Transfer Learning and Transformers

Outline

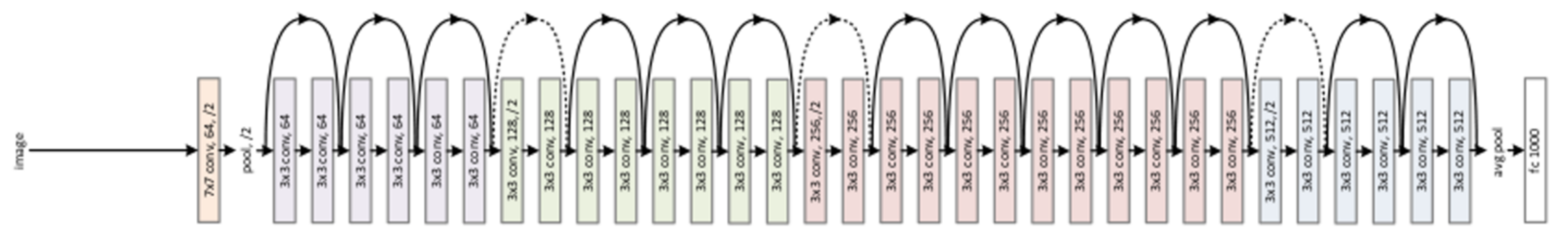
- Transfer Learning in Computer Vision
- Embeddings and Language Models
- "NLP's ImageNet Moment": ELMO/ULMFit
- Transformers
  - Attention in detail
  - BERT, GPT-2, DistillBERT, T5

Outline

- **Transfer Learning in Computer Vision**
- Embeddings and Language Models
- "NLP's ImageNet Moment": ELMO/ULMFit
- Transformers
  - Attention in detail
  - BERT, GPT-2, DistillBERT, T5

Transfer Learning

- Let's classify birds! We have 10K labeled images.
- From ImageNet, we know that deep NN like Resnet-50 works well.



- Problem: NN is so large that it overfits on our small data!
- Solution: use NN trained on ImageNet (1M images), and *fine-tune* on bird data.

- Result: better performance than anything else!

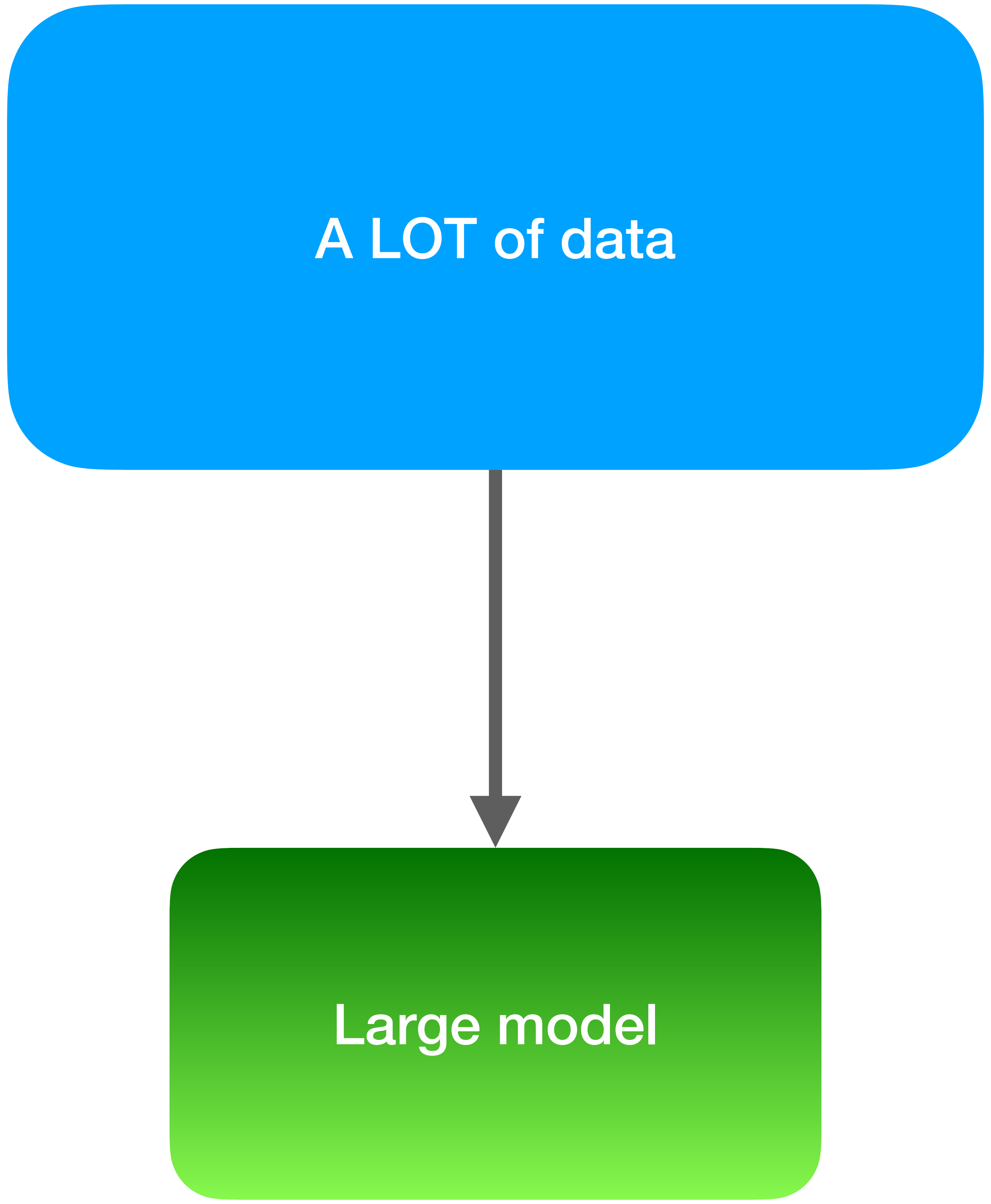
Transfer Learning

A LOT of data

Large model

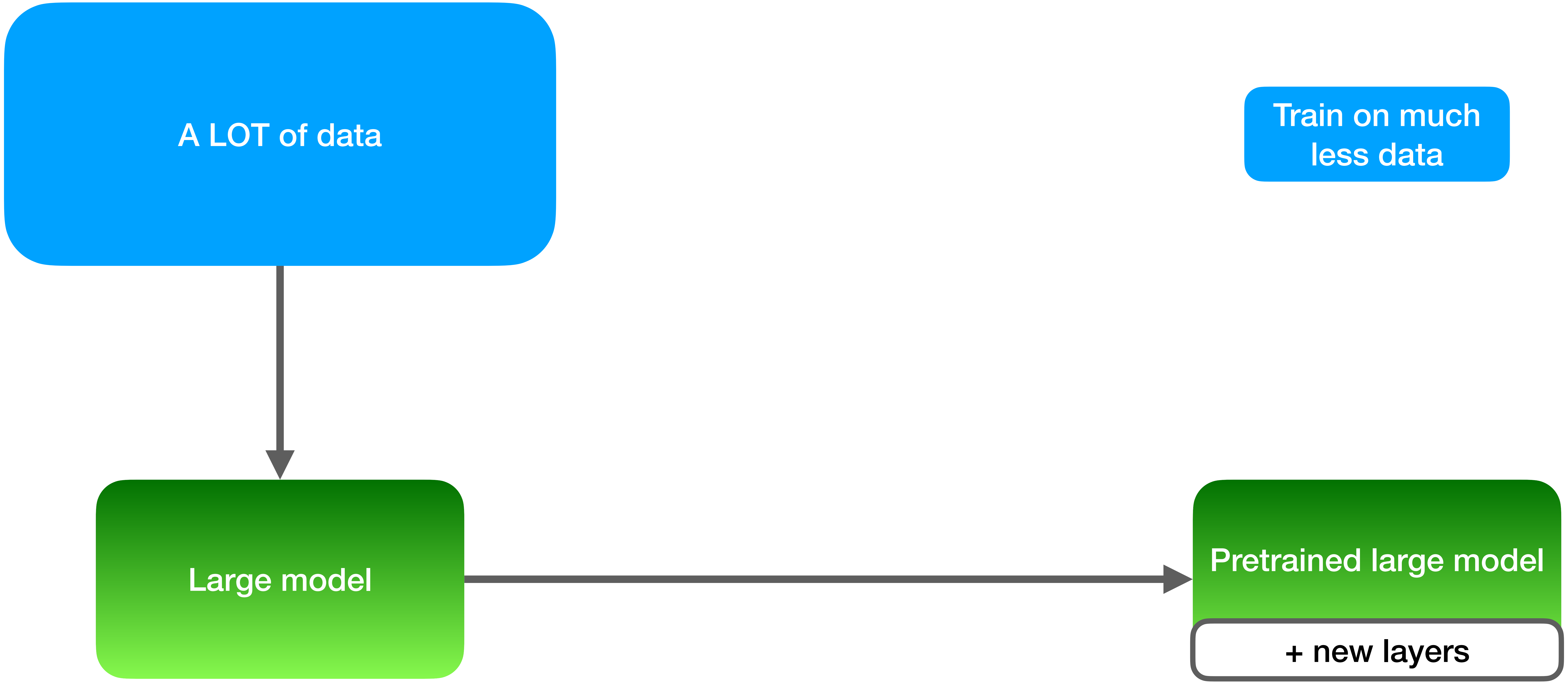
Traditional Machine Learning:

Transfer Learning



Traditional Machine Learning:
slow training on a lot of data

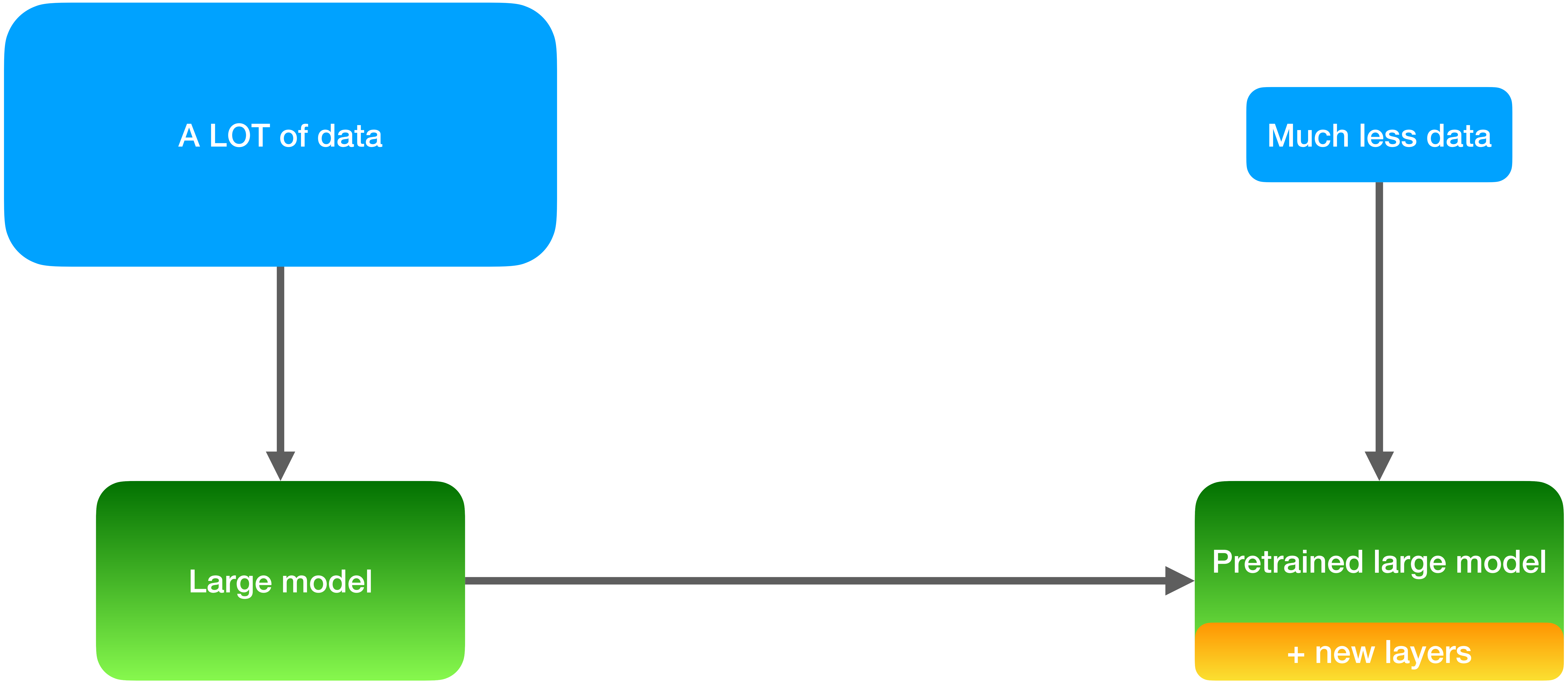
Transfer Learning



Traditional Machine Learning: slow training on a lot of data

Transfer learning:

Transfer Learning

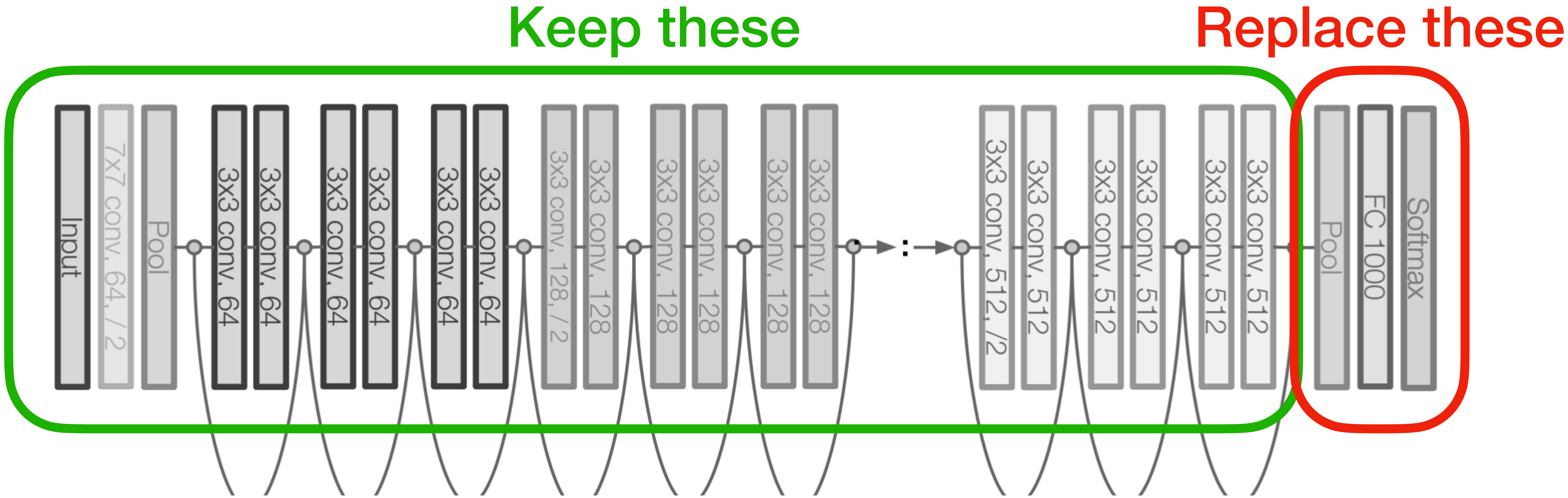


Traditional Machine Learning:
slow training on a lot of data

Transfer learning:
fast training on a little data

Transfer Learning

Imagenet-pretrained model



Transfer Learning

Transfer Learning

```python
class LitModel(pl.LightningModule):
    def __init__(self, input_shape, num_classes, learning_rate=2e-4):
        super().__init__()

        # log hyperparameters
        self.save_hyperparameters()
        self.learning_rate = learning_rate
        self.dim = input_shape
        self.num_classes = num_classes

        # transfer learning if pretrained=True
        self.feature_extractor = models.resnet18(pretrained=True)
        # layers are frozen by using eval()
        self.feature_extractor.eval()

        n_sizes = self._get_conv_output(input_shape)

        self.classifier = nn.Linear(n_sizes, num_classes)

    # returns the size of the output tensor going into the Linear layer from the conv
    def _get_conv_output(self, shape):
        batch_size = 1
        input = torch.autograd.Variable(torch.rand(batch_size, *shape))

        output_feat = self._forward_features(input)
        n_size = output_feat.data.view(batch_size, -1).size(1)
        return n_size
```

```python
    # returns the feature tensor from the conv block
    def _forward_features(self, x):
        x = self.feature_extractor(x)
        return x

    # will be used during inference
    def forward(self, x):
       x = self._forward_features(x)
       x = x.view(x.size(0), -1)
       x = F.log_softmax(self.classifier(x), dim=1)

       return x
```

https://wandb.ai/wandb/wandb-lightning/reports/Transfer-Learning-Using-PyTorch-Lightning--VmlldzoyODk2MjA

Model Zoos

TORCHVISION.MODELS

The models subpackage contains definitions of models for addressing different tasks, including: image classification, pixelwise semantic segmentation, object detection, instance segmentation, person keypoint detection and video classification.

Classification

The models subpackage contains definitions for the following model architectures for image classification:

- AlexNet
- VGG
- ResNet
- SqueezeNet
- DenseNet
- Inception v3
- GoogLeNet
- ShuffleNet v2
- MobileNet v2
- ResNeXt
- Wide ResNet
- MNASNet

https://pytorch.org/docs/stable/torchvision/models.html

TensorFlow Model Garden

Computer Vision

Model	Description	Reference
MNIST	A basic model to classify digits from the MNIST dataset	Link
ResNet	A deep residual network for image recognition	arXiv:1512.03385
RetinaNet	A fast and powerful object detector	arXiv:1708.02002
Mask R-CNN	An object detection and instance segmentation model	arXiv:1703.06870

https://github.com/tensorflow/models/tree/master/official

Questions?

Outline

- Transfer Learning in Computer Vision
- **Embeddings and Language Models**
- "NLP's ImageNet Moment": ELMO/ULMFit
- Transformers
  - Attention in detail
  - BERT, GPT-2, DistillBERT, T5

Converting words to vectors

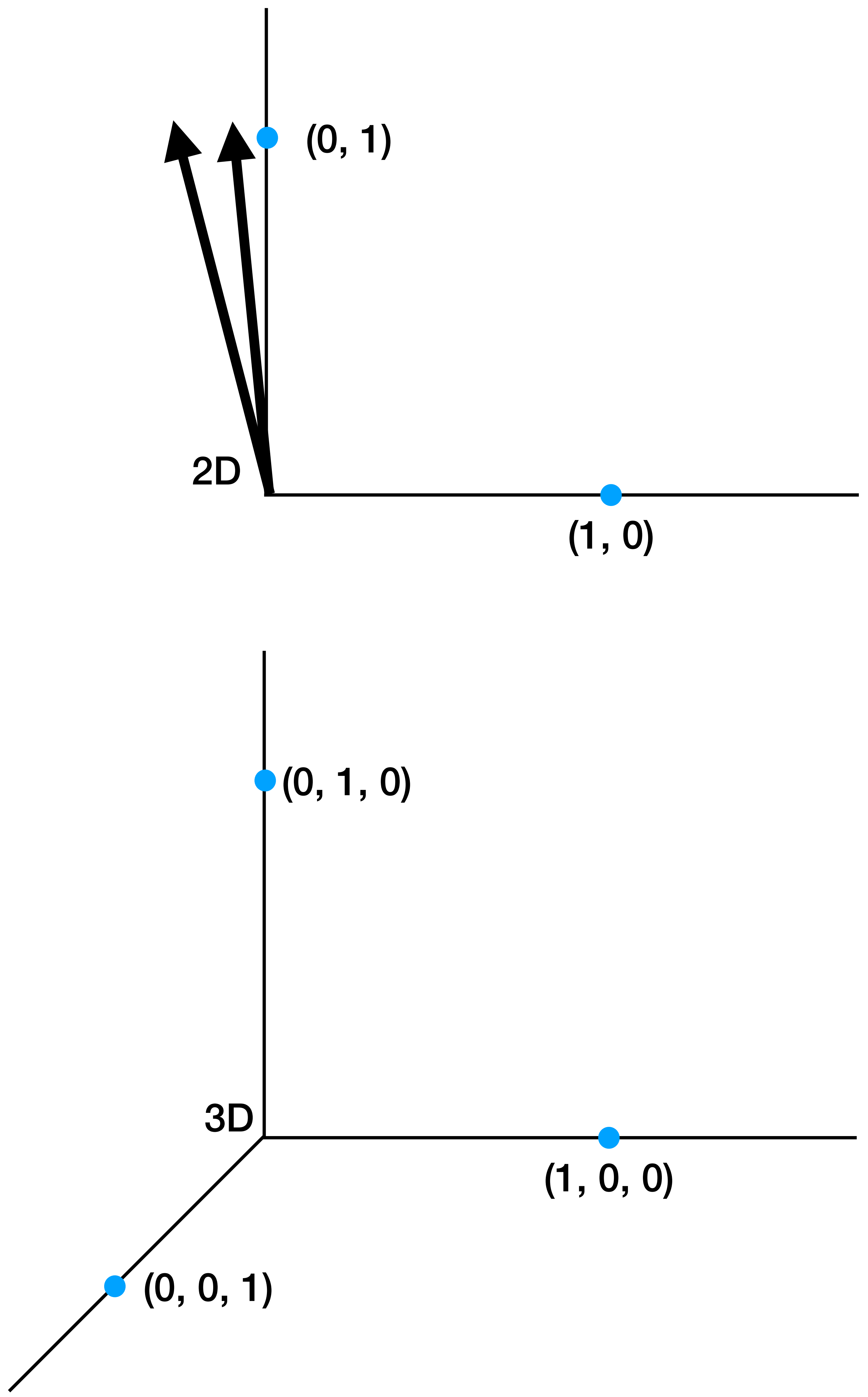
- In NLP, our inputs are sequences of words, but deep learning needs vectors.
- How to convert words to vectors?
- Idea: one-hot encoding

One-hot encoding

https://towardsdatascience.com/why-do-we-use-embeddings-in-nlp-2f20e1b632d2

Problems with one-hot encoding

- Scales poorly with vocabulary size
- Very high-dimensional sparse vectors -> NN operations work poorly
- Violates what we know about word similarity (e.g. "run" is as far away from "running" as from "tertiary," or "poetry")



Map one-hot to dense vectors

sparse one-hot
encoding of words

aardvark	1	0	0	...	0	0	0
black	0	0	...	1	...	0	0
cat	0	0	...	1	...	0	0
duvet	0	0	...	1	...	0	0
zombie	0	0	0	...	0	0	1

VxV matrix

VxE
embedding matrix

	animal	fluffiness	dangerous	spooky
aardvark	0.97	0.03	0.15	0.04
black	0.07	0.01	0.20	0.95
cat	0.98	0.98	0.45	0.35
duvet	0.01	0.84	0.12	0.02
zombie	0.74	0.05	0.98	0.93

VxE matrix

- Problem: how do we find the values of the embedding matrix?

https://towardsdatascience.com/why-do-we-use-embeddings-in-nlp-2f20e1b632d2

Solution 1: Learn as part of the task

EMBEDDING

CLASS `torch.nn.Embedding(num_embeddings: int, embedding_dim: int, padding_idx: Optional[int] = None, max_norm: Optional[float] = None, norm_type: float = 2.0, scale_grad_by_freq: bool = False, sparse: bool = False, _weight: Optional[torch.Tensor] = None)` [SOURCE]

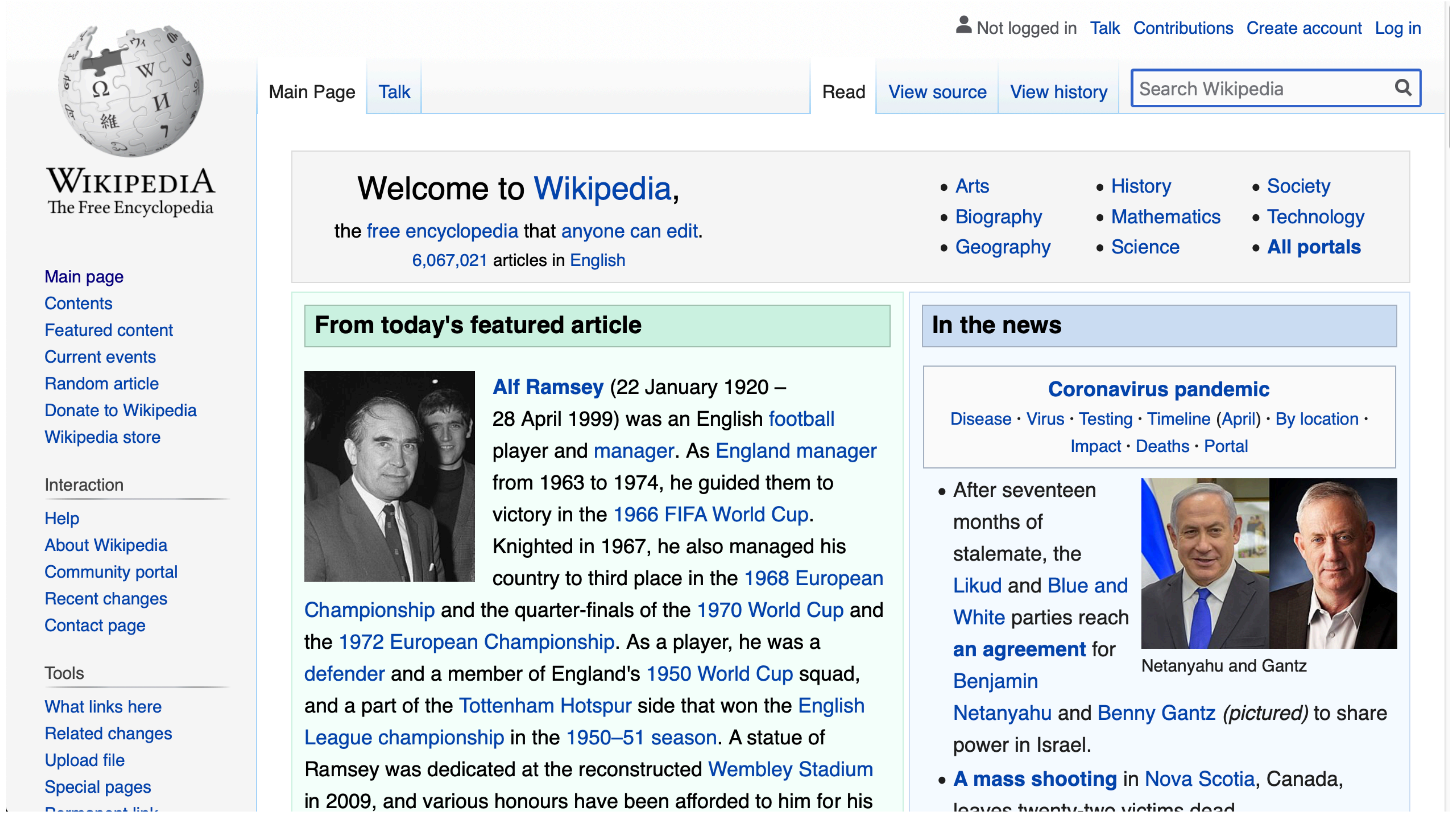
A simple lookup table that stores embeddings of a fixed dictionary and size.

