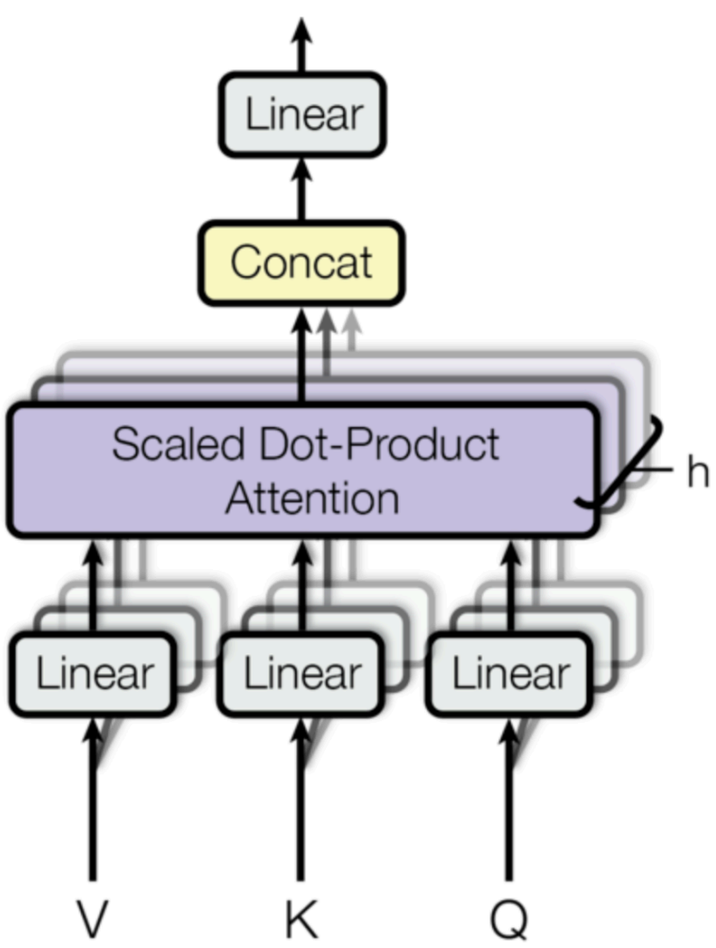
Attention treats all relations alike



Multi-head attention

Extract different aspects of each token representation, for different purposes





$$\text{attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

$$\text{concat}(\text{head}_1, \ldots, \text{head}_h)$$

$$\text{head}_i = \text{attention}(QW_i^Q, KW_i^K, VW_i^V)$$

(W_i^Q projects Q into d_k dimensions, etc:
If embedding has 256 dimensions,
and $h = 8$, then $d_k = 32$)