This module is often used to store word embeddings and retrieve them using indices. The input to the module is a list of indices, and the output is the corresponding word embeddings.

Solution 2: Learn a Language Model

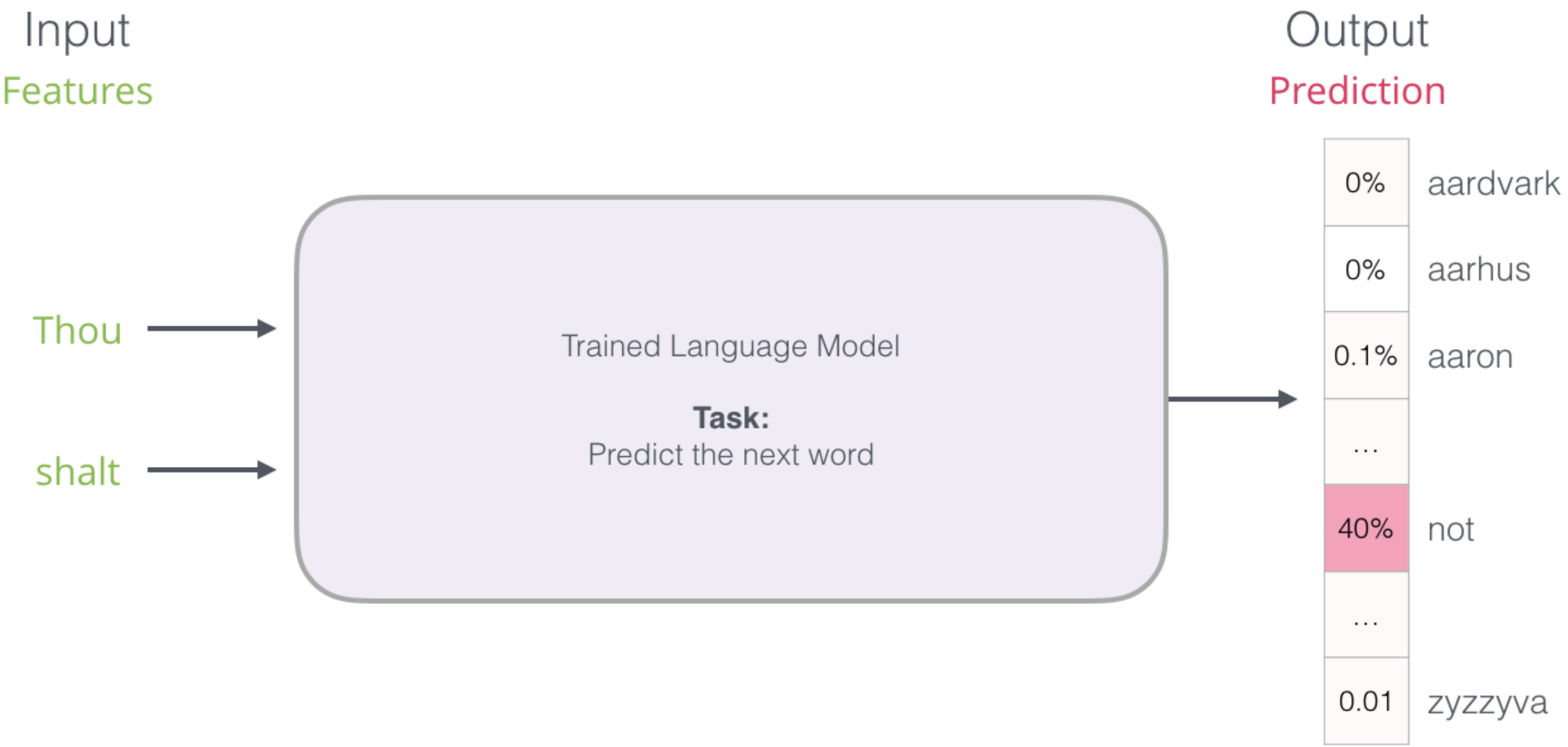
- "Pre-train" for your NLP task by learning a really good word embedding!

- How to learn a really good embedding? Train for a very general task on a large corpus of text.



+ Jay was hit by a ___?

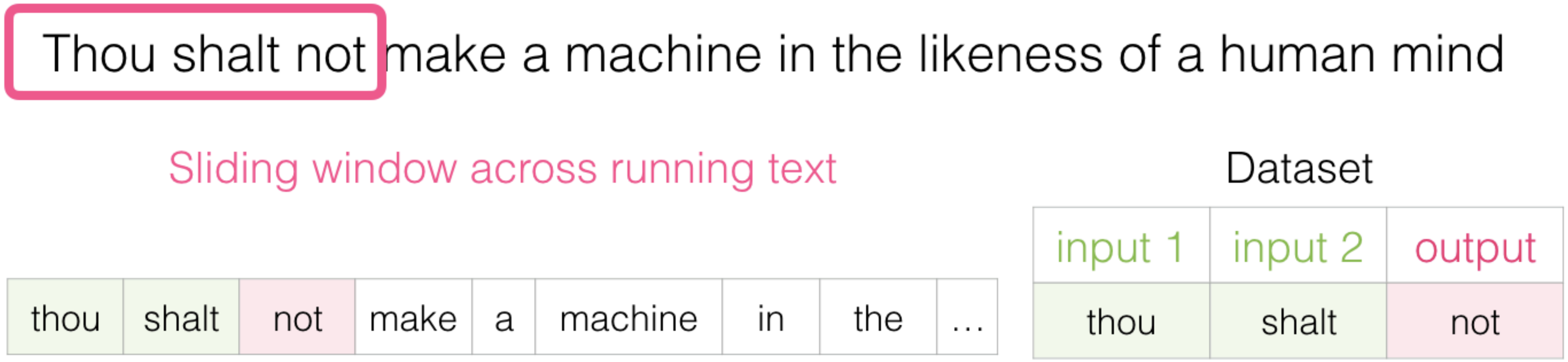
Solution 2: Learn a Language Model



http://jalammar.github.io/illustrated-word2vec/

N-Grams

- Slide an N-sized window through the text, forming a dataset of predicting the last word.



http://jalammar.github.io/illustrated-word2vec/

N-Grams

- Slide an N-sized window through the text, forming a dataset of predicting the last word.

Thou $\boxed{\text{shalt not make}}$ a machine in the likeness of a human mind

Sliding window across running text

thou	shalt	not	make	a	machine	in	the	…
thou	shalt	not	make	a	machine	in	the	

Dataset

input 1	input 2	output
thou	shalt	not
shalt	not	make

http://jalammar.github.io/illustrated-word2vec/

N-Grams

- Slide an N-sized window through the text, forming a dataset of predicting the last word.

Thou shalt not make a machine in the likeness of a human mind

Sliding window across running text

thou	shalt	not	make	a	machine	in	the	…
thou	shalt	not	make	a	machine	in	the	
thou	shalt	not	make	a	machine	in	the	
thou	shalt	not	make	a	machine	in	the	
thou	shalt	not	make	a	machine	in	the	

Dataset

input 1	input 2	output
thou	shalt	not
shalt	not	make
not	make	a
make	a	machine
a	machine	in

http://jalammar.github.io/illustrated-word2vec/

Skip-grams

- Look on both sides of the target word, and form multiple samples from each N-gram

Thou shalt not make a machine in the likeness of a human mind

thou	shalt	not	make	a	machine	in	the	…

input word	target word
not	thou
not	shalt
not	make
not	a

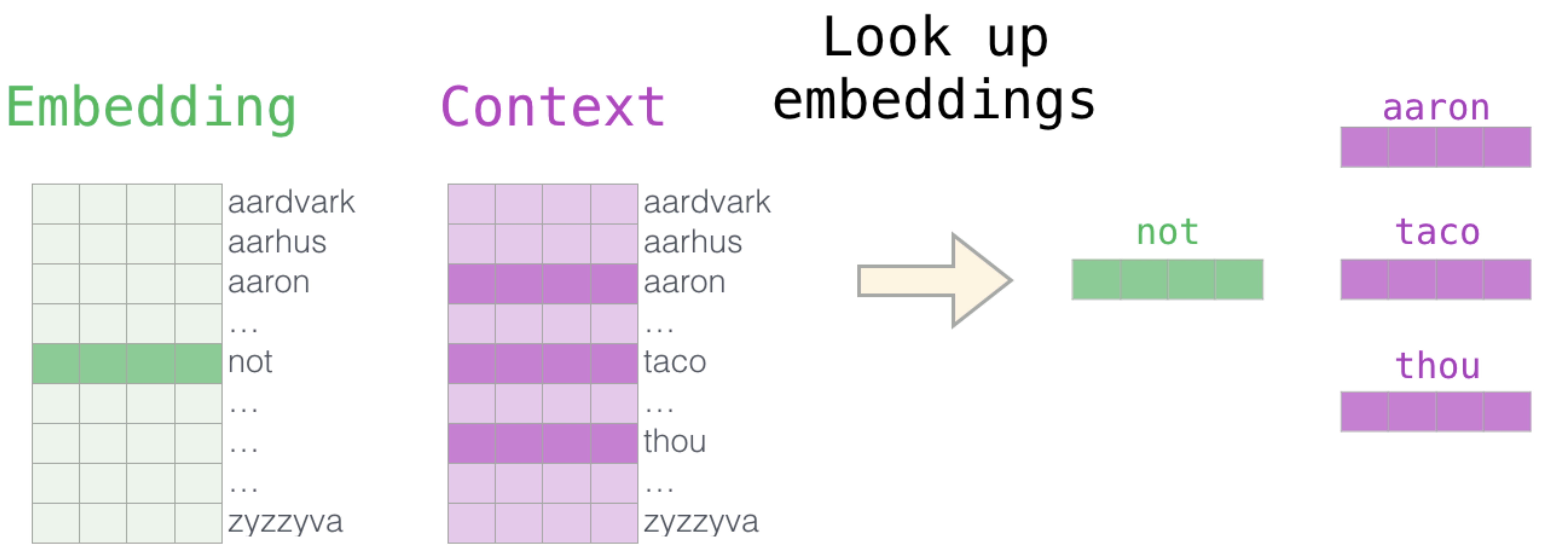
http://jalammar.github.io/illustrated-word2vec/

Speed up training

- Binary instead of multi-class (faster training)

http://jalammar.github.io/illustrated-word2vec/

Training

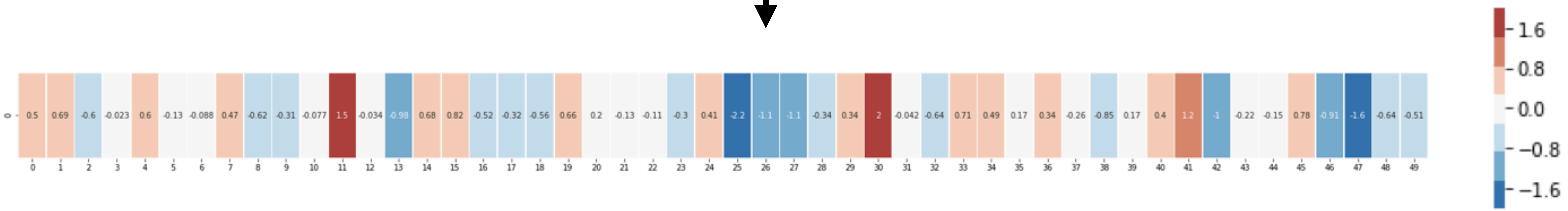


input word	output word	target	input • output	sigmoid()
not	thou	1	0.2	0.55
not	aaron	0	-1.11	0.25
not	taco	0	0.74	0.68

http://jalammar.github.io/illustrated-word2vec/

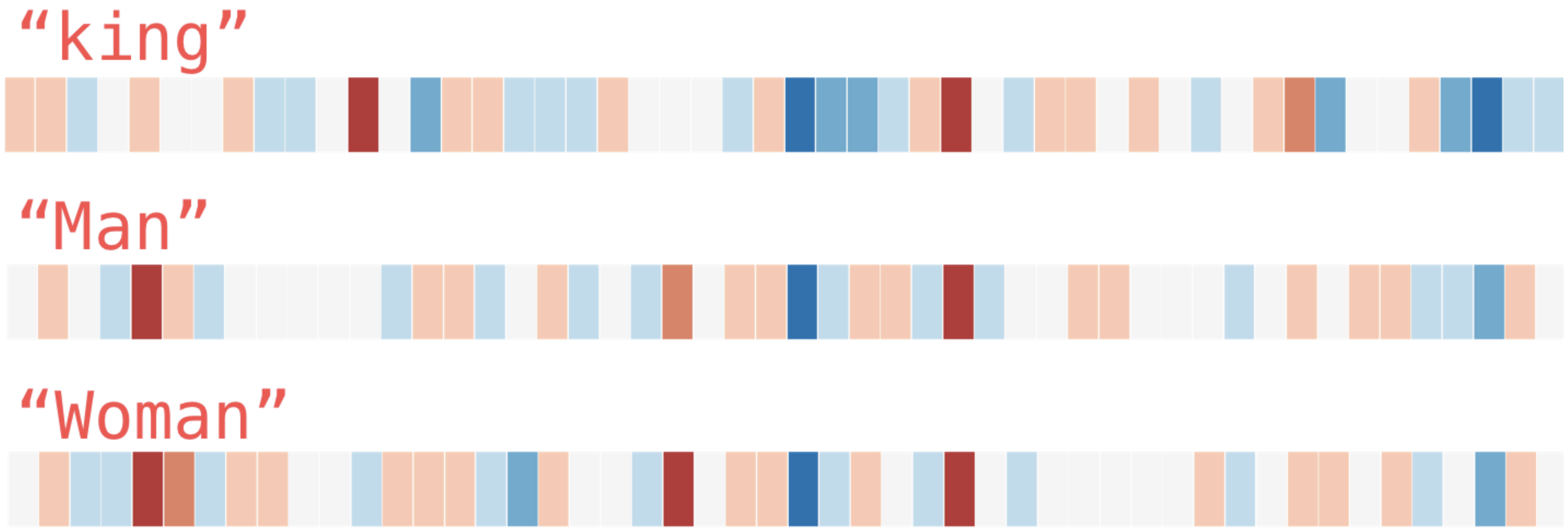
Word2Vec (2013)

"king"



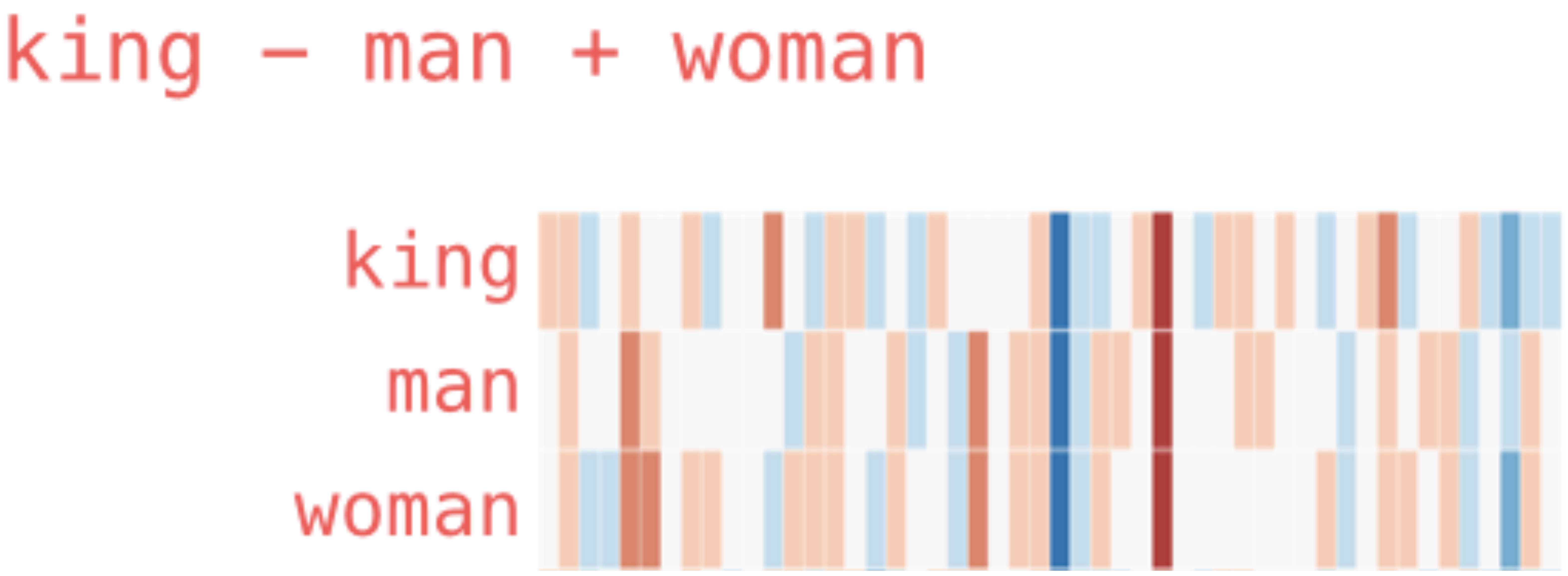
http://jalammar.github.io/illustrated-word2vec/

Word2Vec (2013)



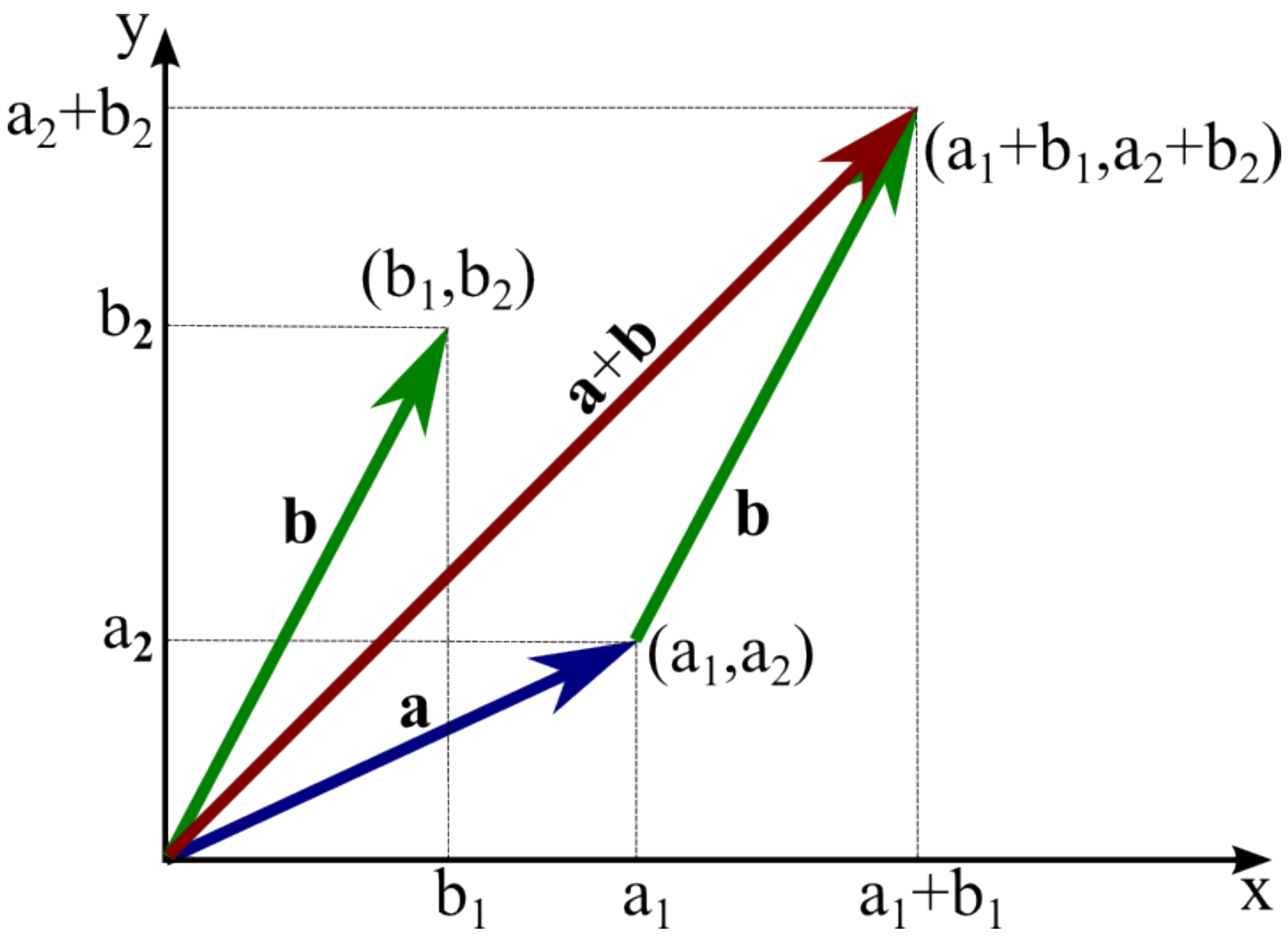
http://jalammar.github.io/illustrated-word2vec/

Word2Vec (2013)



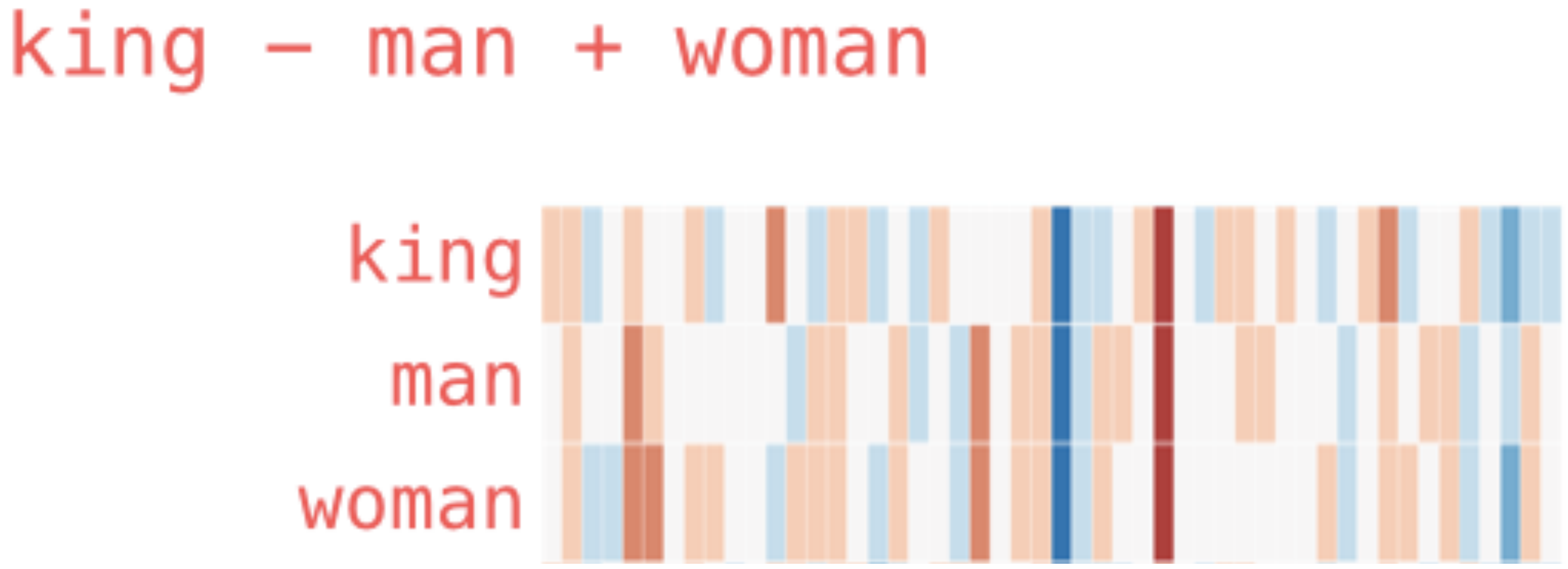
http://jalammar.github.io/illustrated-word2vec/

Vector math...



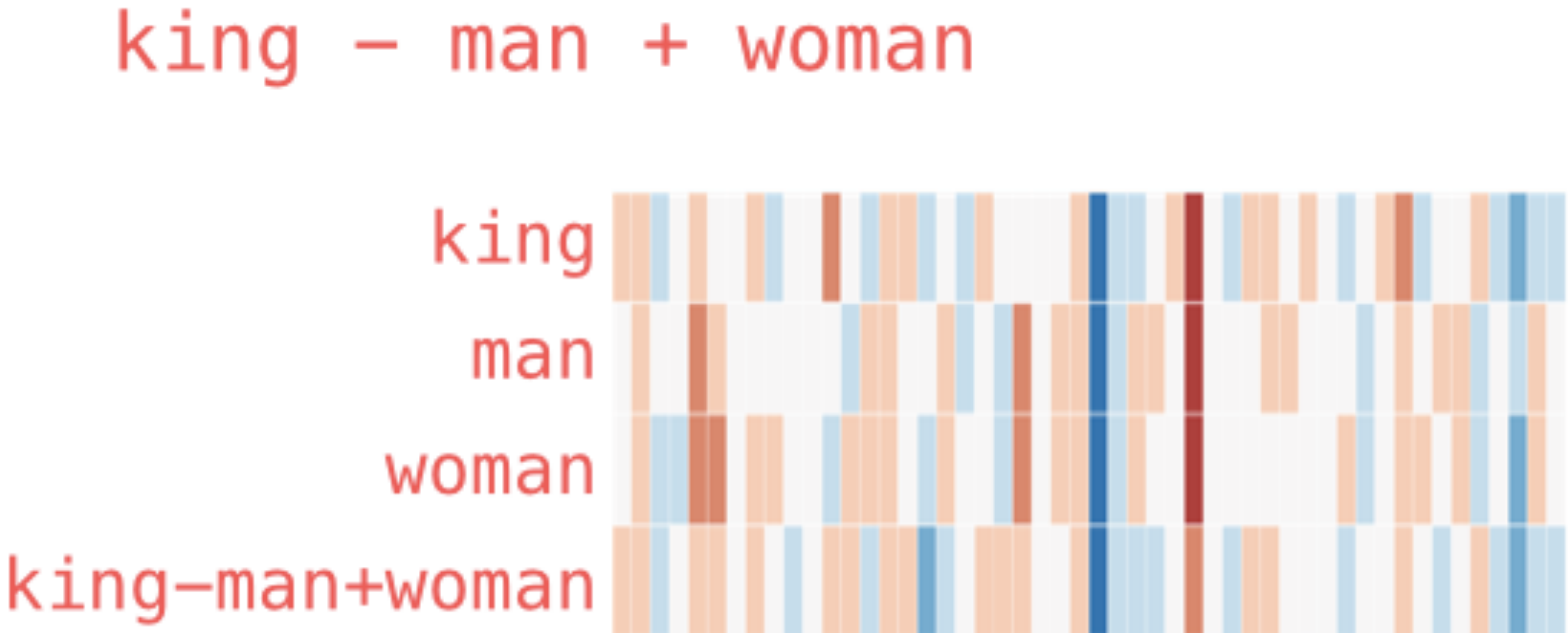
https://mathinsight.org/vectors_cartesian_coordinates_2d_3d

Word2Vec (2013)



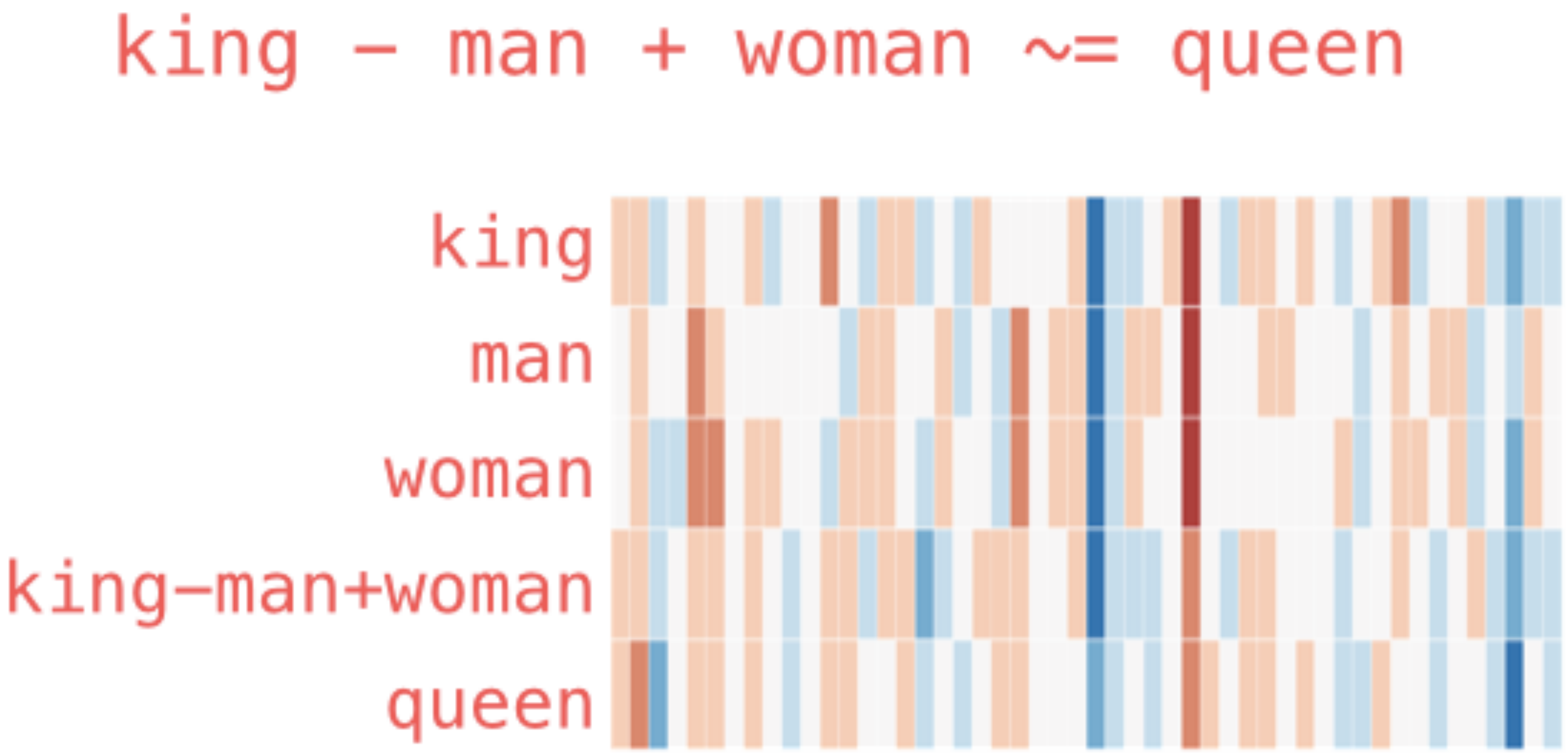
http://jalammar.github.io/illustrated-word2vec/

Word2Vec (2013)



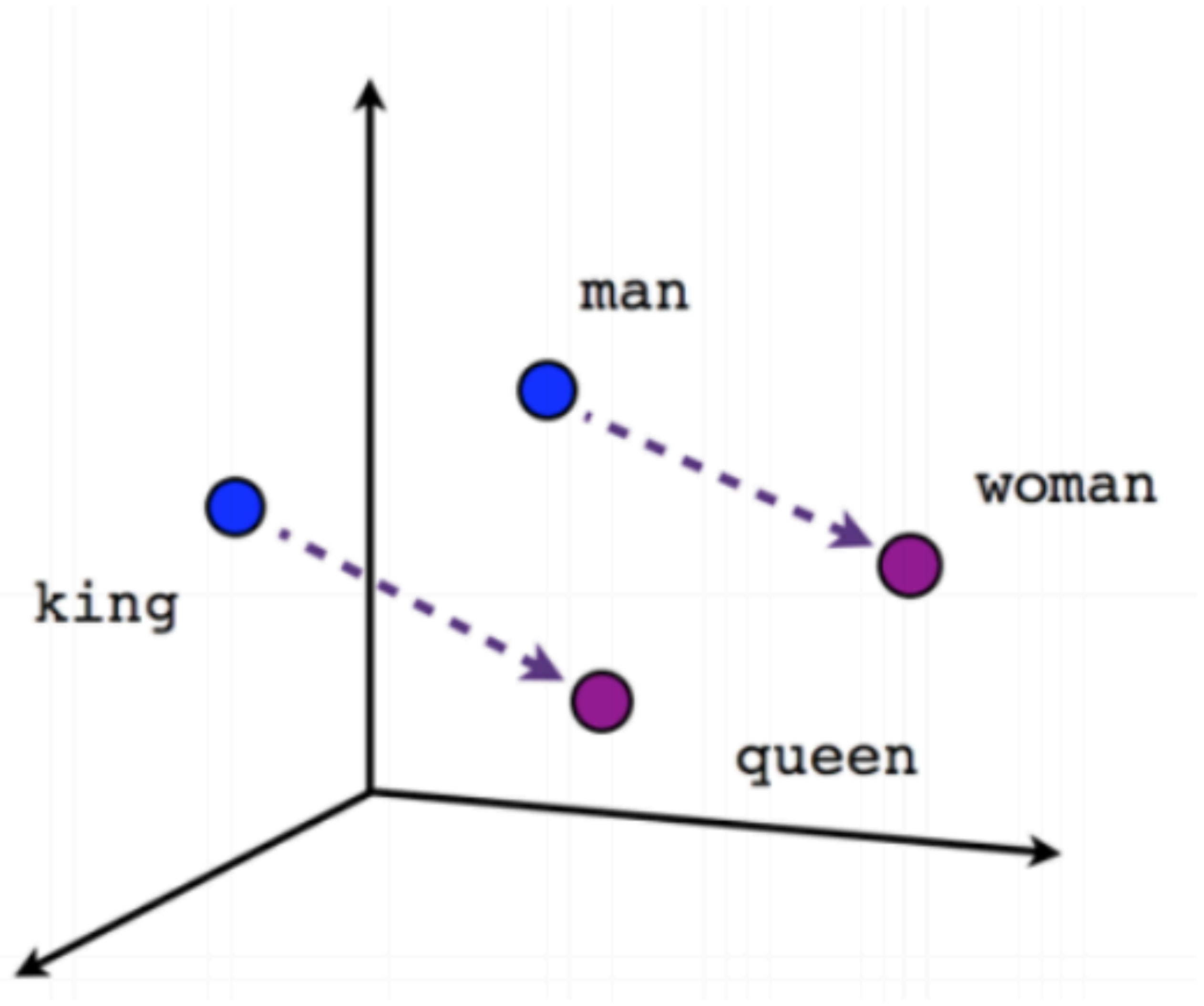
http://jalammar.github.io/illustrated-word2vec/

Word2Vec (2013)

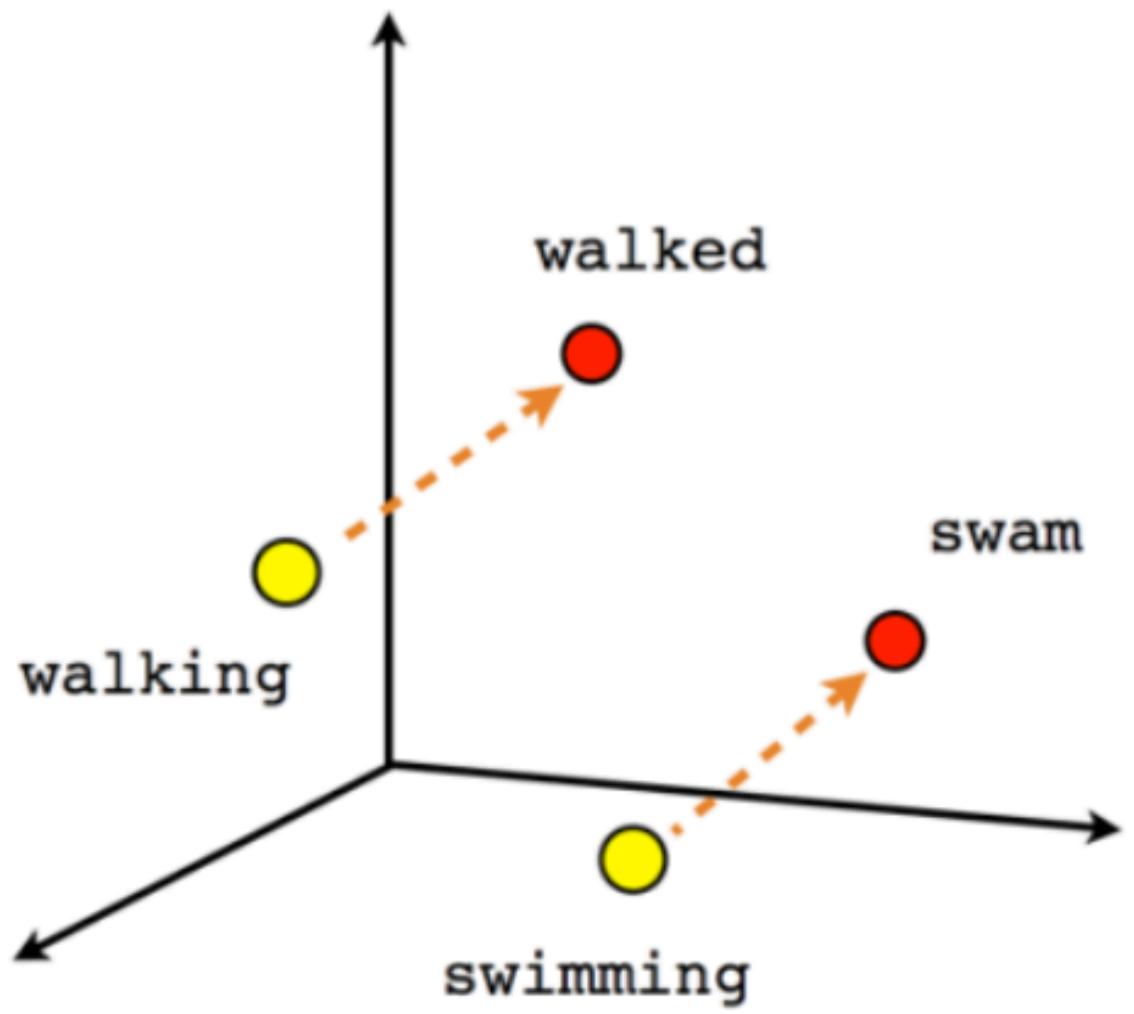


http://jalammar.github.io/illustrated-word2vec/

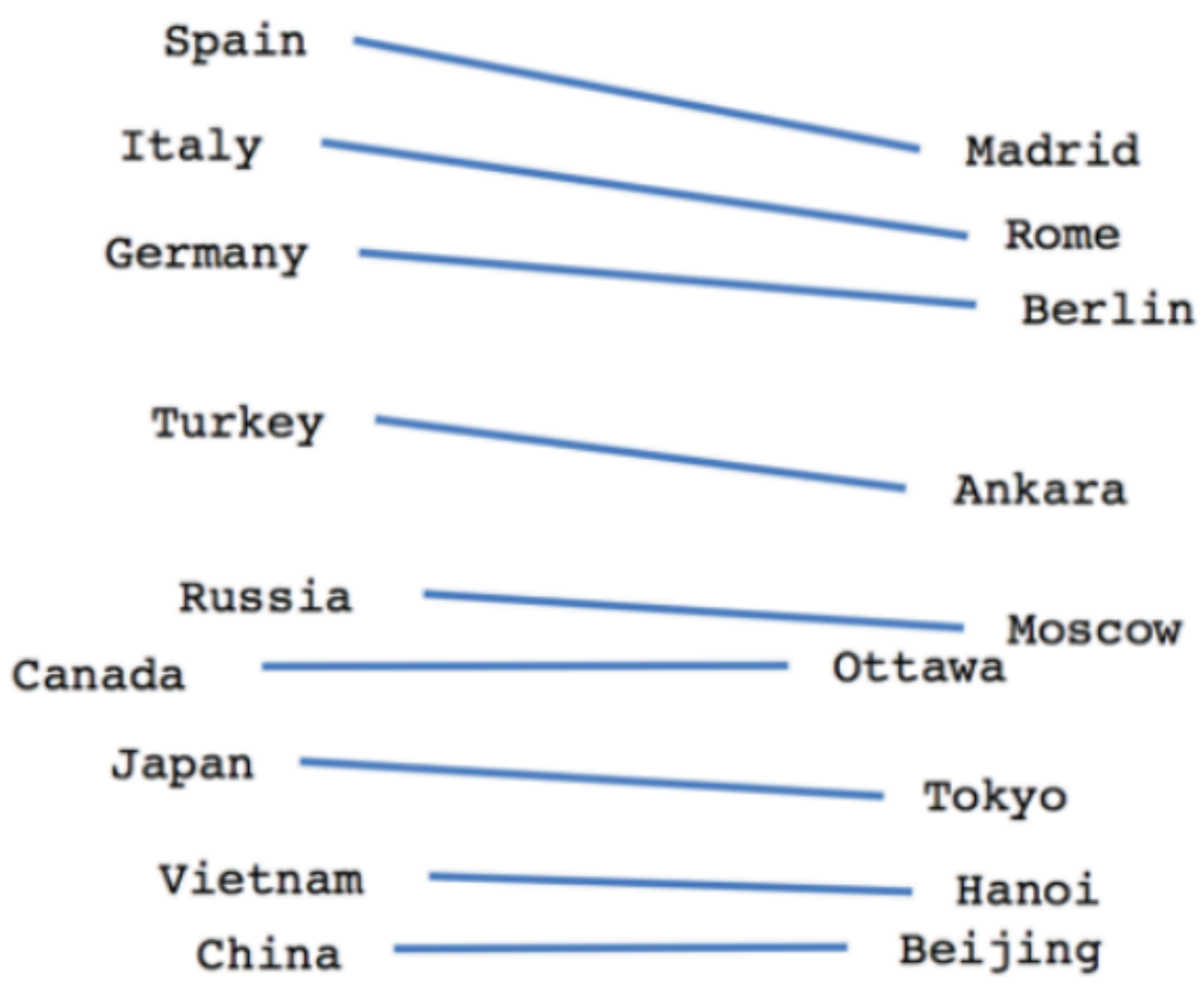
Word2Vec (2013)



Male-Female



Verb tense



Country-Capital

https://ruder.io/nlp-imagenet/

Questions?

Outline

- Transfer Learning in Computer Vision
- Embeddings and Language Models
- **"NLP's ImageNet Moment": ELMO/ULMFit**
- Transformers
  - Attention in detail
  - BERT, GPT-2, DistillBERT, T5

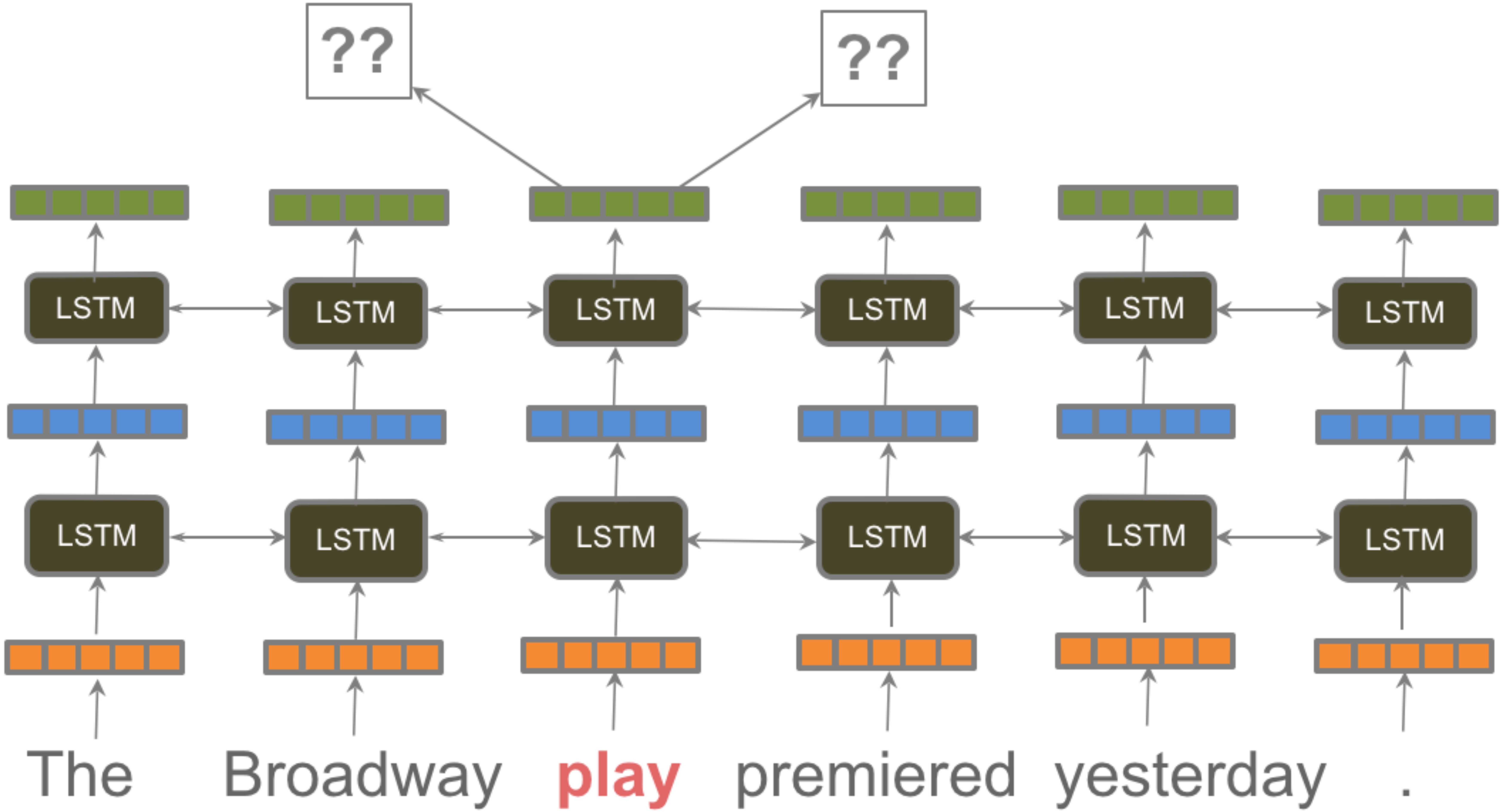
Beyond embeddings

- Word2Vec and GloVe embeddings became popular in ~2013-14
- Boosted accuracy on many tasks by low single-digit %
- But these representations are shallow:
    - only first layer would have benefit of seeing all of Wikipedia
    - rest of the model -- LSTMs, etc -- would be trained only on the task dataset (much smaller)
    - Why not pre-train more layers, and thus disambiguate words (e.g. *rule* in "to rule" vs "a rule"), learn grammar, etc?