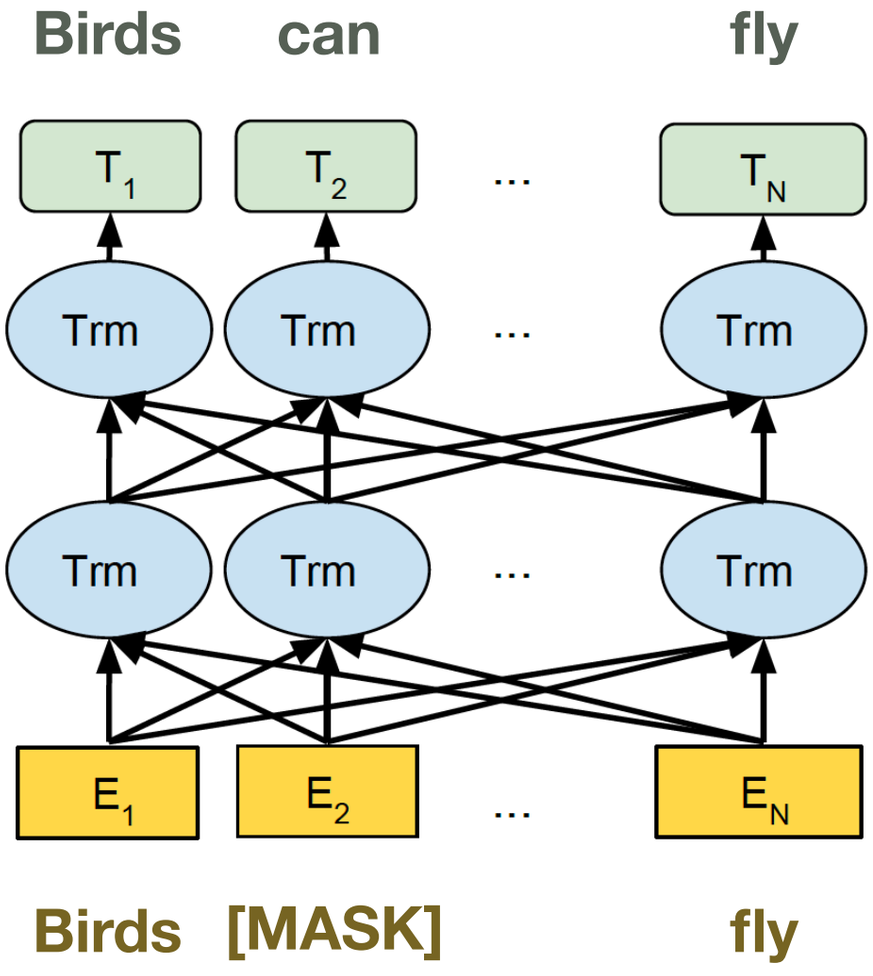
Transformers for machine translation

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [15]	23.75			
Deep-Att + PosUnk [32]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [31]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [8]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [26]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [32]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [31]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [8]	26.36	**41.29**	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	**$3.3 \cdot 10^{18}$**	
Transformer (big)	**28.4**	**41.0**	$2.3 \cdot 10^{19}$	

(Vaswani et al. 2017)

BERT: masked language modeling

- No encoder/decoder separation
- MLM loss: predict some fraction of input words; these words are usually masked (“Cloze task”), sometimes unchanged, sometimes replaced with random other word
- BERT-base (L=12, H=768, A=12, Total Parameters=110M)
- BERT-large (L=24, H=1024, A=16, Total Parameters=340M)



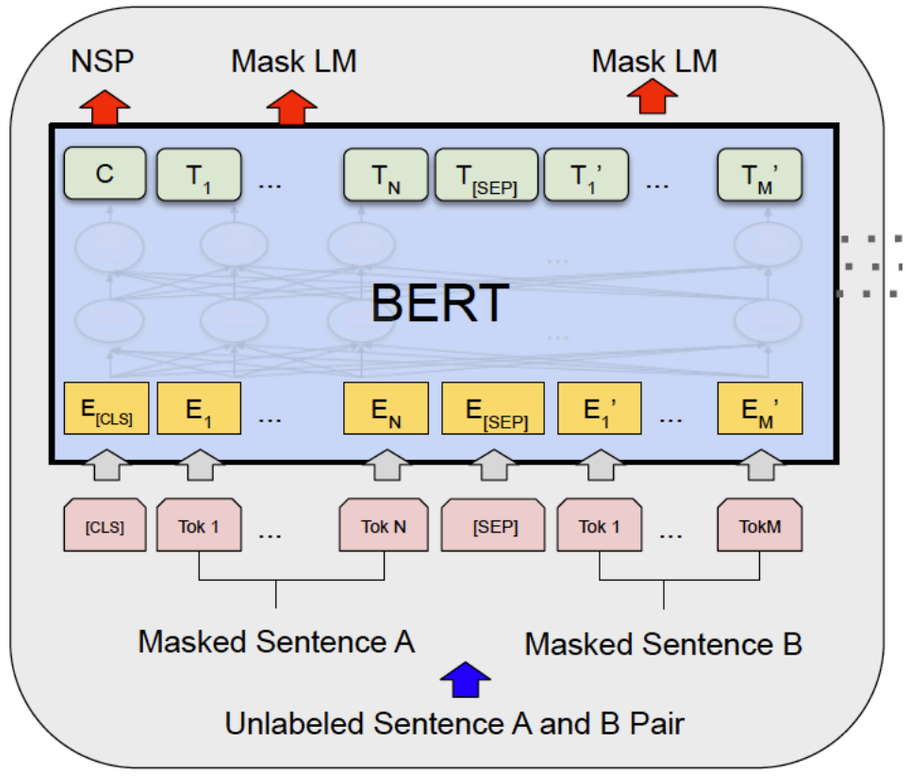
(Devlin et al., 2019)