Elmo (2018)

Allen Institute for AI

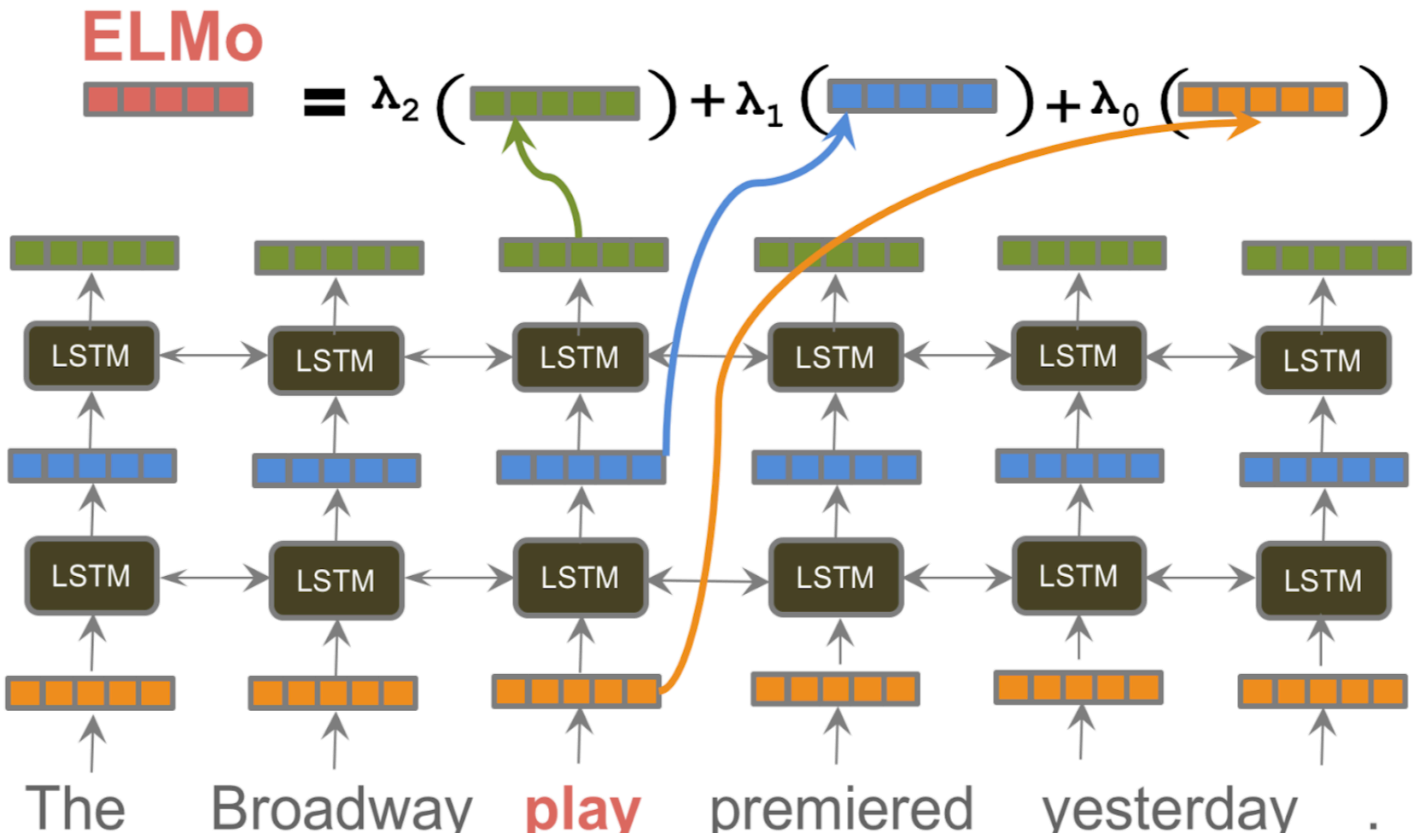
- Bidirectional stacked LSTM!



https://ruder.io/nlp-imagenet/

Elmo (2018)

Allen Institute for AI



Elmo (2018)

Allen Institute for AI

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

https://www.slideshare.net/shuntaroy/a-review-of-deep-contextualized-word-representations-peters-2018

Q&A: SQuAD

- 100K question-answer pairs
- Answers are always spans in the question

https://arxiv.org/abs/1606.05250

In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under **gravity**. The main forms of precipitation include drizzle, rain, sleet, snow, **graupel** and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals **within a cloud**. Short, intense periods of rain in scattered locations are called "showers".

What causes precipitation to fall?
gravity

What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?
graupel

Where do water droplets collide with ice crystals to form precipitation?
within a cloud

Natural Language Inference: SNLI

- What relation exists between piece of text and hypothesis?
- 570K pairs

A man inspects the uniform of a figure in some East Asian country.	**contradiction** C C C C C	The man is sleeping
An older and younger man smiling.	**neutral** N N E N N	Two men are smiling and laughing at the cats playing on the floor.
A black race car starts up in front of a crowd of people.	**contradiction** C C C C C	A man is driving down a lonely road.
A soccer game with multiple males playing.	**entailment** E E E E E	Some men are playing a sport.
A smiling costumed woman is holding an umbrella.	**neutral** N N E C N	A happy woman in a fairy costume holds an umbrella.

https://nlp.stanford.edu/pubs/snli_paper.pdf

Elmo (2018)

Allen Institute for AI

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

https://www.slideshare.net/shuntaroy/a-review-of-deep-contextualized-word-representations-peters-2018

GLUE

Description	Data example
Is the sentence grammatical or ungrammatical?	"This building is than that one." = **Ungrammatical**
Is the movie review positive, negative, or neutral?	"The movie is funny , smart , visually inventive , and most of all , alive ." = **.93056 (Very Positive)**
Is the sentence B a paraphrase of sentence A?	A) "Yesterday , Taiwan reported 35 new infections , bringing the total number of cases to 418 ." B) "The island reported another 35 probable cases yesterday , taking its total to 418 ." = **A Paraphrase**
How similar are sentences A and B?	A) "Elephants are walking down a trail." B) "A herd of elephants are walking along a trail." = **4.6 (Very Similar)**
Are the two questions similar?	A) "How can I increase the speed of my internet connection while using a VPN?" B) "How can Internet speed be increased by hacking through DNS?" = **Not Similar**
Does sentence A entail or contradict sentence B?	A) "Tourist Information offices can be very helpful." B) "Tourist Information offices are never of any help." = **Contradiction**

https://mccormickml.com/2019/11/05/GLUE/

GLUE

Does sentence B contain the answer to the question in sentence A?	A) "What is essential for the mating of the elements that create radio waves?" B) "Antennas are required by any radio receiver or transmitter to couple its electrical connection to the electromagnetic field." = **Answerable**
Does sentence A entail sentence B?	A) "In 2003, Yunus brought the microcredit revolution to the streets of Bangladesh to support more than 50,000 beggars, whom the Grameen Bank respectfully calls Struggling Members." B) "Yunus supported more than 50,000 Struggling Members." = **Entailed**
Sentence B replaces sentence A's ambiguous pronoun with one of the nouns - is this the correct noun?	A) "Lily spoke to Donna, breaking her concentration." B) "Lily spoke to Donna, breaking Lily's concentration." = **Incorrect Referent**

- 9 tasks, model score is averaged across them

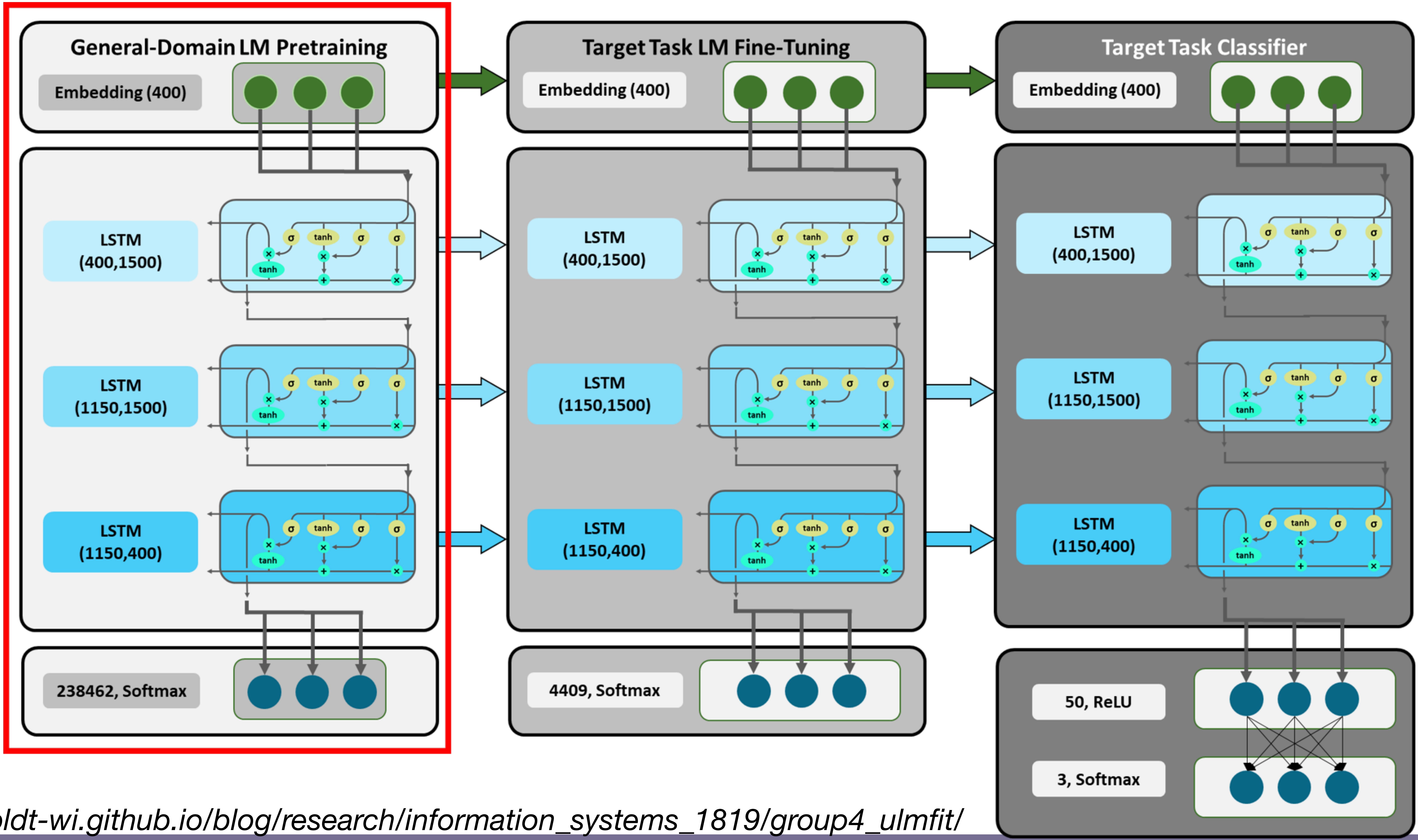
https://mccormickml.com/2019/11/05/GLUE/

ULMFit (2018)

fast.ai Making neural nets uncool again

https://humboldt-wi.github.io/blog/research/information_systems_1819/group4_ulmfit/

ULMFit (2018)



https://humboldt-wi.github.io/blog/research/information_systems_1819/group4_ulmfit/

Pre-training is here!

NATURAL LANGUAGE PROCESSING

NLP's ImageNet moment has arrived

Big changes are underway in the world of NLP. The long reign of word vectors as NLP's core representation technique has seen an exciting new line of challengers emerge. These approaches demonstrated that pretrained language models can achieve state-of-the-art results and herald a watershed moment.

SEBASTIAN RUDER
12 JUL 2018 • 15 MIN READ

Pre-training is here!

TensorFlow Model Garden

Natural Language Processing

Model	Description	Reference
ALBERT	A Lite BERT for Self-supervised Learning of Language Representations	arXiv:1909.11942
BERT	A powerful pre-trained language representation model: BERT (Bidirectional Encoder Representations from Transformers)	arXiv:1810.04805
NHNet	A transformer-based multi-sequence to sequence model: Generating Representative Headlines for News Stories	arXiv:2001.09386
Transformer	A transformer model to translate the WMT English to German dataset	arXiv:1706.03762
XLNet	XLNet: Generalized Autoregressive Pretraining for Language Understanding	arXiv:1906.08237

Questions?

Outline

- Transfer Learning in Computer Vision
- Embeddings and Language Models
- "NLP's ImageNet Moment": ELMO/ULMFit
- **Transformers**
  - Attention in detail
  - BERT, GPT-2, DistillBERT, T5

Rise of Transformers

TensorFlow Model Garden

Natural Language Processing

Model	Description	Reference
ALBERT	A Lite BERT for Self-supervised Learning of Language Representations	arXiv:1909.11942
BERT	A powerful pre-trained language representation model: BERT (Bidirectional Encoder Representations from Transformers)	arXiv:1810.04805
NHNet	A transformer-based multi-sequence to sequence model: Generating Representative Headlines for News Stories	arXiv:2001.09386
Transformer	A transformer model to translate the WMT English to German dataset	arXiv:1706.03762
XLNet	XLNet: Generalized Autoregressive Pretraining for Language Understanding	arXiv:1906.08237

Rise of Transformers

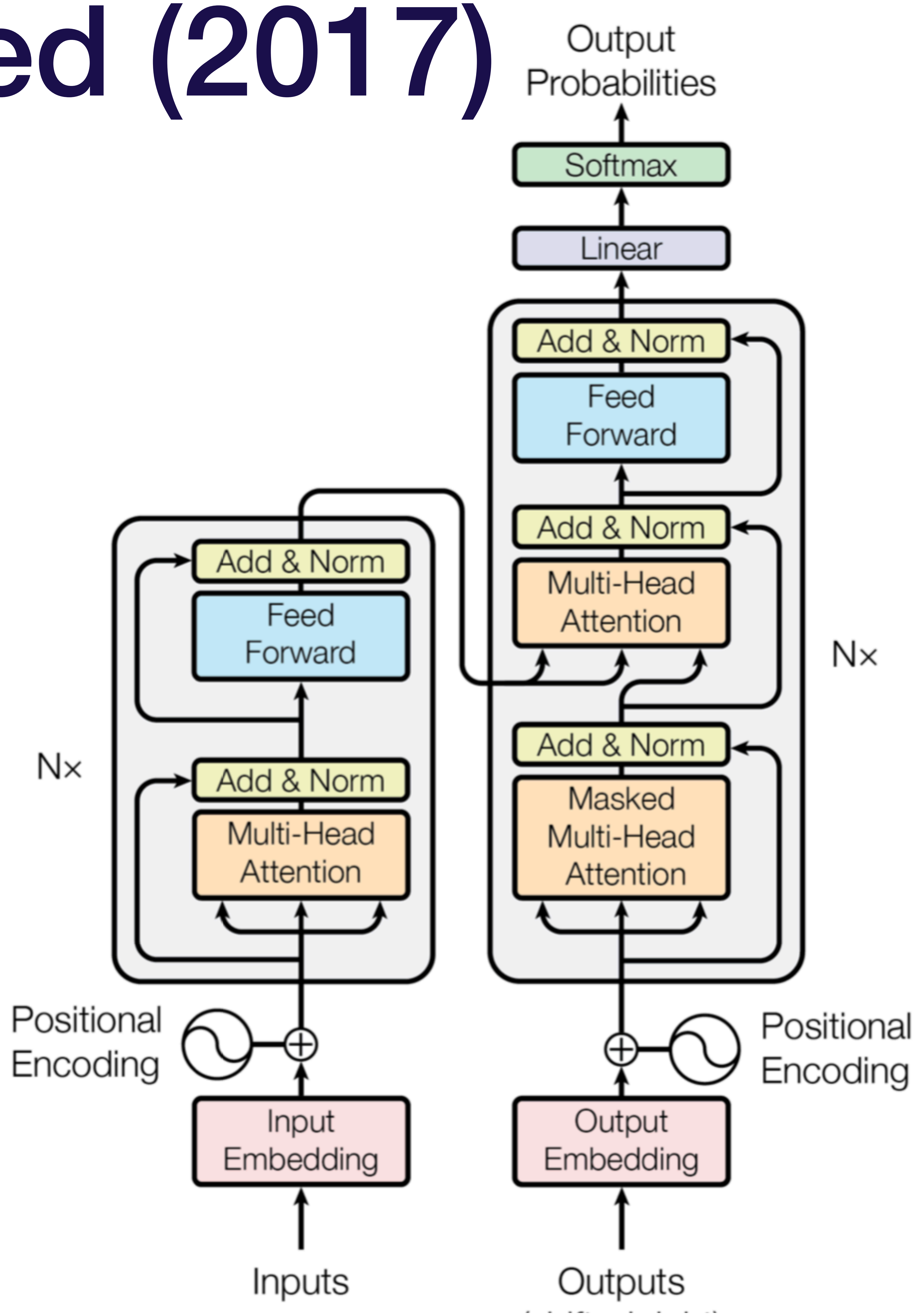
T rden

Natura

Mod	nce
ALBER	11942
BERT	04805
NHNet	09386
Transf	03762
XLNet	08237

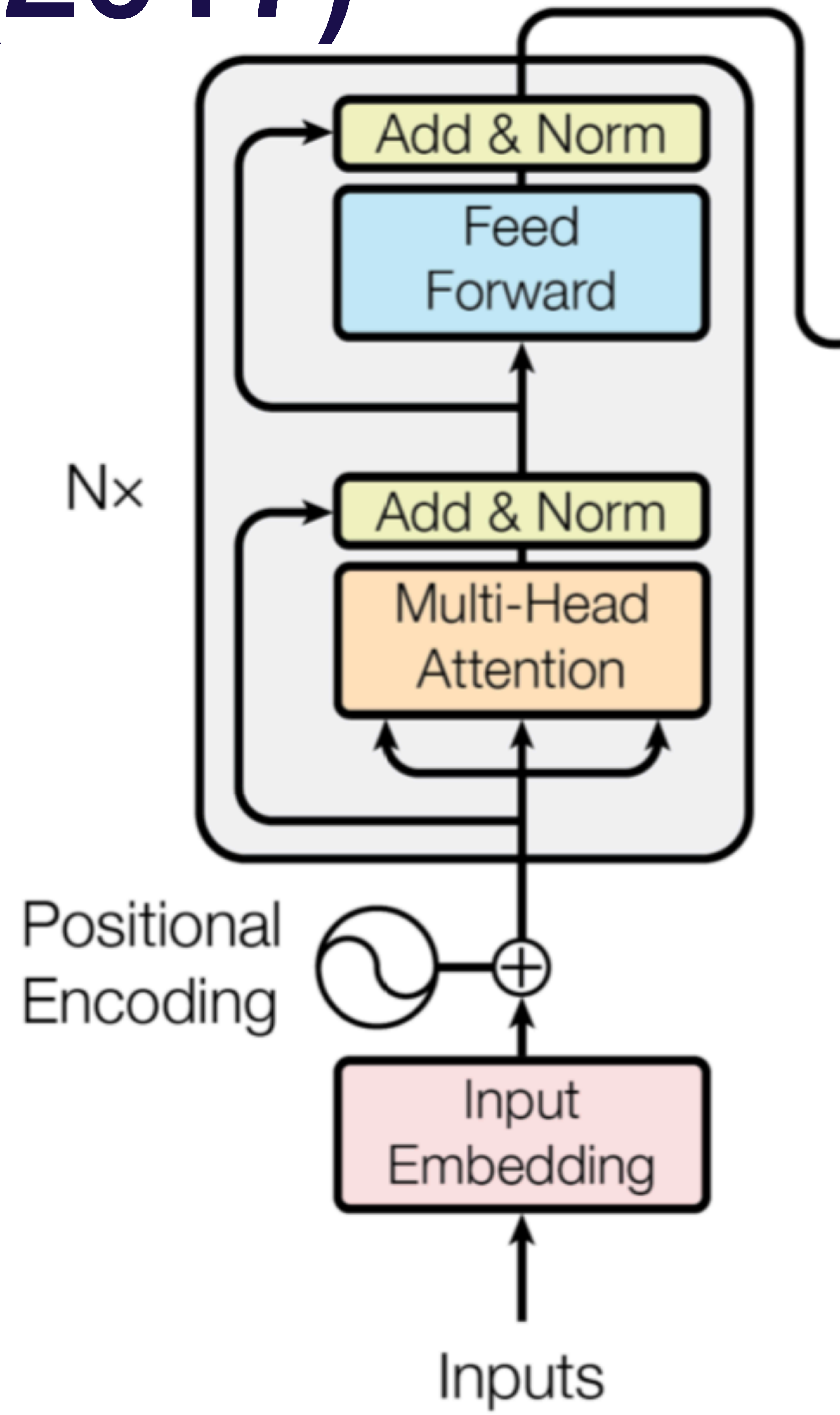
Attention is all you need (2017)

- Encoder-decoder with only attention and fully-connected layers (no recurrence or convolutions)
- Set new SOTA on translation datasets.



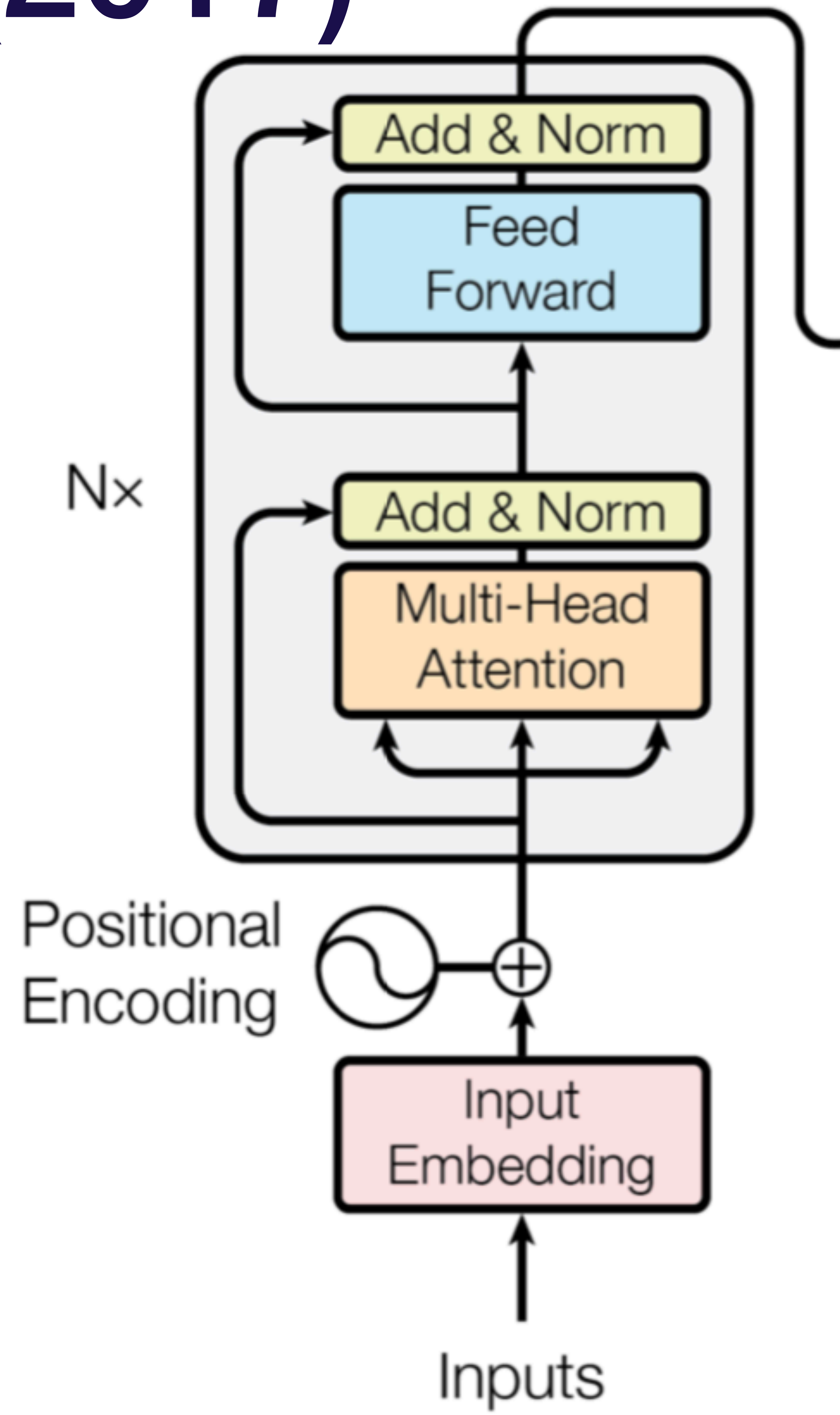
Attention is all you need (2017)

- For simplicity, can focus just on the encoder
    - (e.g BERT is just the encoder)



Attention is all you need (2017)

- The components:
    - (Masked) Self-attention
    - Positional encoding
    - Layer normalization



Outline

- Transfer Learning in Computer Vision
- Embeddings and Language Models
- "NLP's ImageNet Moment": ELMO/ULMFit
- Transformers
  - **Attention in detail**
  - BERT, GPT-2, DistillBERT, T5

Basic self-attention

- Input: sequence of tensors

$$\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_t$$

http://www.peterbloem.nl/blog/transformers

Basic self-attention

- Input: sequence of tensors

$$\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_t$$

- Output: sequence of tensors, each one a weighted sum of the input sequence

$$\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_t$$

$$\mathbf{y}_i = \sum_j w_{ij} \mathbf{x}_j$$

http://www.peterbloem.nl/blog/transformers

Basic self-attention

- Input: sequence of tensors

  $\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_t$

- Output: sequence of tensors, each one a weighted sum of the input sequence

  $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_t$

  $$\mathbf{y}_i = \sum_j w_{ij} \mathbf{x}_j$$

  - not a learned weight, but a function of x_i and x_j $\longrightarrow$ $w'_{ij} = \mathbf{x}_i{}^{\mathsf{T}} \mathbf{x}_j$

http://www.peterbloem.nl/blog/transformers

Basic self-attention

- Input: sequence of tensors

  $\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_t$

- Output: sequence of tensors, each one a weighted sum of the input sequence

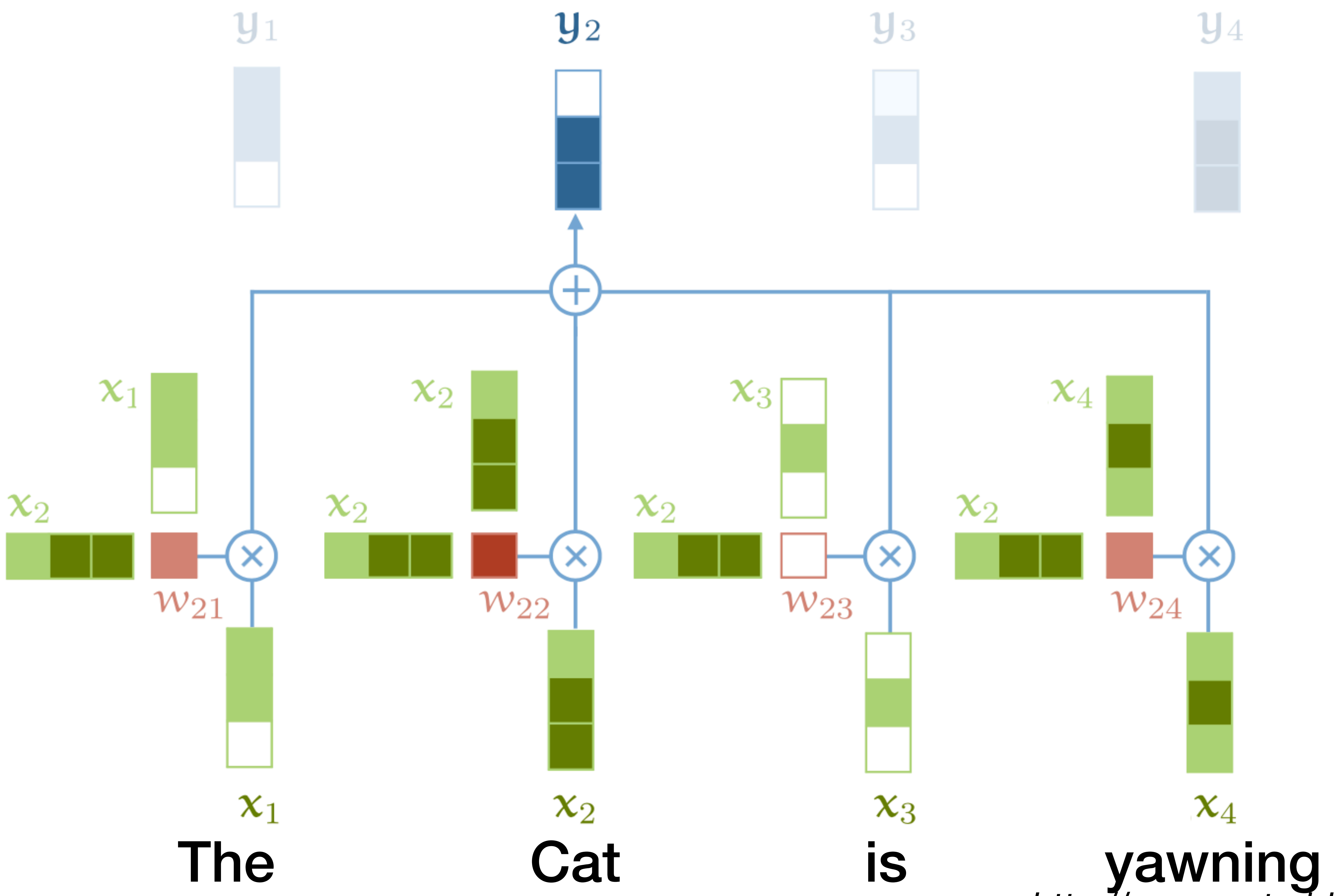
  $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_t$

  $$\mathbf{y}_i = \sum_j w_{ij} \mathbf{x}_j$$

  - not a learned weight, but a function of x_i and x_j $\longrightarrow$ $w'_{ij} = \mathbf{x}_i^T \mathbf{x}_j$
  - must sum to 1 over *j* $\longrightarrow$ $w_{ij} = \dfrac{\exp w'_{ij}}{\sum_j \exp w'_{ij}}$

http://www.peterbloem.nl/blog/transformers

Basic self-attention



http://www.peterbloem.nl/blog/transformers

Basic self-attention

- SO FAR:
  - No learned weights
  - Order of the sequence does not affect result of computations

http://www.peterbloem.nl/blog/transformers

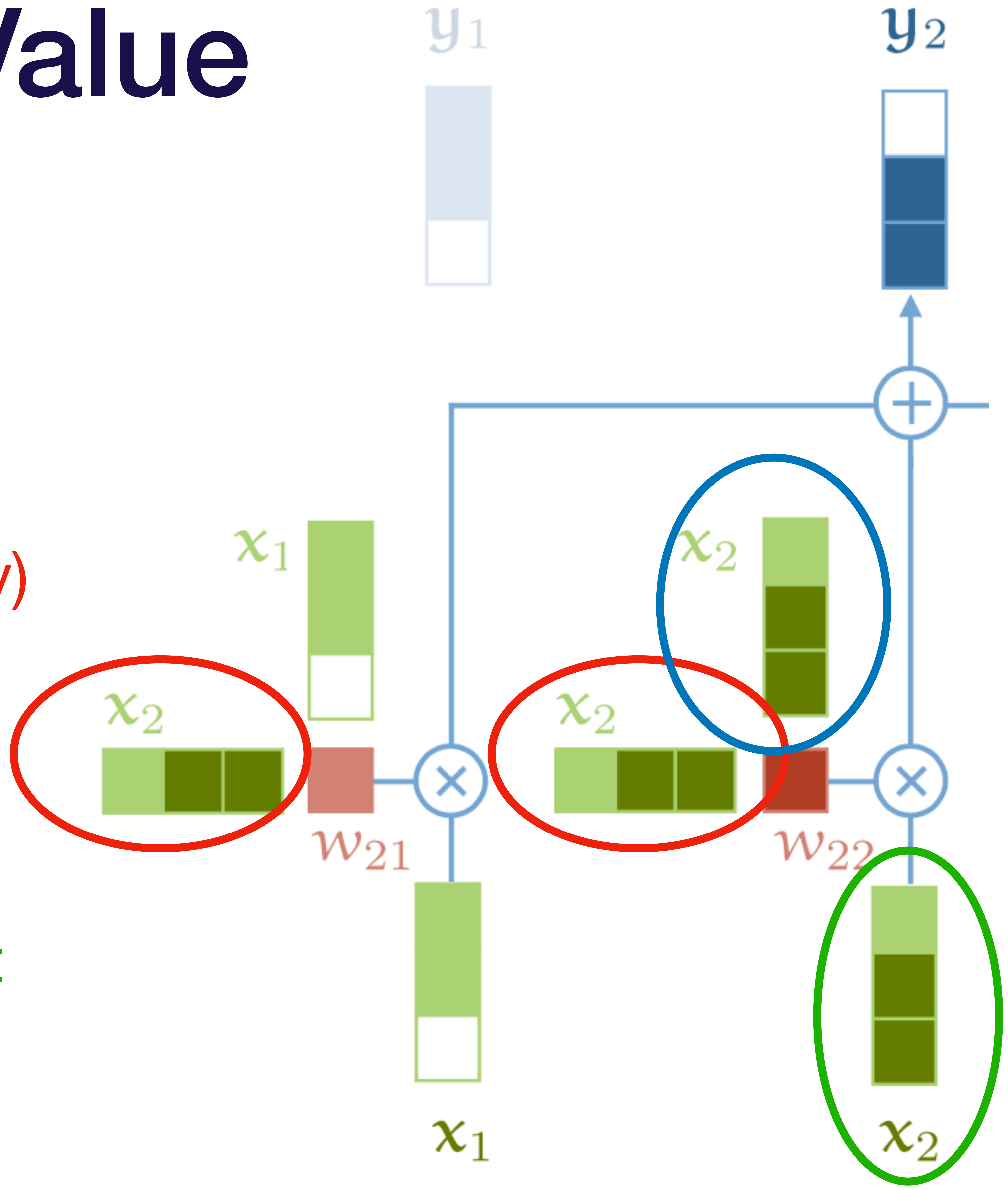
Basic self-attention

- SO FAR:
  - **No learned weights** → **Let's learn some weights!**
  - Order of the sequence does not affect result of computations

http://www.peterbloem.nl/blog/transformers

Query, Key, Value

- Every input vector x_i is used in 3 ways:
  - Compared to every other vector to compute attention weights for its own output y_i (query)
  - Compared to every other vector to compute attention weight w_ij for output y_j (key)
  - Summed with other vectors to form the result of the attention weighted sum (value)



http://www.peterbloem.nl/blog/transformers

Query, Key, Value

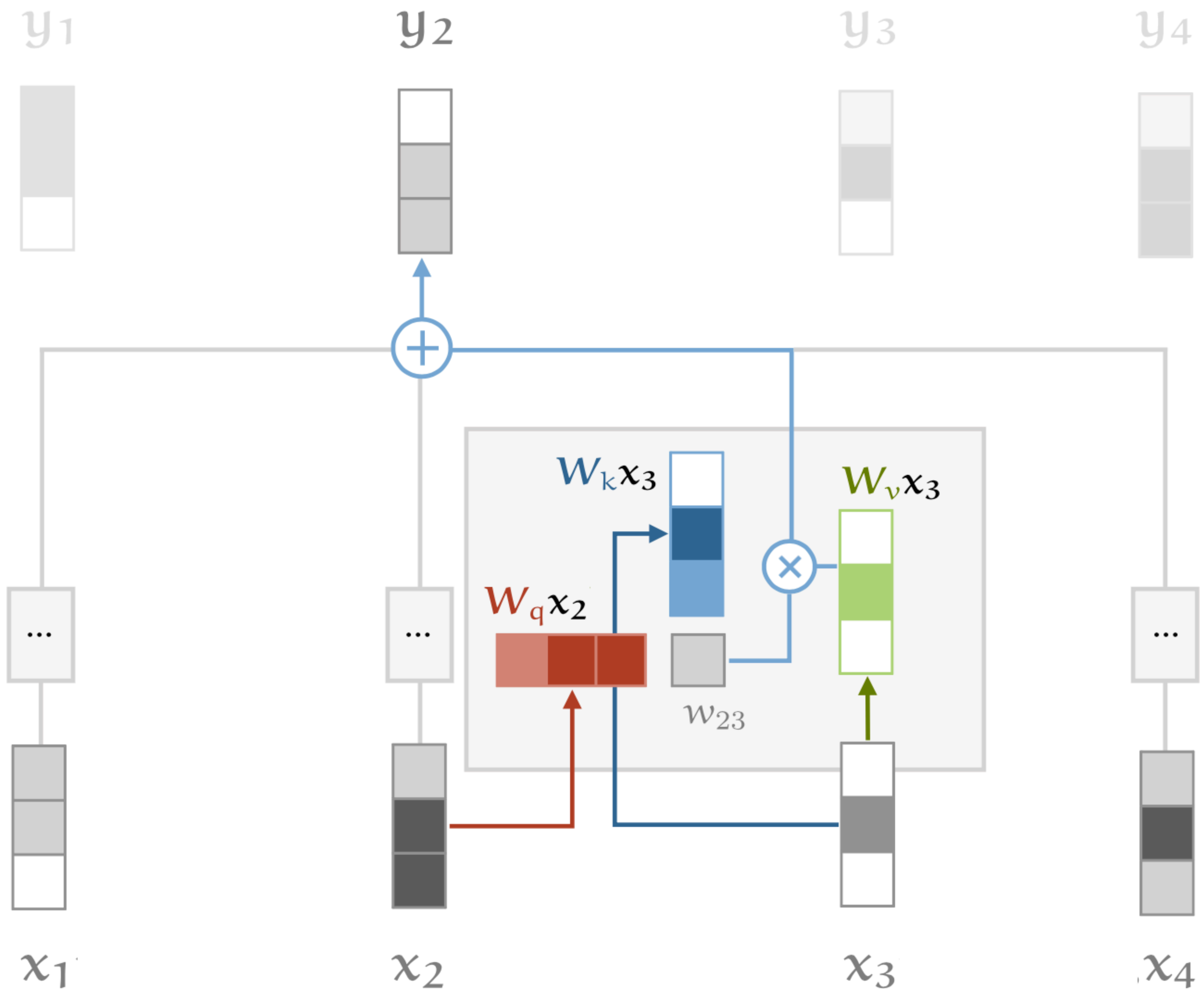
- We can process each input vector to fulfill the three roles with matrix multiplication
- Learning the matrices --> learning attention

$$\mathbf{q}_i = \mathbf{W}_q \mathbf{x}_i \qquad \mathbf{k}_i = \mathbf{W}_k \mathbf{x}_i \qquad \mathbf{v}_i = \mathbf{W}_v \mathbf{x}_i$$

$$w'_{ij} = \mathbf{q}_i{}^\mathsf{T} \mathbf{k}_j$$

$$w_{ij} = \text{softmax}(w'_{ij})$$

$$\mathbf{y}_i = \sum_j w_{ij} \mathbf{v}_j \,.$$

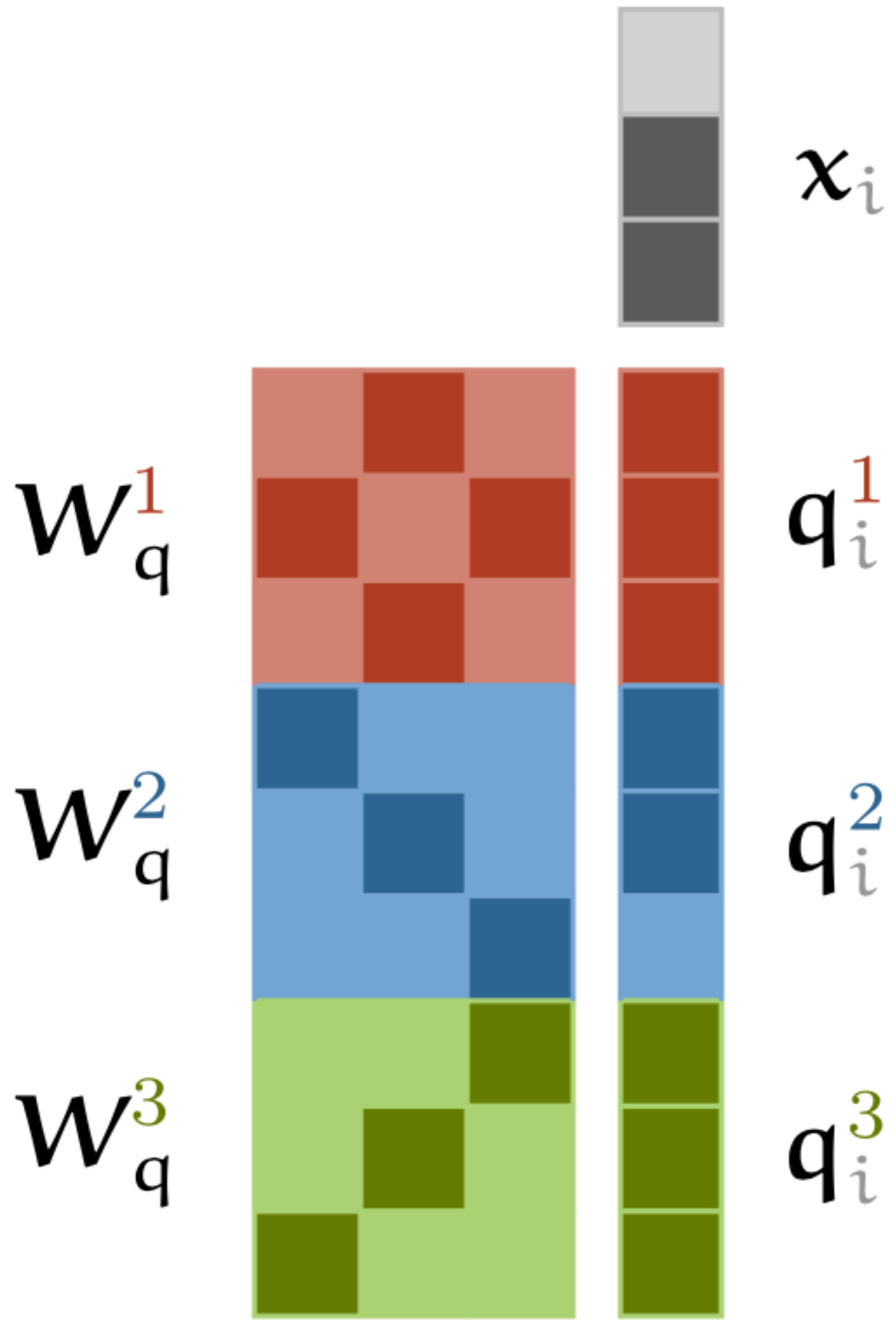


http://www.peterbloem.nl/blog/transformers

Questions?

Multi-head attention

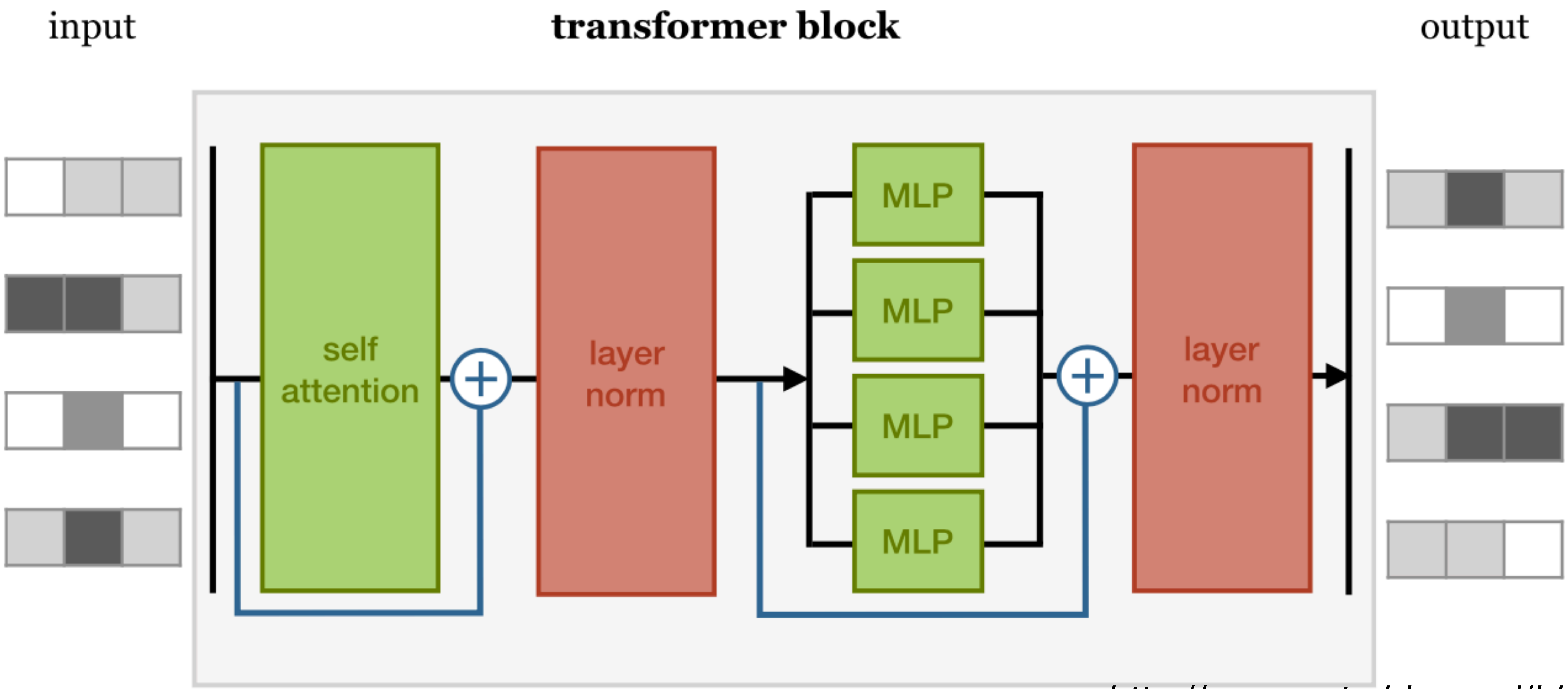
- Multiple "heads" of attention just means learning different sets of W_q, W_k, and W_v matrices simultaneously.
- Implemented as just a single matrix anyway...



http://www.peterbloem.nl/blog/transformers

Transformer

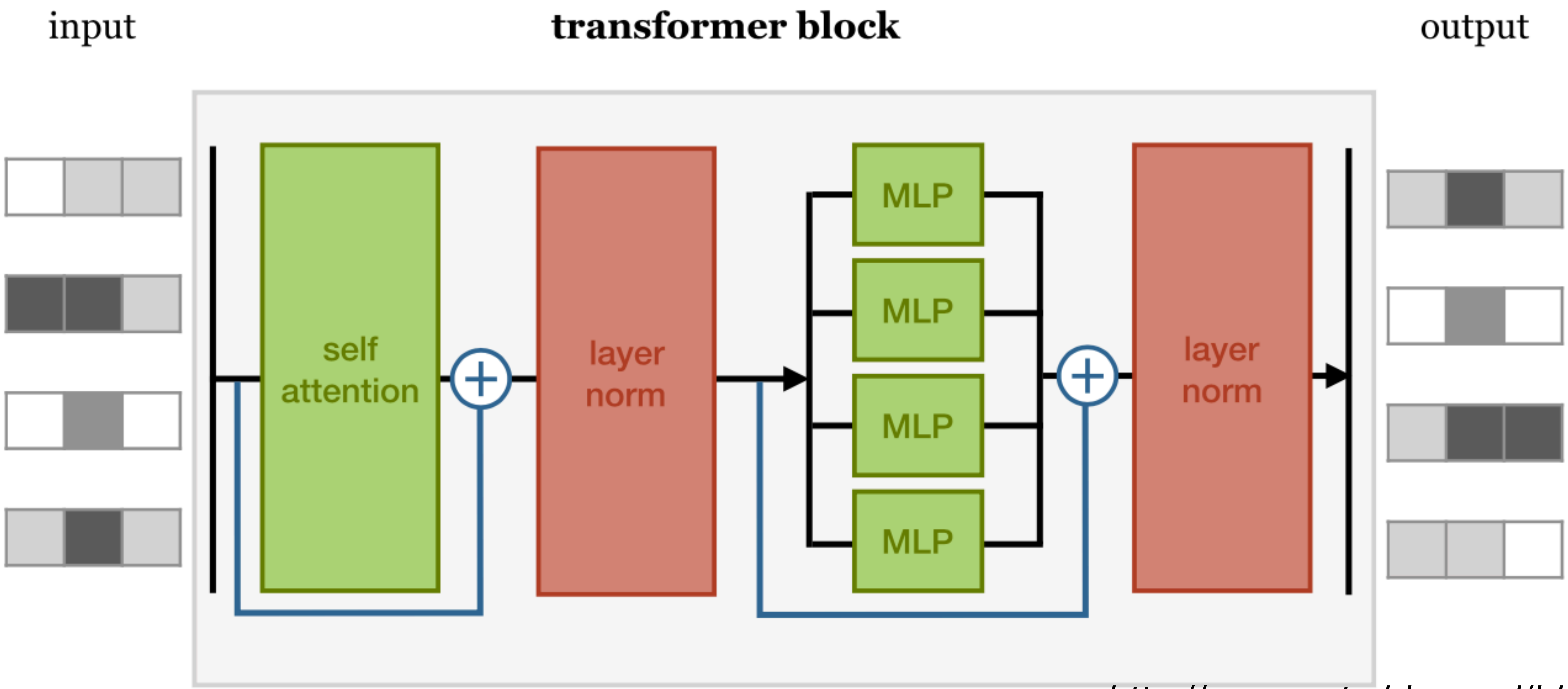
- Self-attention layer -> Layer normalization -> Dense layer



http://www.peterbloem.nl/blog/transformers

Transformer

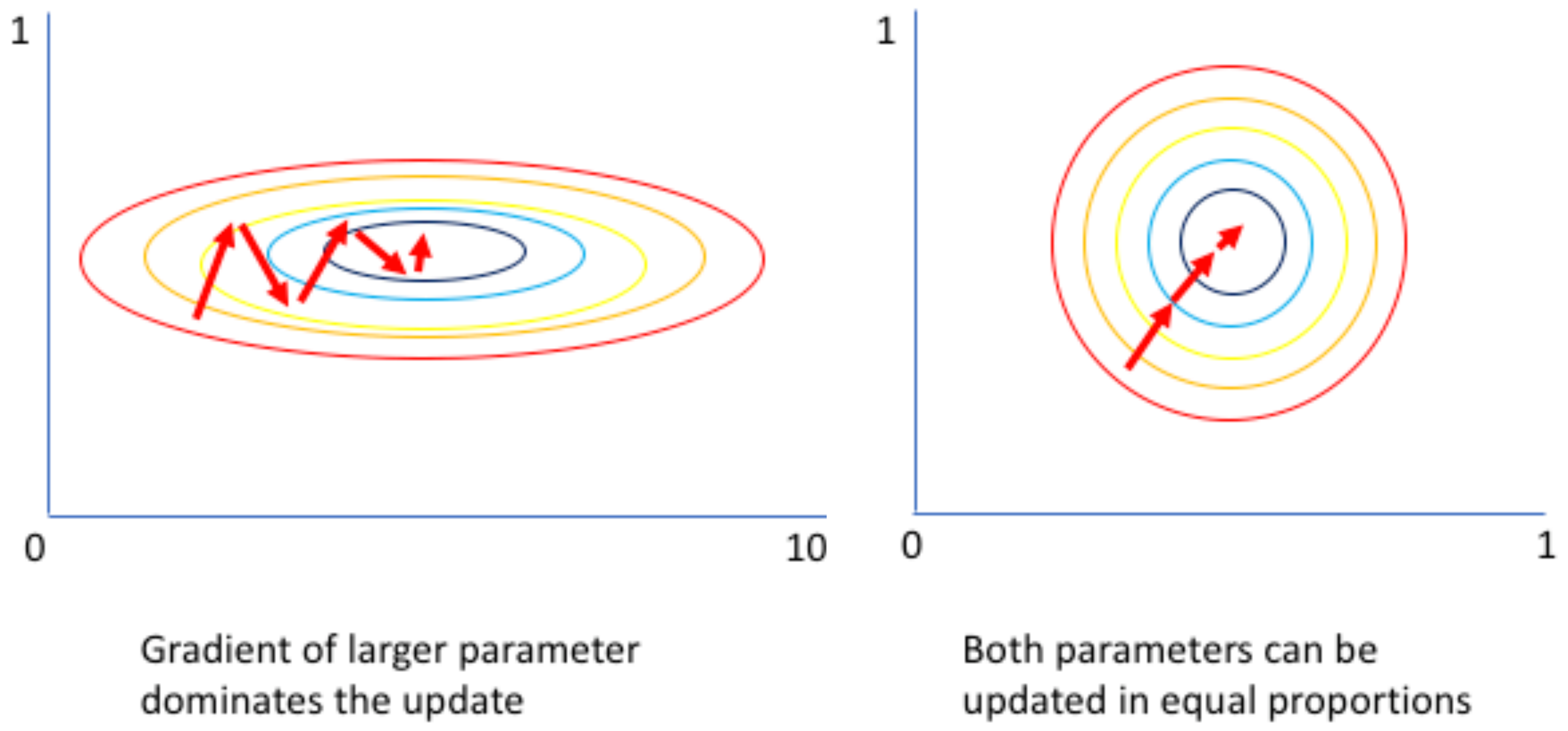
- Self-attention layer -> **Layer normalization** -> Dense layer