BERT

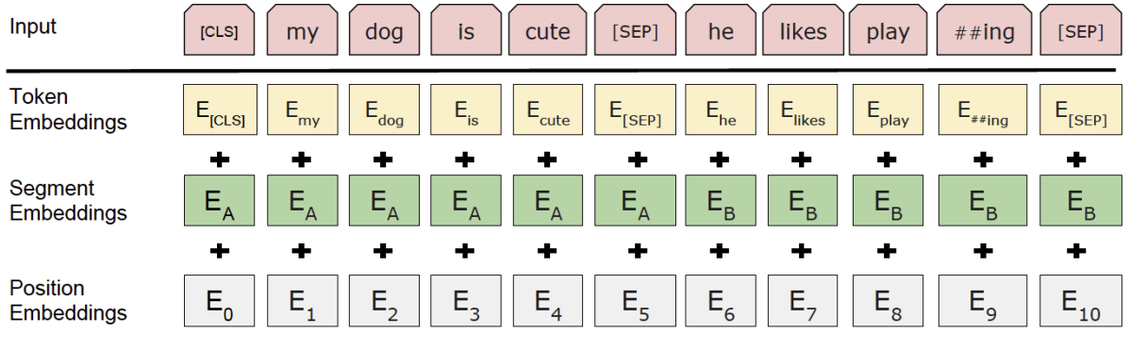
- Next sentence prediction (NSP) loss: are these two (potentially multi sentence) segments contiguous in the original corpus?

- Fine-tuning on sentence-pair or single-sentence tasks: train a classifier with the C vector as input (continue to train BERT)

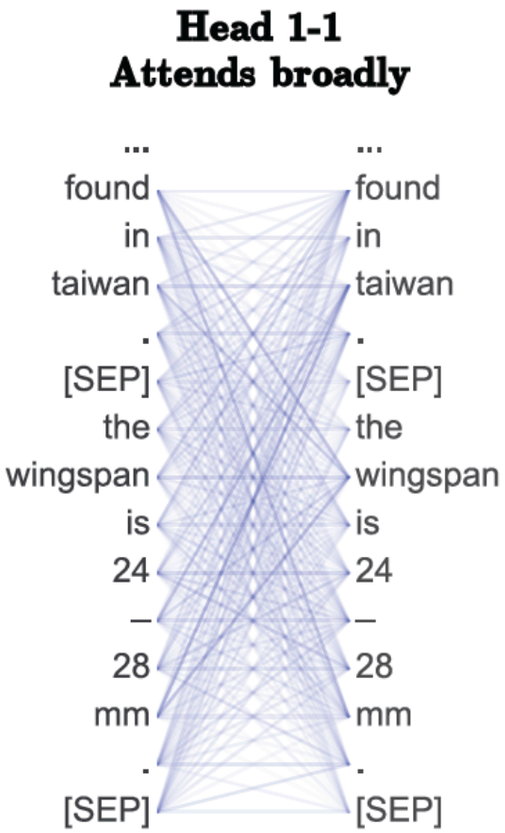
- NSP objective later shown to be ineffective (Liu et al 2019)