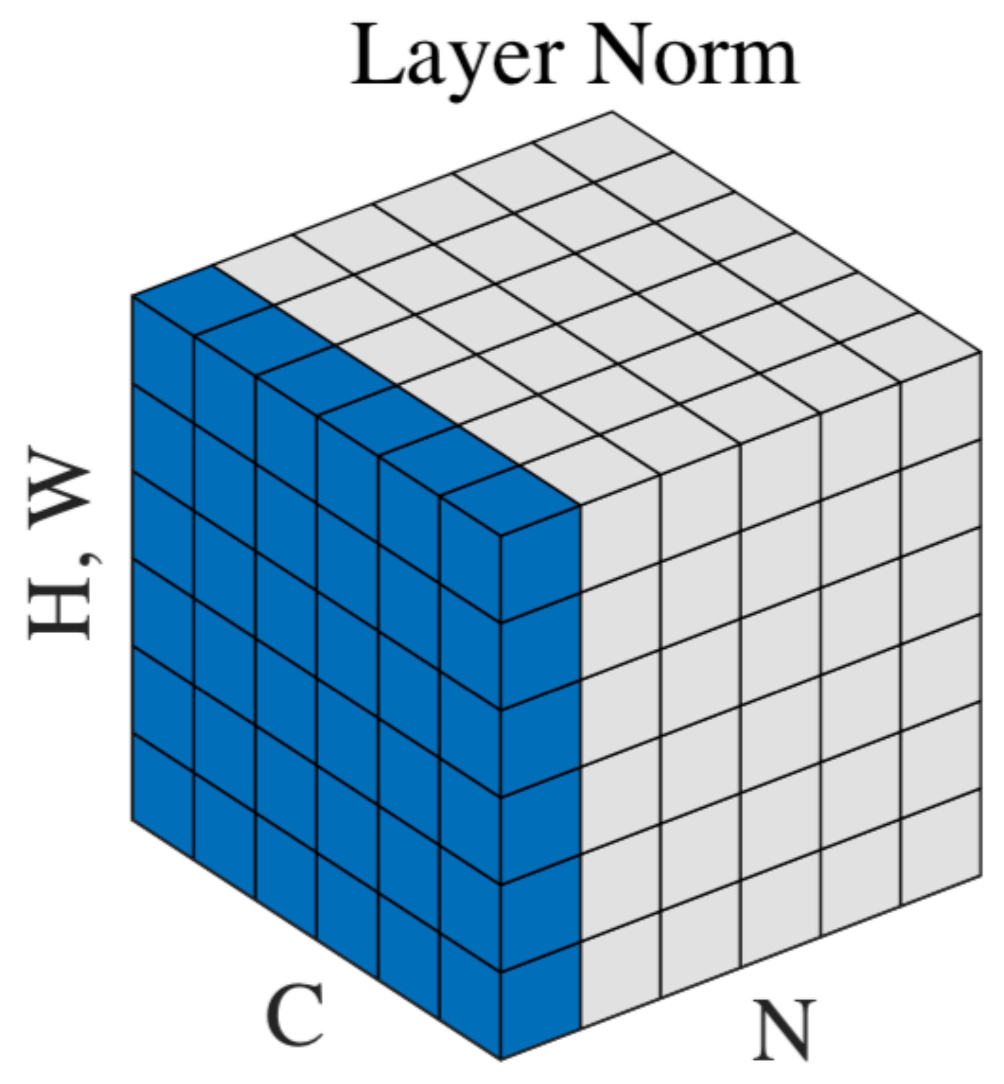


http://www.peterbloem.nl/blog/transformers

Layer Normalization

- Neural net layers work best when input vectors have uniform mean and std in each dimension
  - (remember input data scaling, weight initialization)
- As inputs flow through the network, means and std's get blown out.
- Layer Normalization is a hack to reset things to where we want them in between layers.

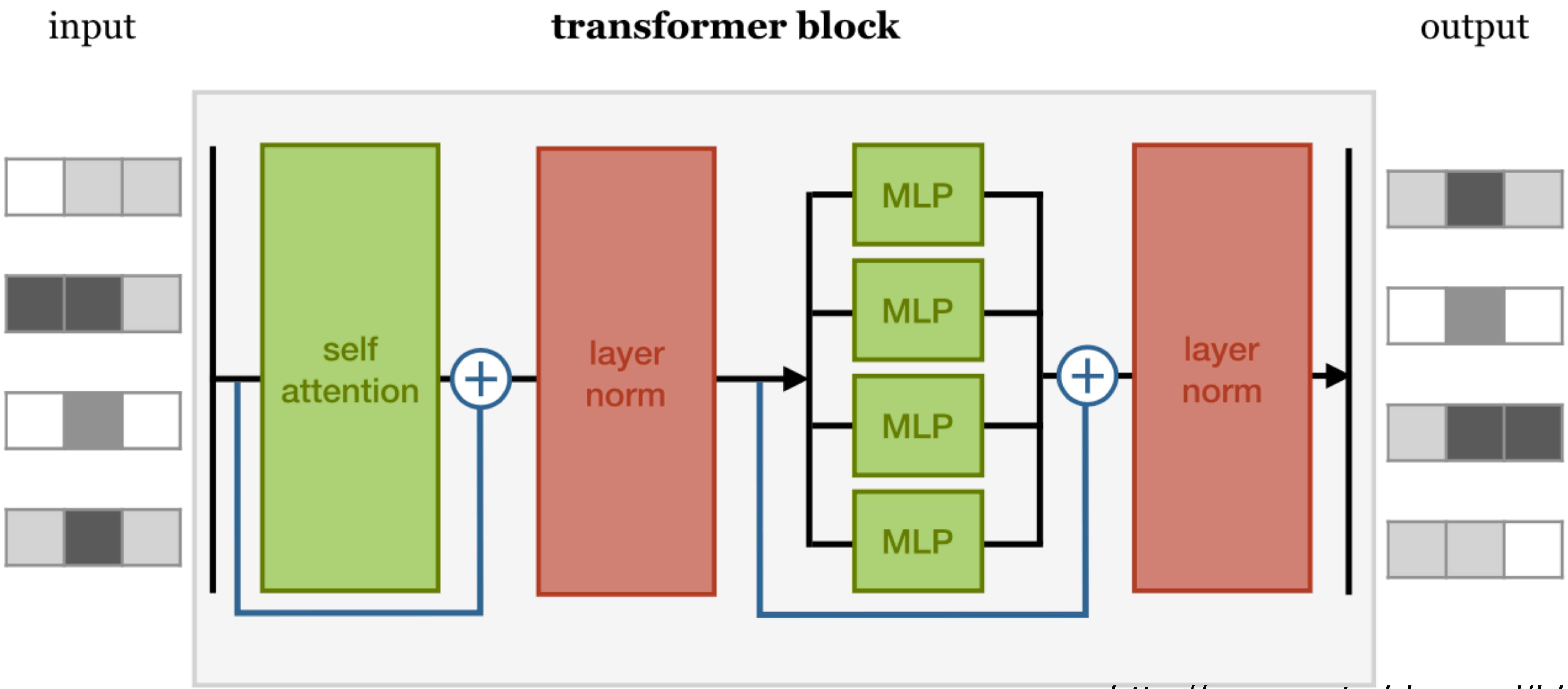




https://arxiv.org/pdf/1803.08494.pdf

Transformer

- Self-attention layer -> Layer normalization -> Dense layer



http://www.peterbloem.nl/blog/transformers

Transformer

- SO FAR:
  - Learned query, key, value weights
  - Multiple heads
  - Order of the sequence does not affect result of computations

http://www.peterbloem.nl/blog/transformers

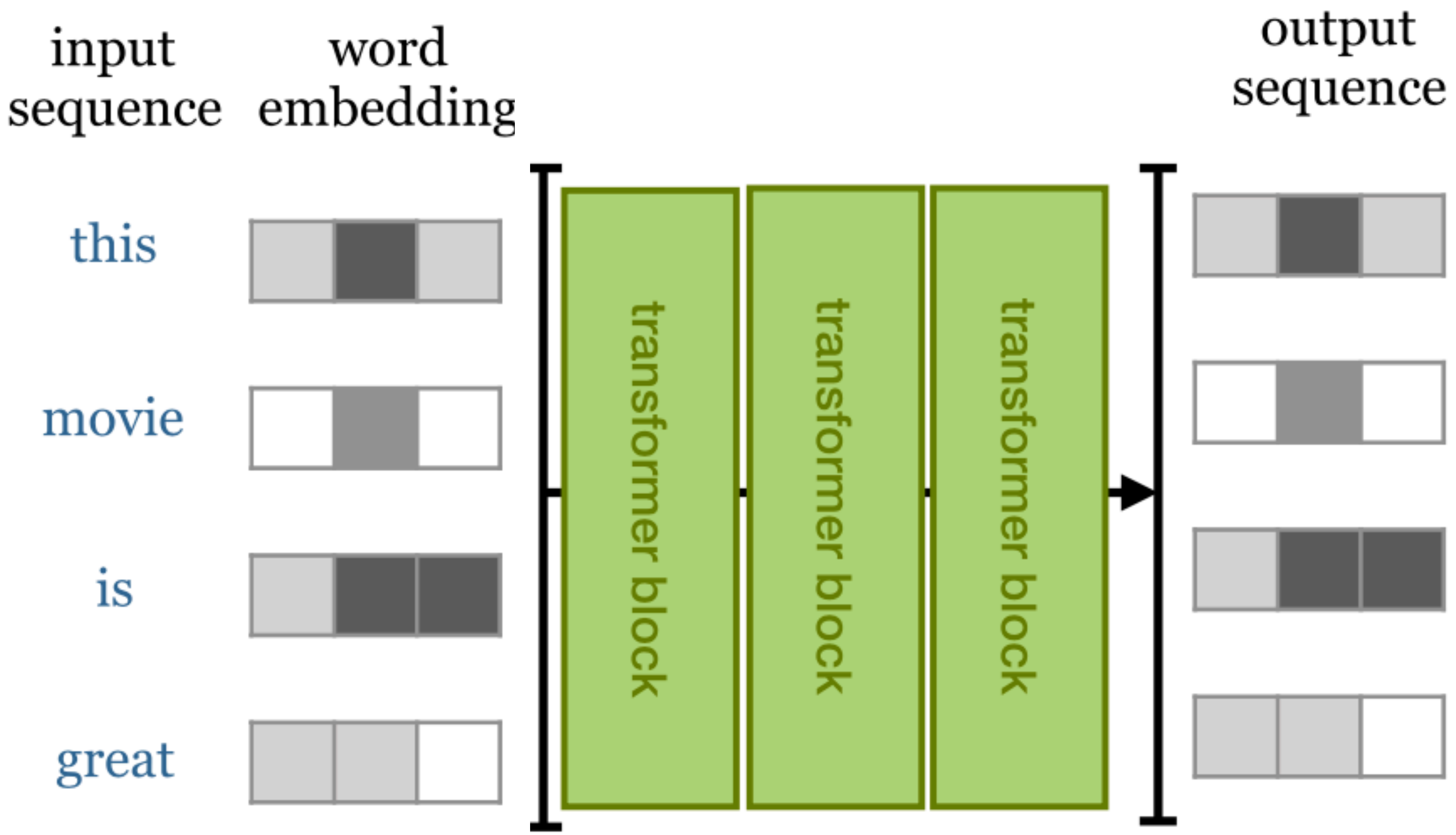
Transformer

- SO FAR:
  - Learned query, key, value weights
  - Multiple heads
  - **Order of the sequence does not affect result of computations**

Let's encode each vector with position

http://www.peterbloem.nl/blog/transformers

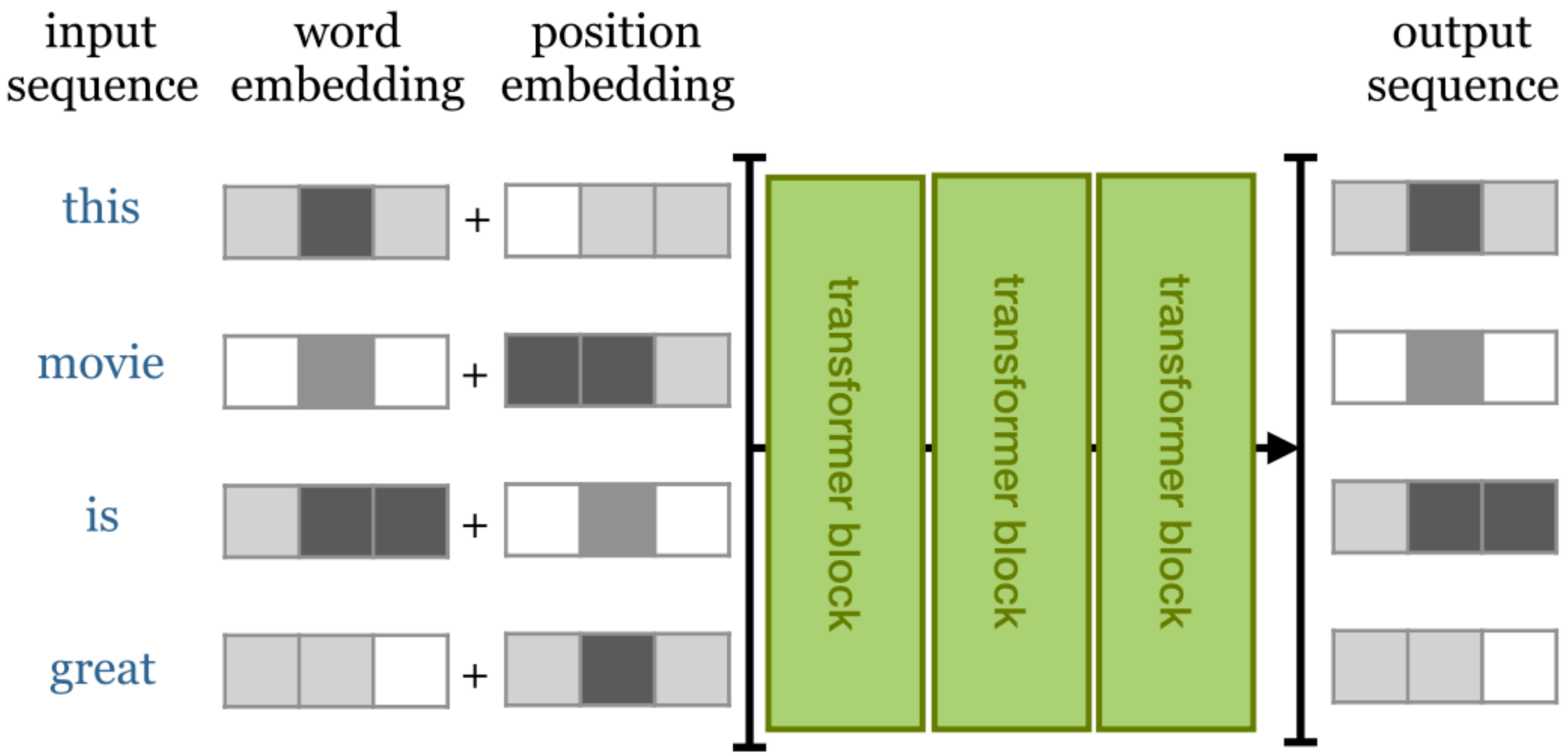
Transformer

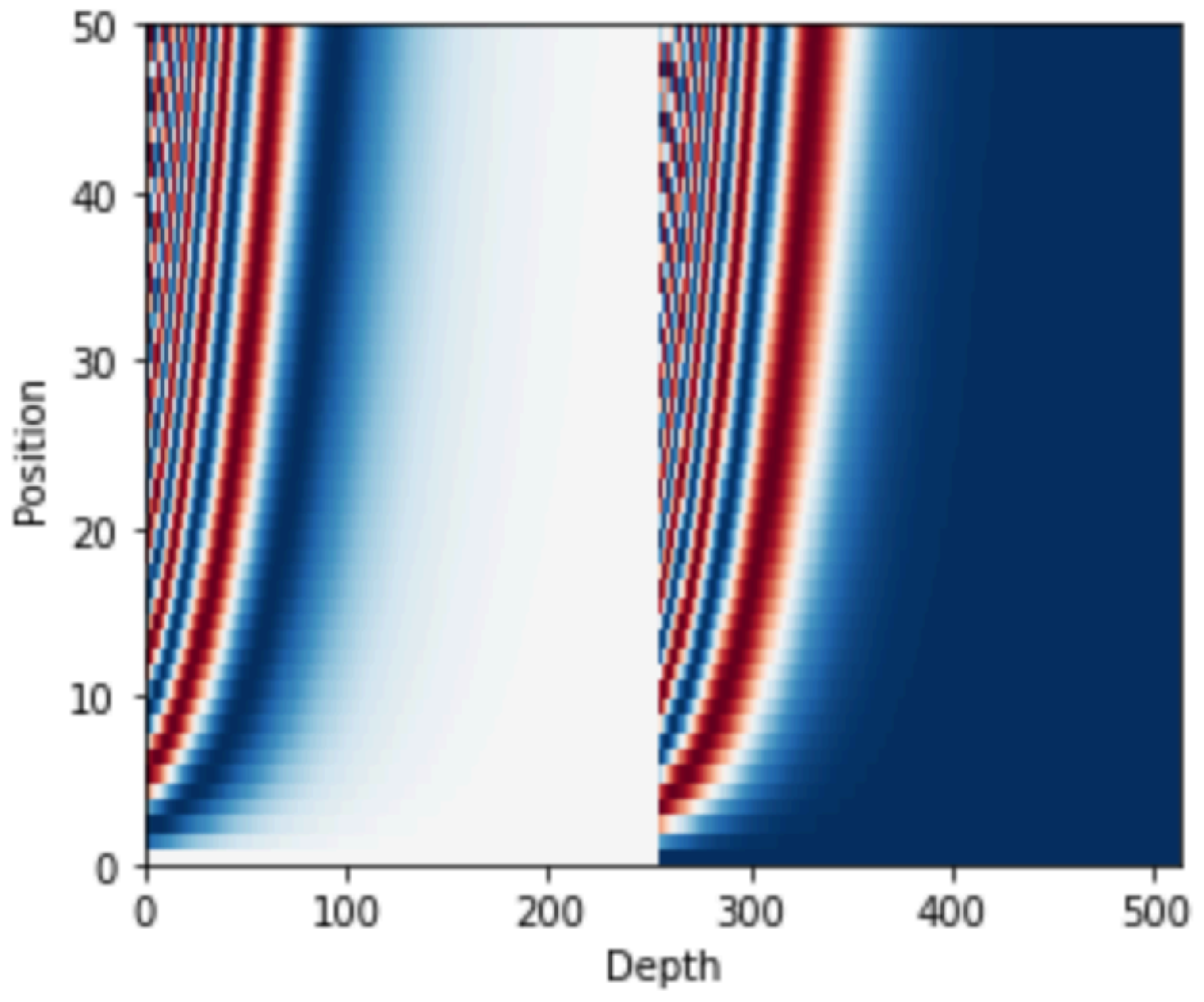


http://www.peterbloem.nl/blog/transformers

Transformer

- Position embedding: just what it sounds!

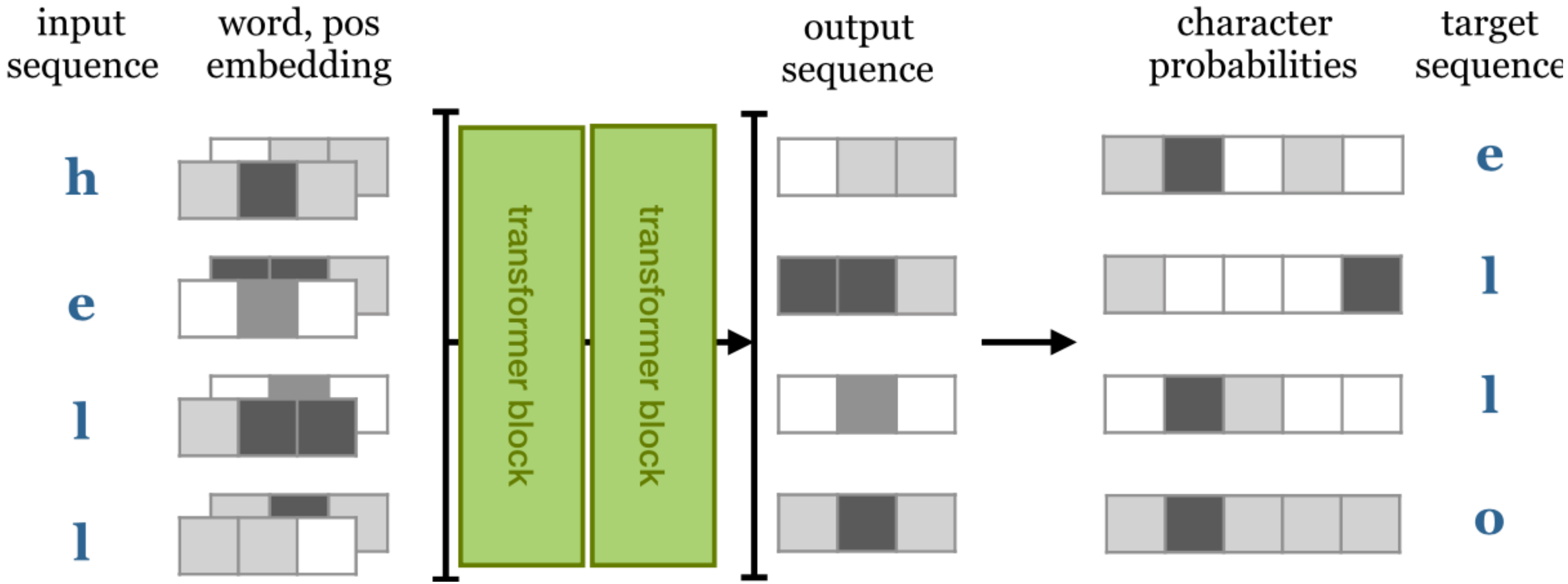




http://www.peterbloem.nl/blog/transformers

Transformer: last trick

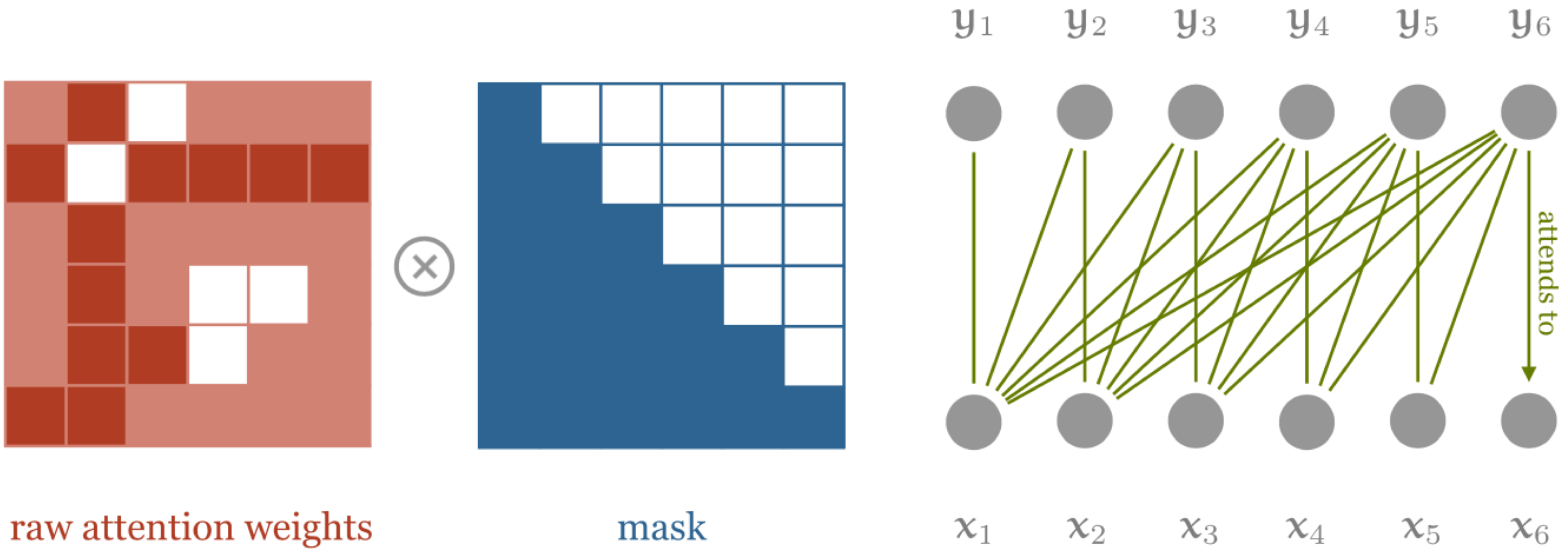
- Since the Transformer sees all inputs at once, to predict next vector in sequence (e.g. generate text), we need to mask the future.



http://www.peterbloem.nl/blog/transformers

Transformer: last trick

- Since the Transformer sees all inputs at once, to predict next vector in sequence (e.g. generate text), we need to mask the future.



http://www.peterbloem.nl/blog/transformers

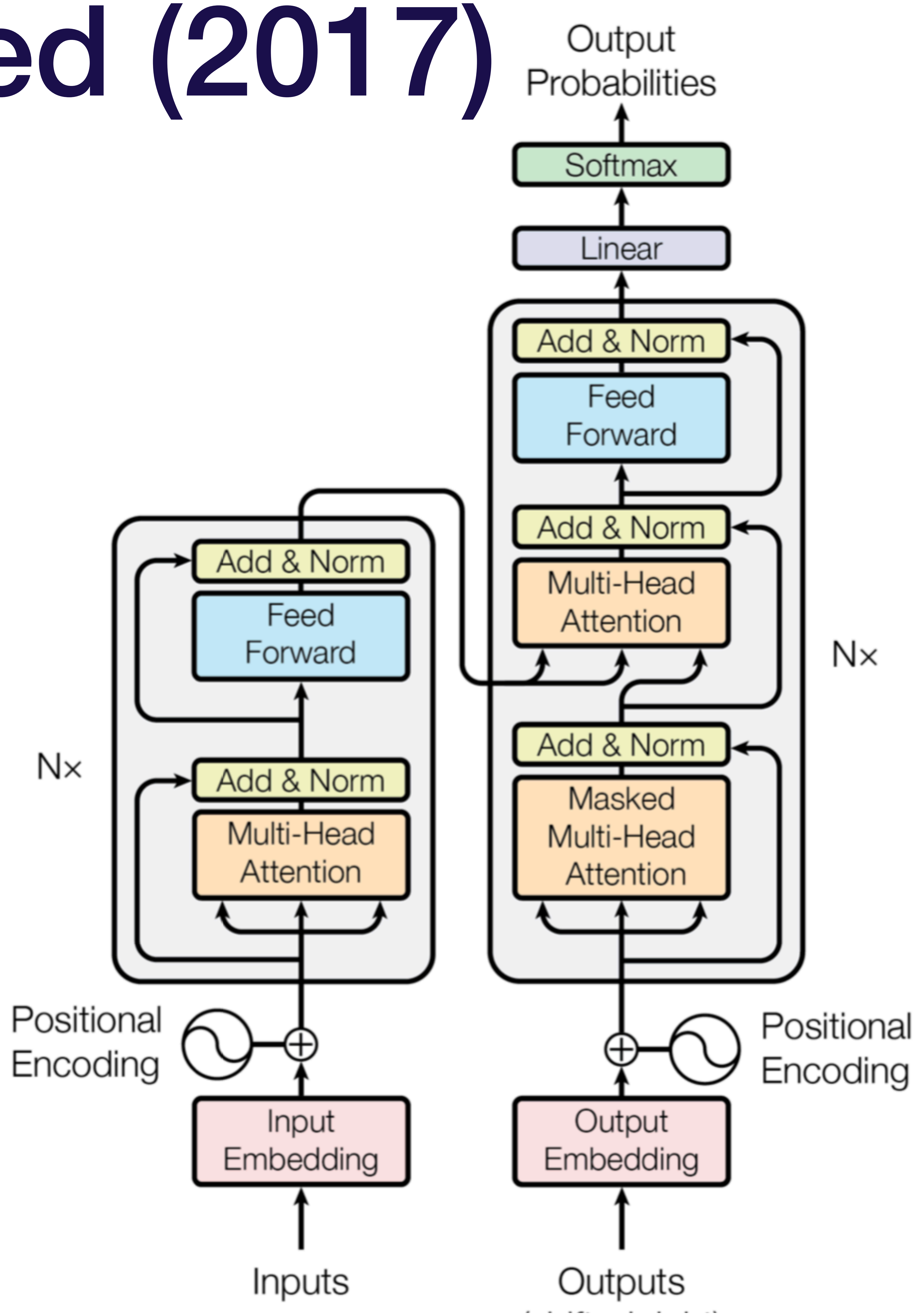
Questions?

Outline

- Transfer Learning in Computer Vision
- Embeddings and Language Models
- "NLP's ImageNet Moment": ELMO/ULMFit
- Transformers
  - Attention in detail
  - **BERT, GPT-2, DistillBERT, T5**

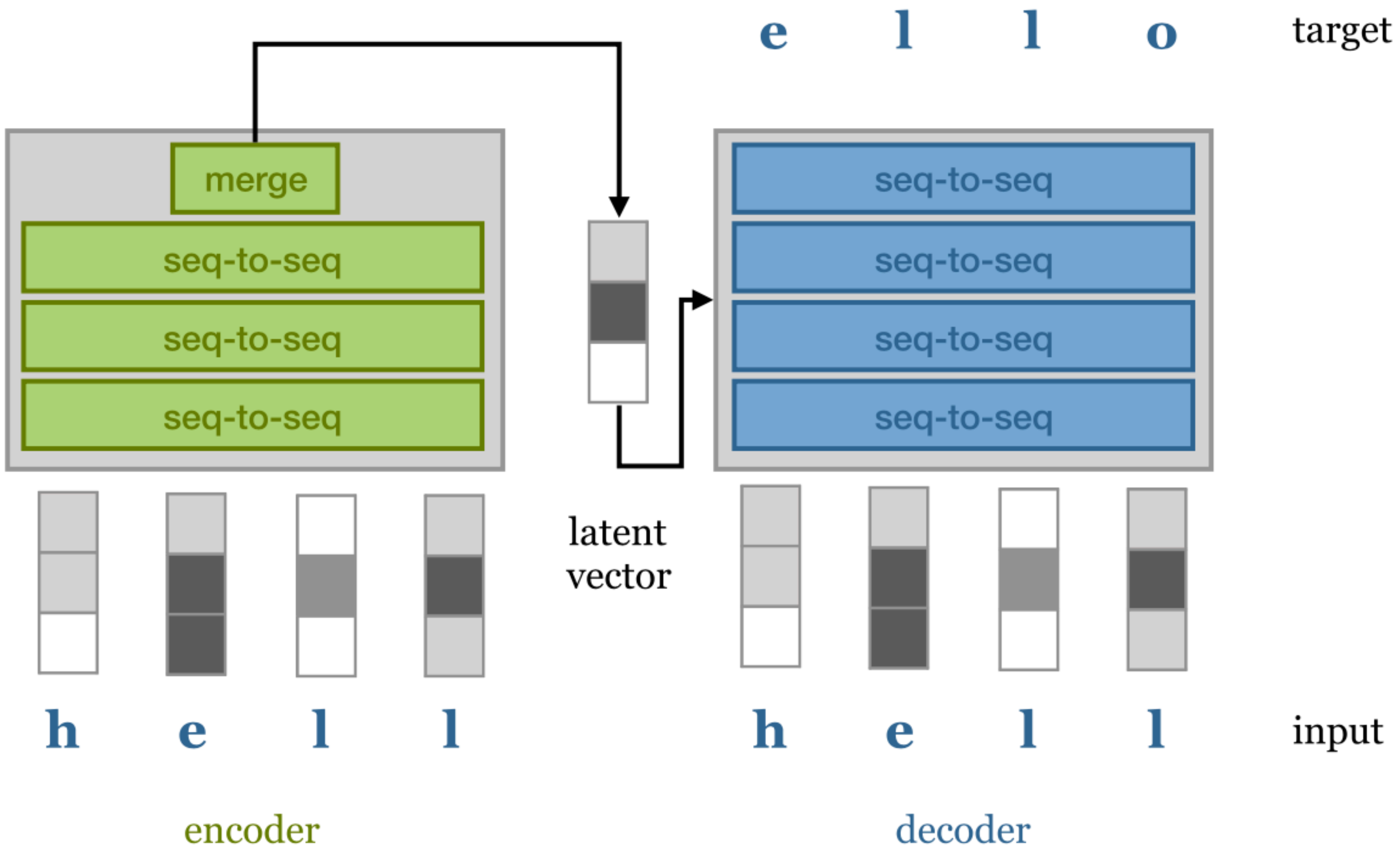
Attention is all you need (2017)

- Encoder-decoder for translation



Attention is all you need (2017)

- Encoder-decoder for translation
- Later models made it mostly just the encoder or just the decoder
- ...but then the latest models are back to encoder-decoder

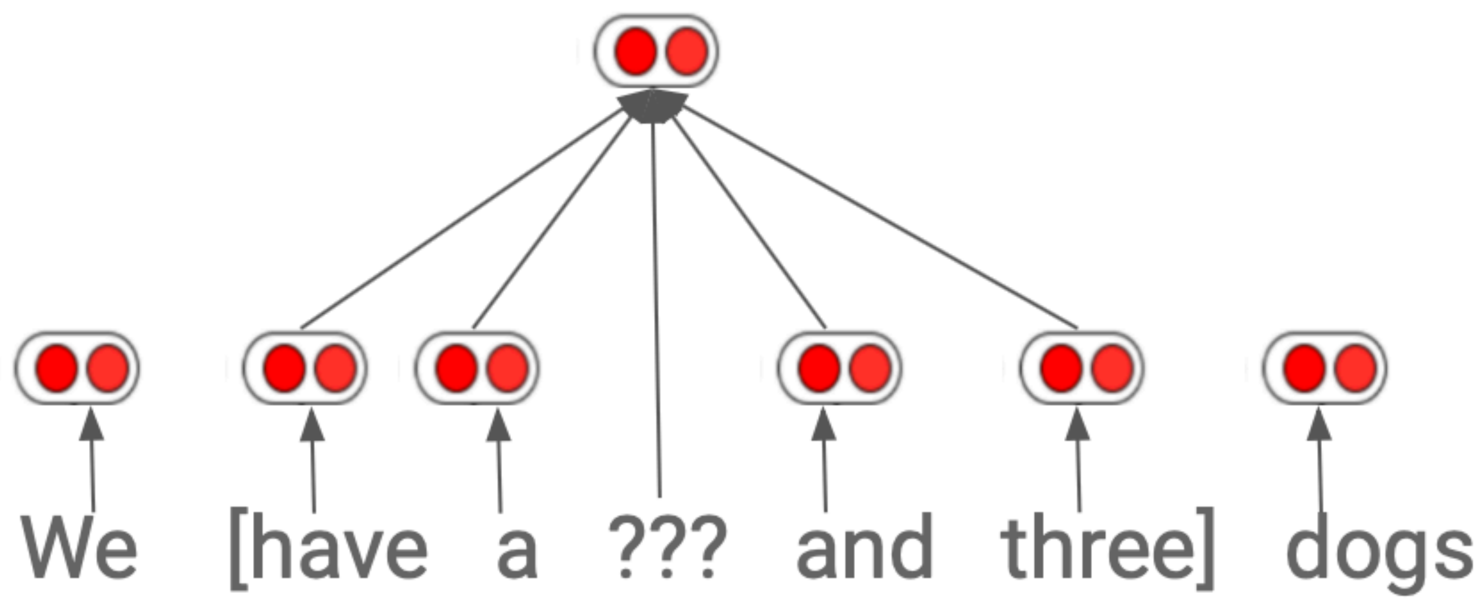


http://www.peterbloem.nl/blog/transformers

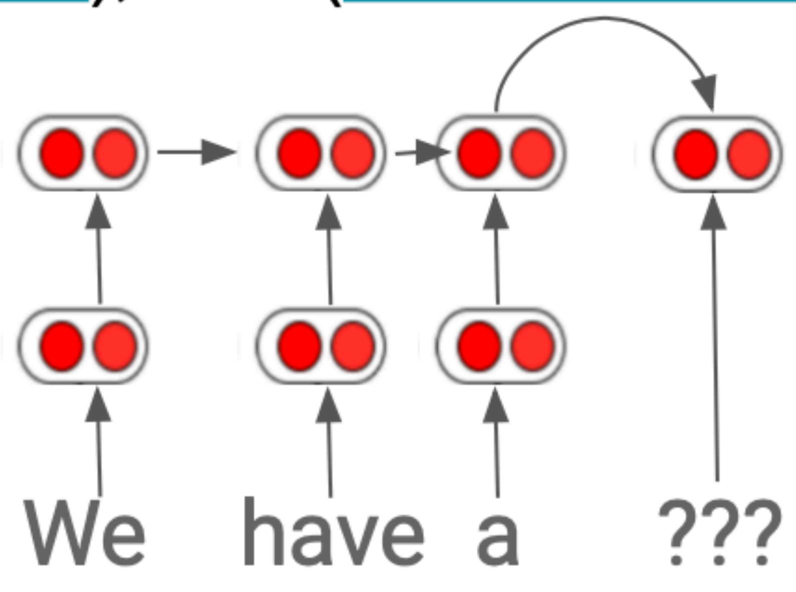
GPT / GPT-2

- Generative Pre-trained Transformer

word2vec, Mikolov et al (2013)



ELMo, Peters et al. 2018, ULMFiT (Howard & Ruder 2018), GPT (Radford et al. 2018)



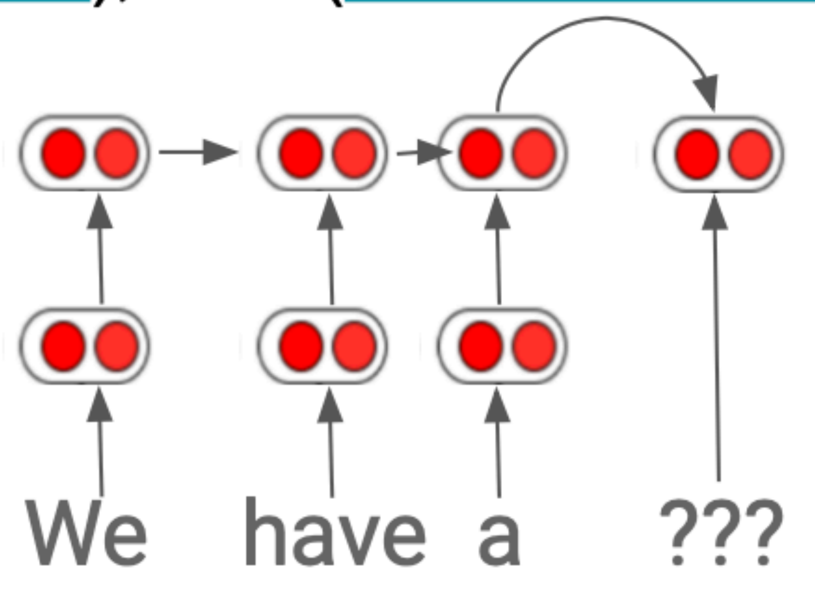
https://docs.google.com/presentation/d/1flhGikFPnb7G5kr58OvYC3GN4io7MznnM0aAgadvJfc

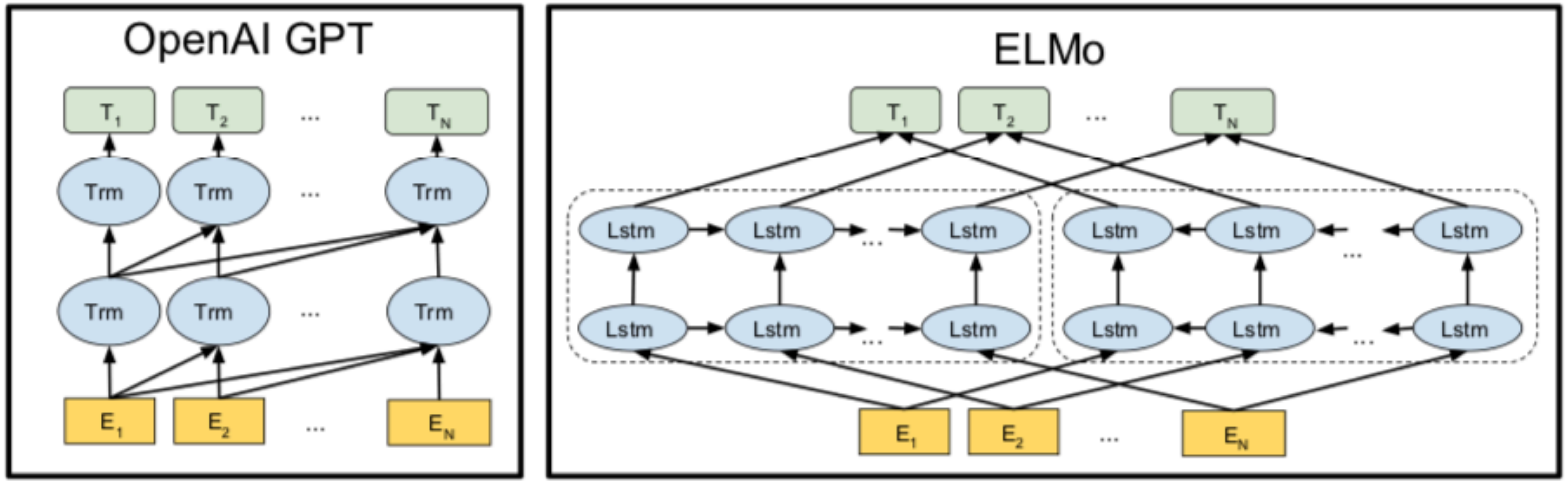
GPT / GPT-2

- Generative Pre-trained Transformer

- GPT learns to predict the next word in the sequence, just like ELMO or ULMFIT

ELMo, Peters et al. 2018, ULMFiT (Howard & Ruder 2018), GPT (Radford et al. 2018)