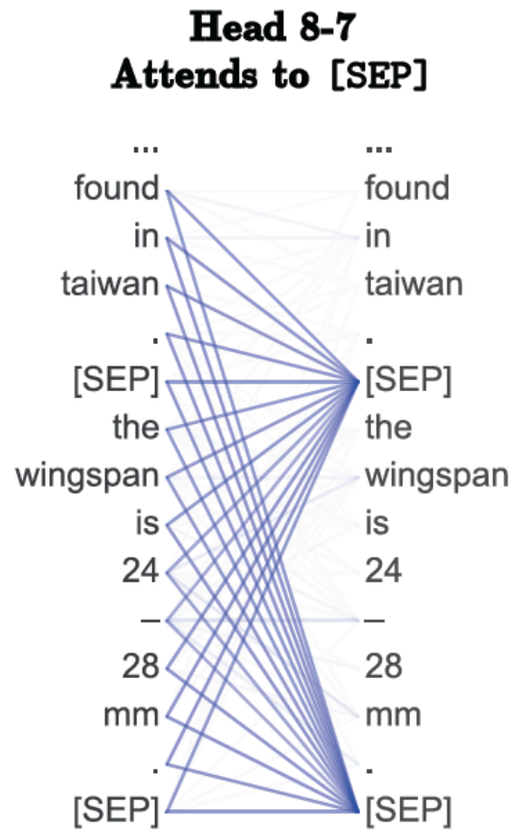


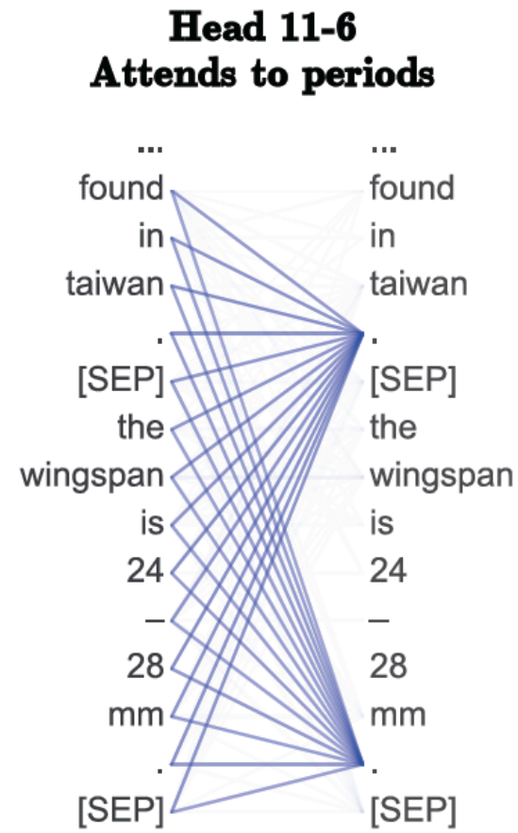
Pre-training



BERT attention heads





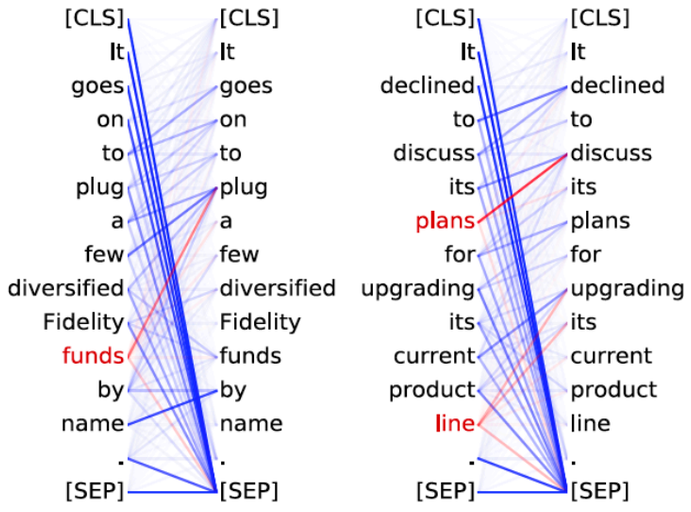


(Clark et al. 2019)

BERT attention heads

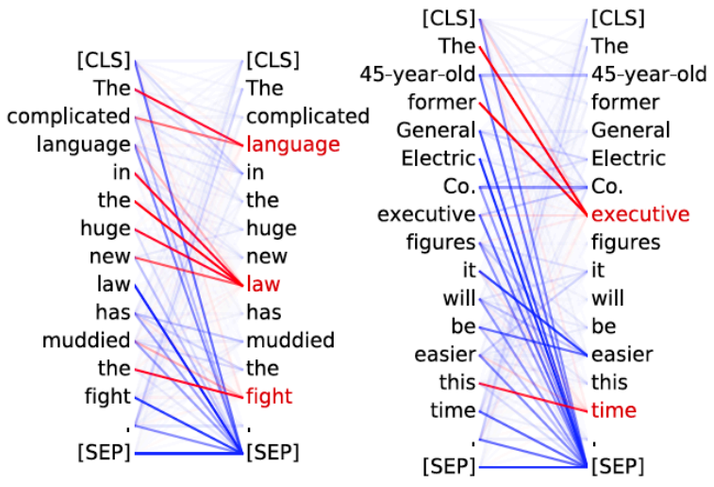
Head 8-10

- **Direct objects** attend to their verbs
- 86.8% accuracy at the `dobj` relation



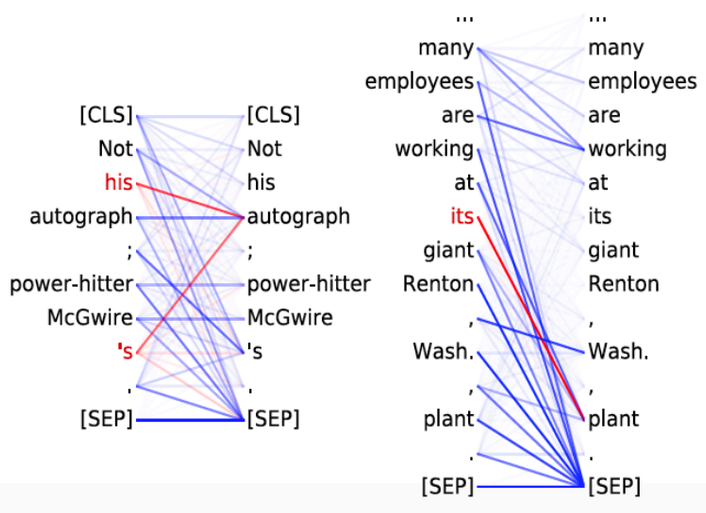
Head 8-11

- **Noun modifiers** (e.g., determiners) attend to their noun
- 94.3% accuracy at the `det` relation



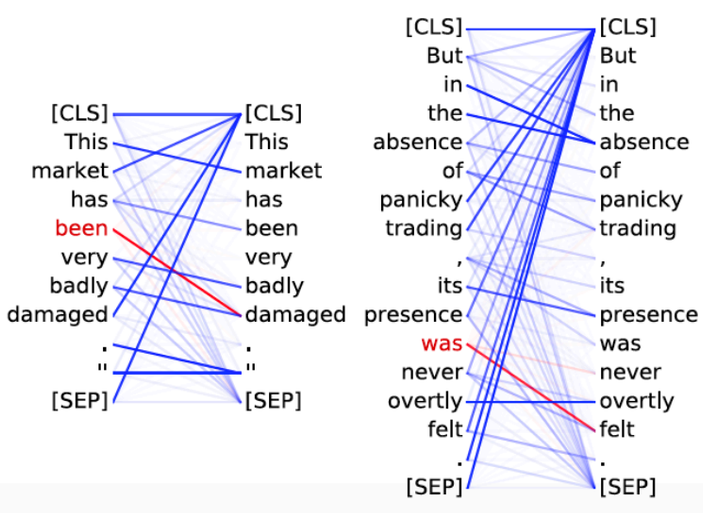
Head 7-6

- **Possessive pronouns** and apostrophes attend to the head of the corresponding NP
- 80.5% accuracy at the `poss` relation



Head 4-10

- **Passive auxiliary verbs** attend to the verb they modify
- 82.5% accuracy at the `auxpass` relation

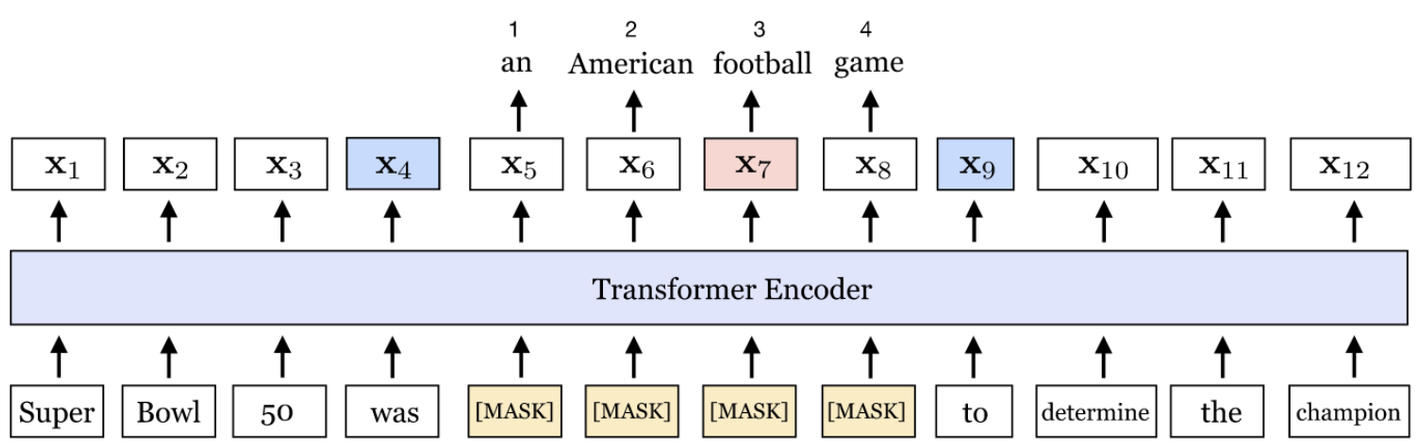


(Clark et al. 2019)

Encoder variants

- More data, larger batch size, no NSP… (RoBERTa: Liu et al 2019)

- Mask contiguous spans (SpanBERT, Joshi et al 2020)



- Segment order prediction: unlike BERT's NSP, segments always come from the same document (ALBERT, Lan et al 2020)

- Etc, etc (see Xia, Wu & van Durme 2020)

Encoder variants

CamemBERT

A Tasty French Language Model

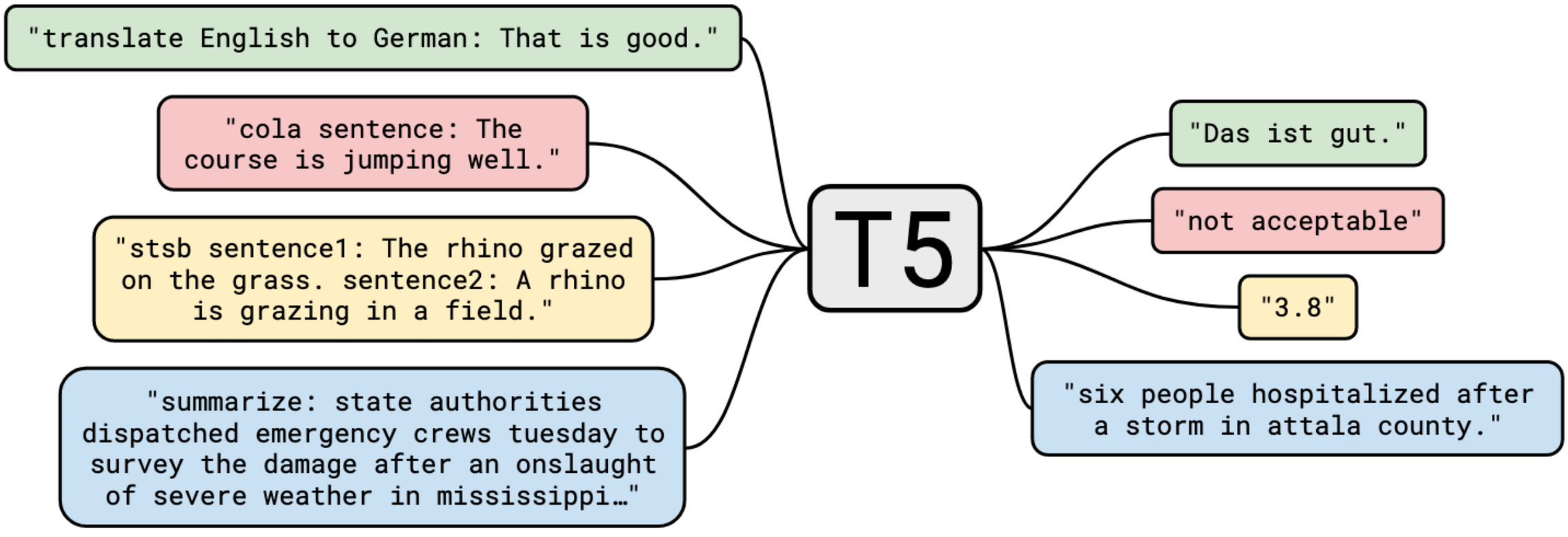
FlauBERT: Unsupervised Language Model Pre-training for French

BERTje: A Dutch BERT Model

AraBERT: Transformer-based Model for Arabic Language Understanding

T5

- Empirical study of many learning objectives, corpus size, architecture variants…



(Raffel et al. 2020)