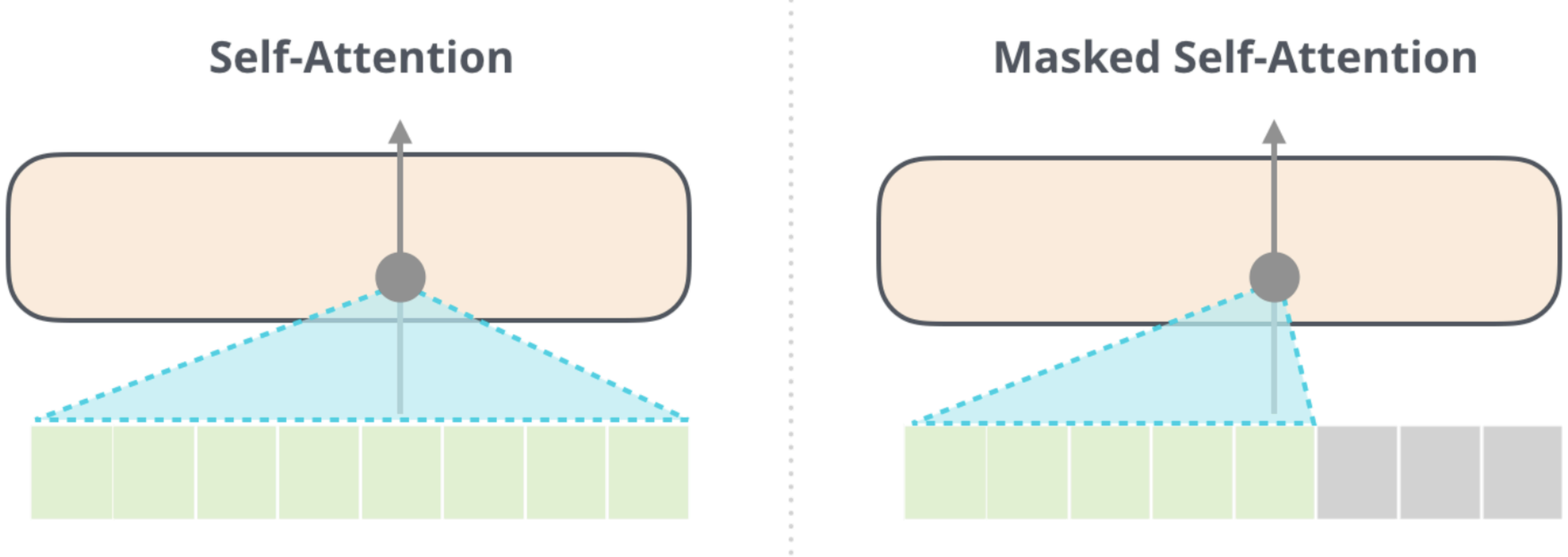




https://arxiv.org/pdf/1810.04805.pdf

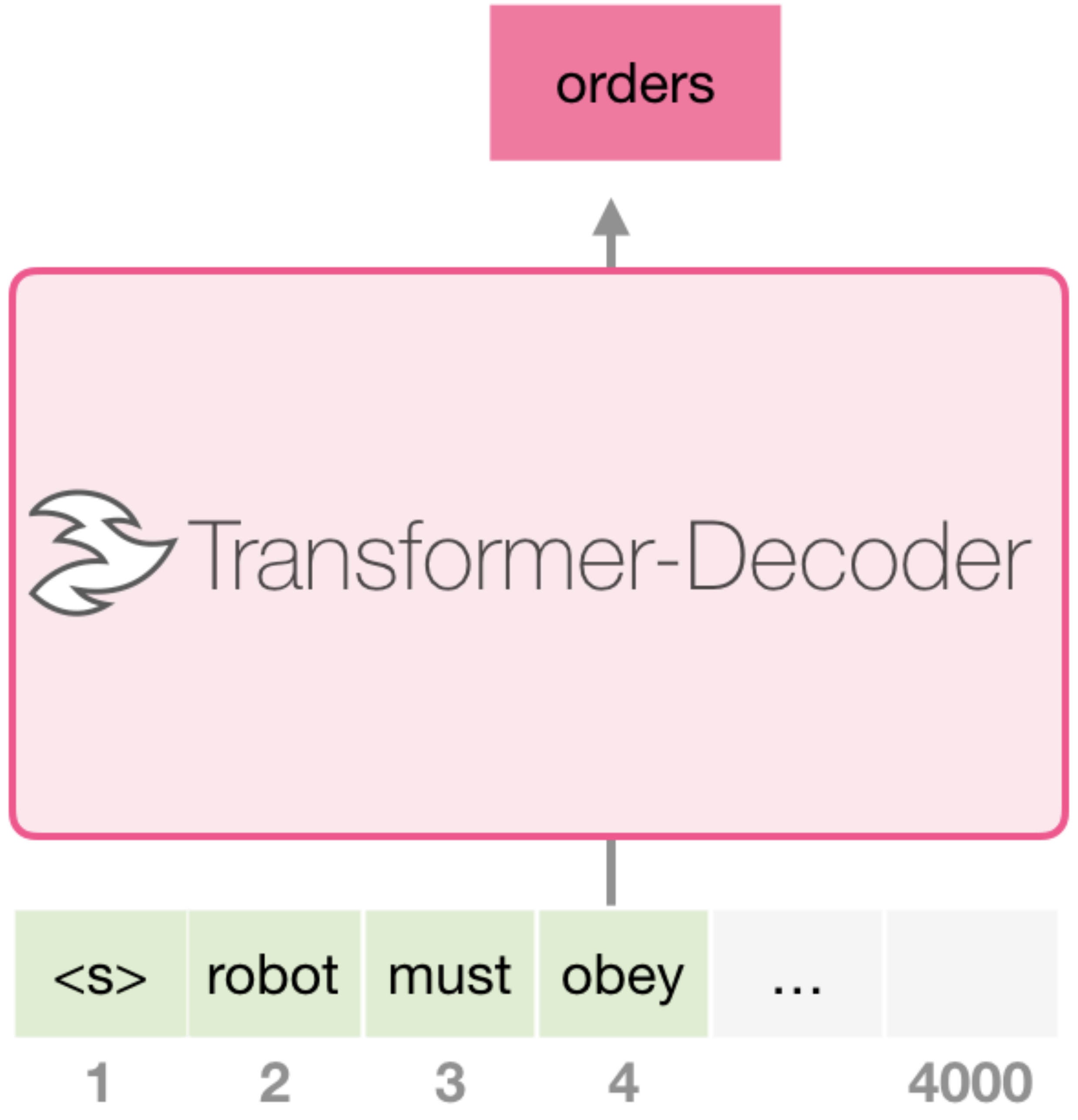
GPT / GPT-2

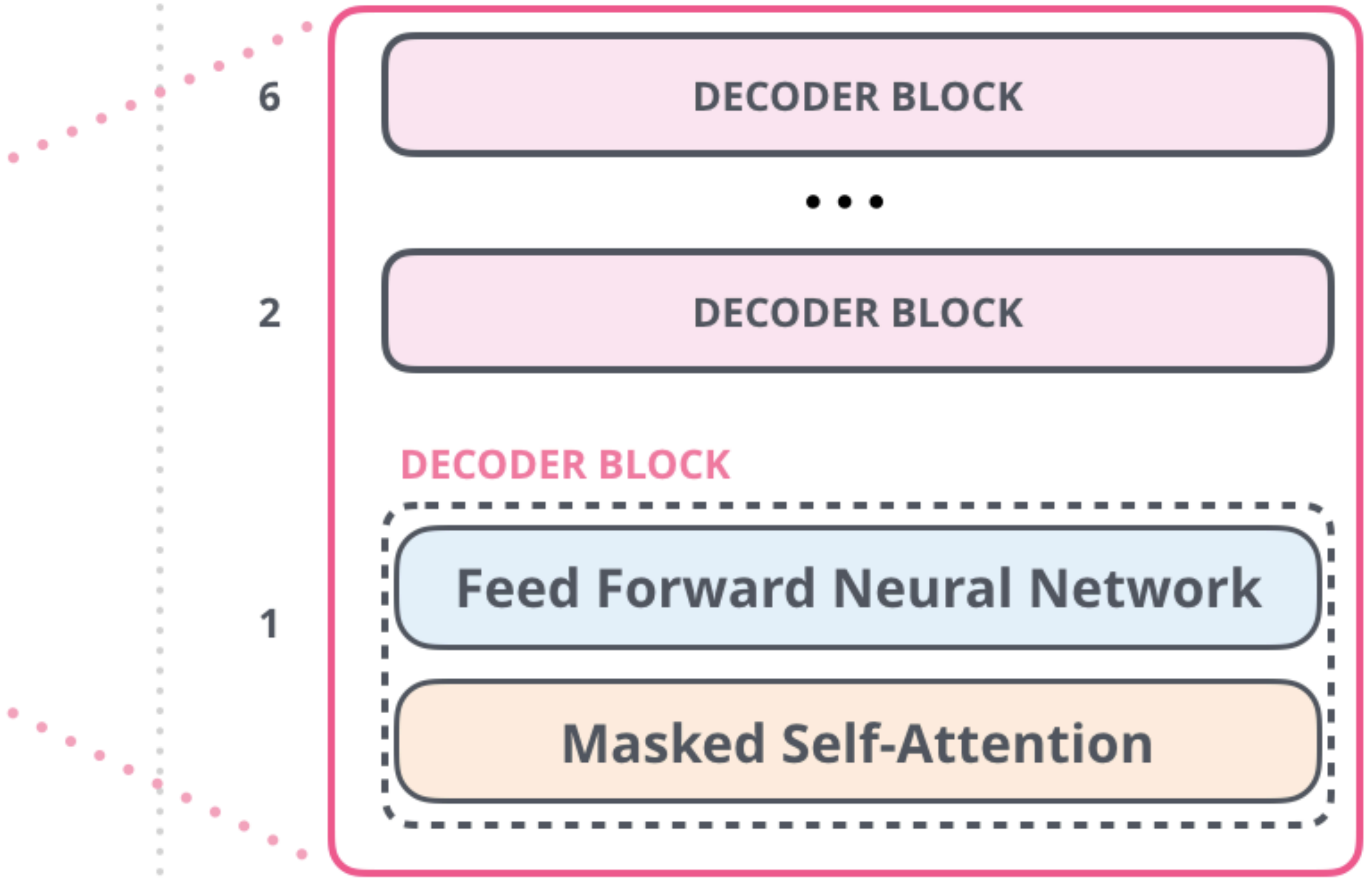
- Generative Pre-trained Transformer
- GPT learns to predict the next word in the sequence, just like ELMO or ULMFIT
- Since it conditions only on preceding words, it uses masked self-attention



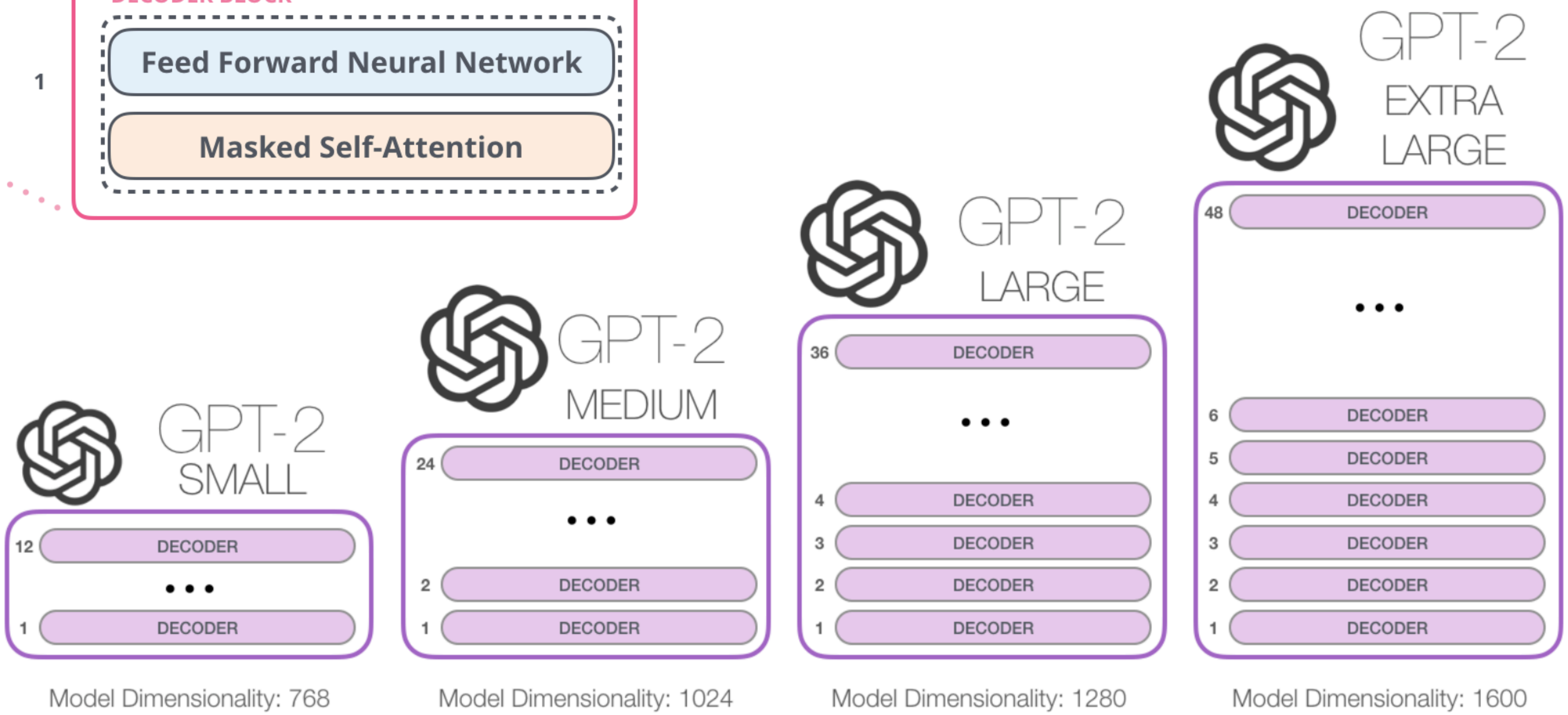
http://jalammar.github.io/illustrated-gpt2/

GPT / GPT-2





Trained on 8M web pages



117M params | 345M params | 762M params | 1.5B params

http://jalammar.github.io/illustrated-gpt2/

Talk to Transformer

Talk to Transformer

See how a modern neural network completes your text. Type a custom snippet or try one of the examples. This is a limited demo of InferKit.

Custom prompt

Full Stack Deep Learning is a UC Berkeley course about

Generate Another

Completion

Full Stack Deep Learning is a UC Berkeley course about deep learning and neural networks, first taught at HCS in Fall 2016, and then at HCS in Spring 2017.

This course requires a light to moderate understanding of computer science, preferably CS for computer science majors.

The class provides an introduction to convolutional neural networks and other deep learning models, including batch learning and software engineering.

The course code was updated to use the tensorflow version of the open source tensorflow for deep learning packages, which

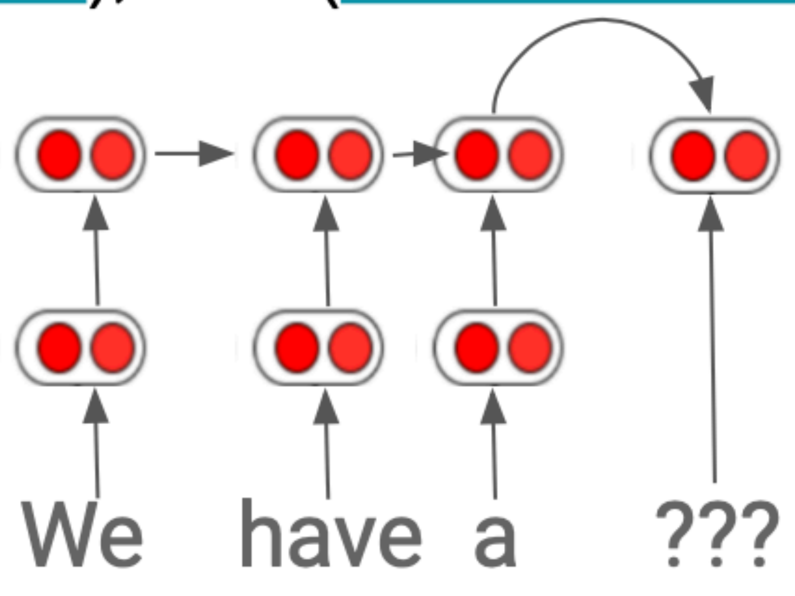
https://talktotransformer.com

1.5B parameter model

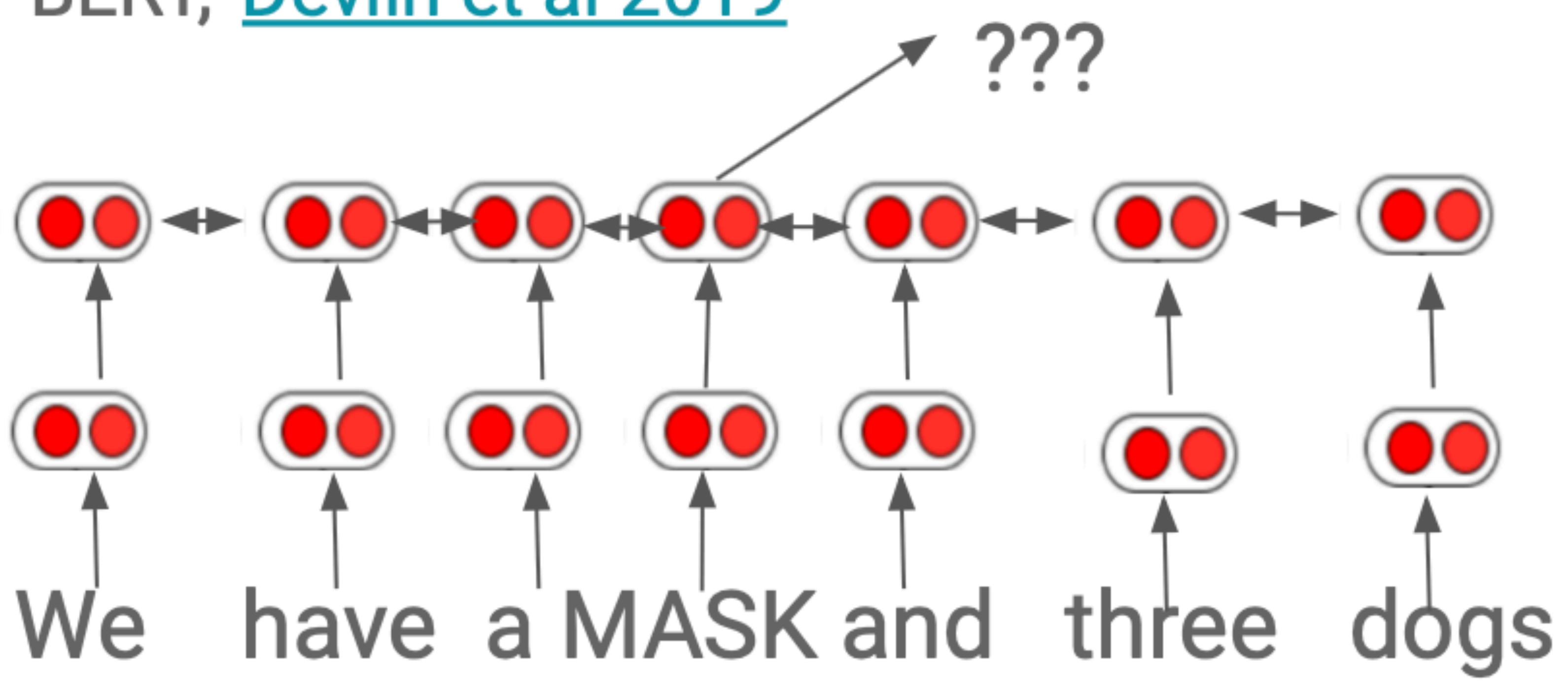
BERT

- *Bidirectional* Encoder Representations from Transformers
- Encoder blocks only (no masking)
- BERT involves pre-training on A LOT of text with 15% of all words masked out
    - also sometimes predicting whether one sentence follows another

ELMo, Peters et al. 2018, ULMFiT (Howard & Ruder 2018), GPT (Radford et al. 2018)



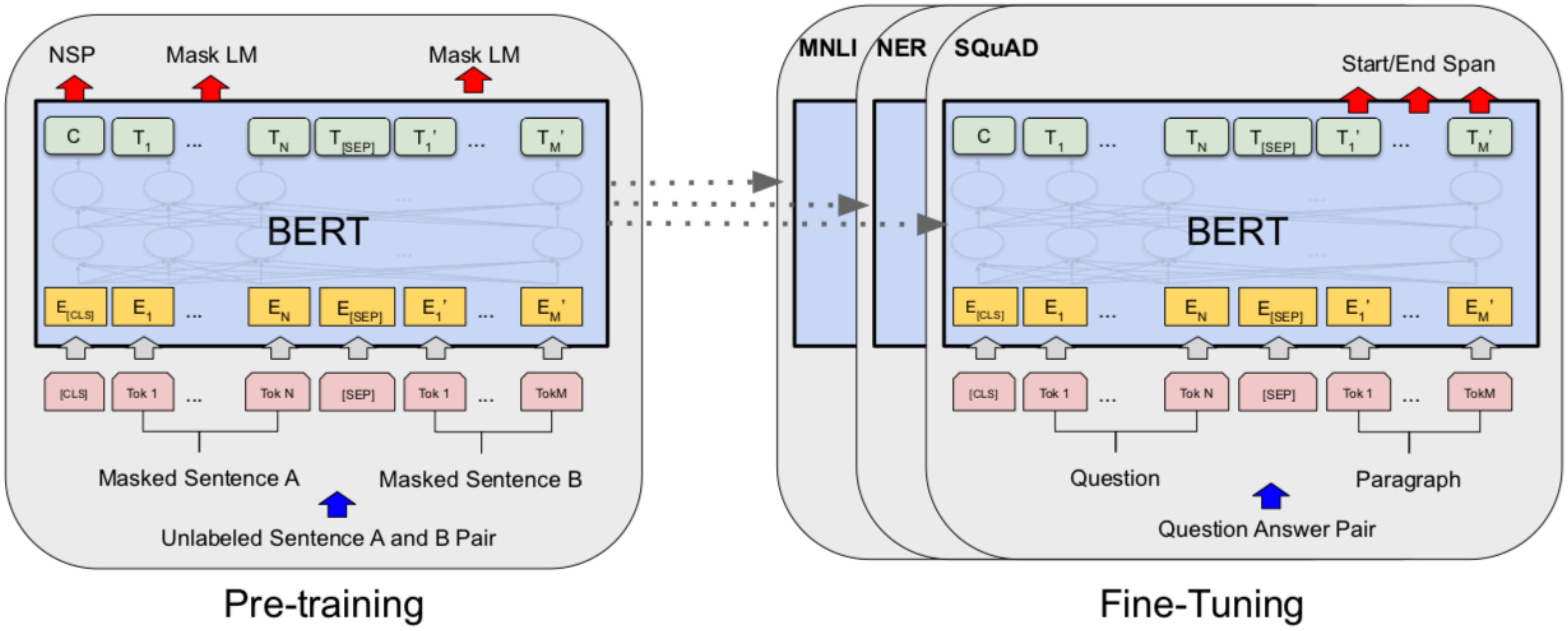
BERT, Devlin et al 2019



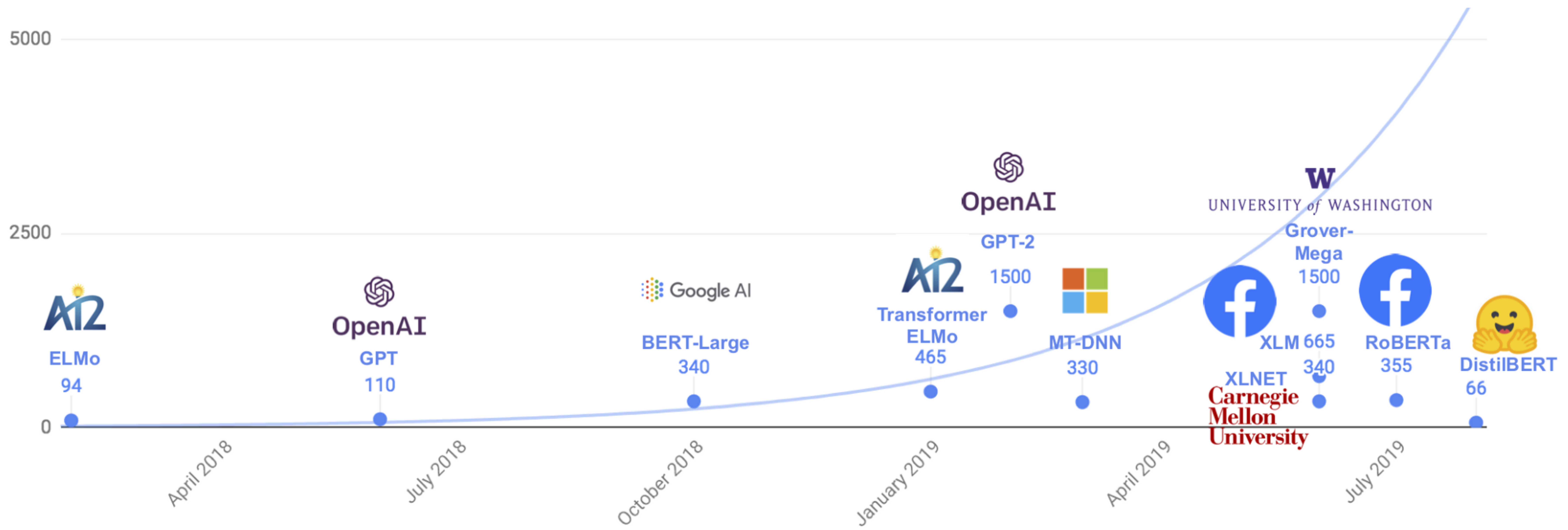
https://docs.google.com/presentation/d/1flhGikFPnb7G5kr58OvYC3GN4io7MznnM0aAgadvJfc

BERT

- 340M parameters:
  - 24 transformer blocks, embedding dim of 1024, 16 attention heads

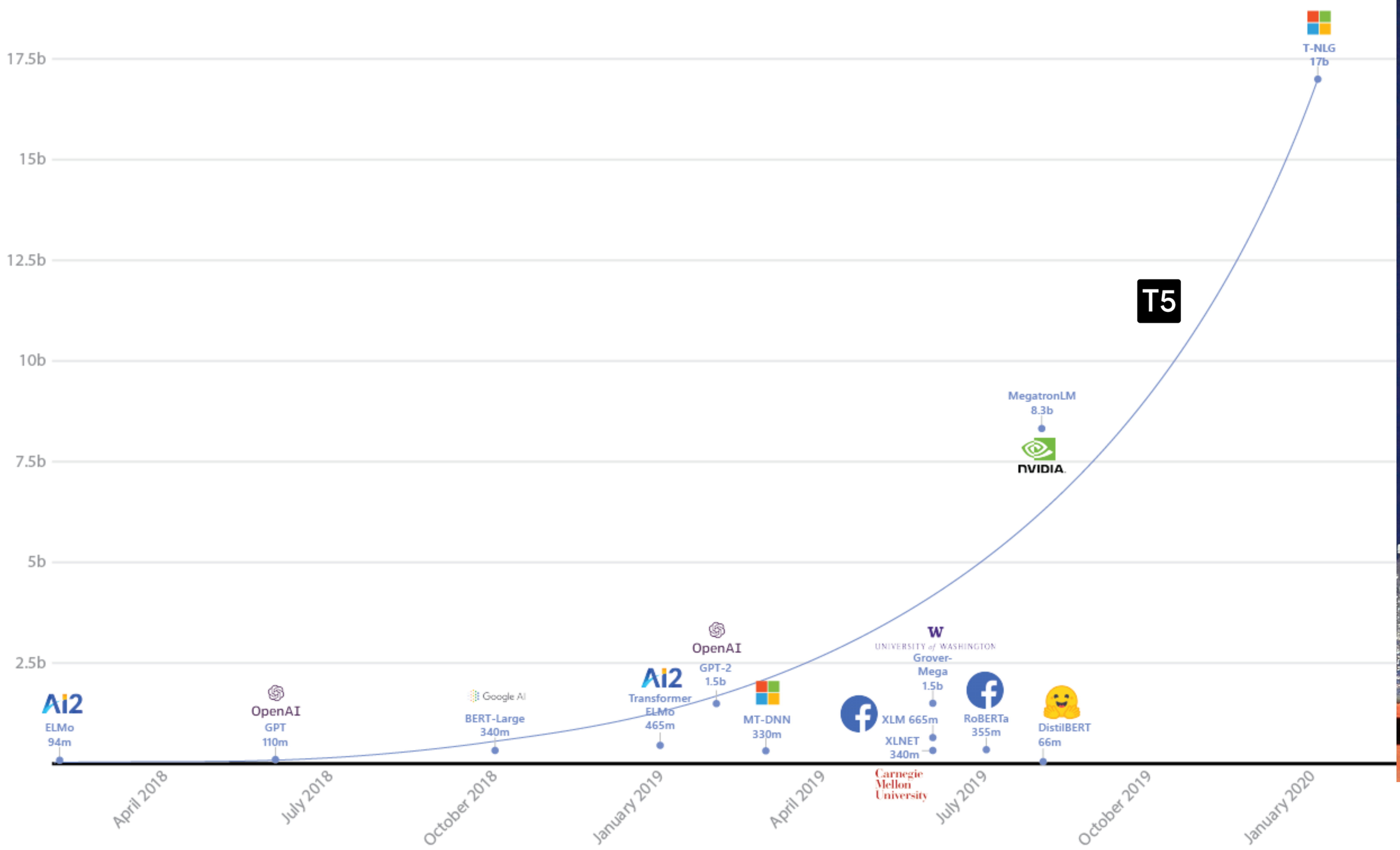


Transformers



https://medium.com/huggingface/encoder-decoders-in-transformers-a-hybrid-pre-trained-architecture-for-seq2seq-af4d7bf14bb8

Number of parameters



https://www.microsoft.com/en-us/research/blog/turing-nlg-a-17-billion-parameter-language-model-by-microsoft/

T5: Text-to-Text Transfer Transformer

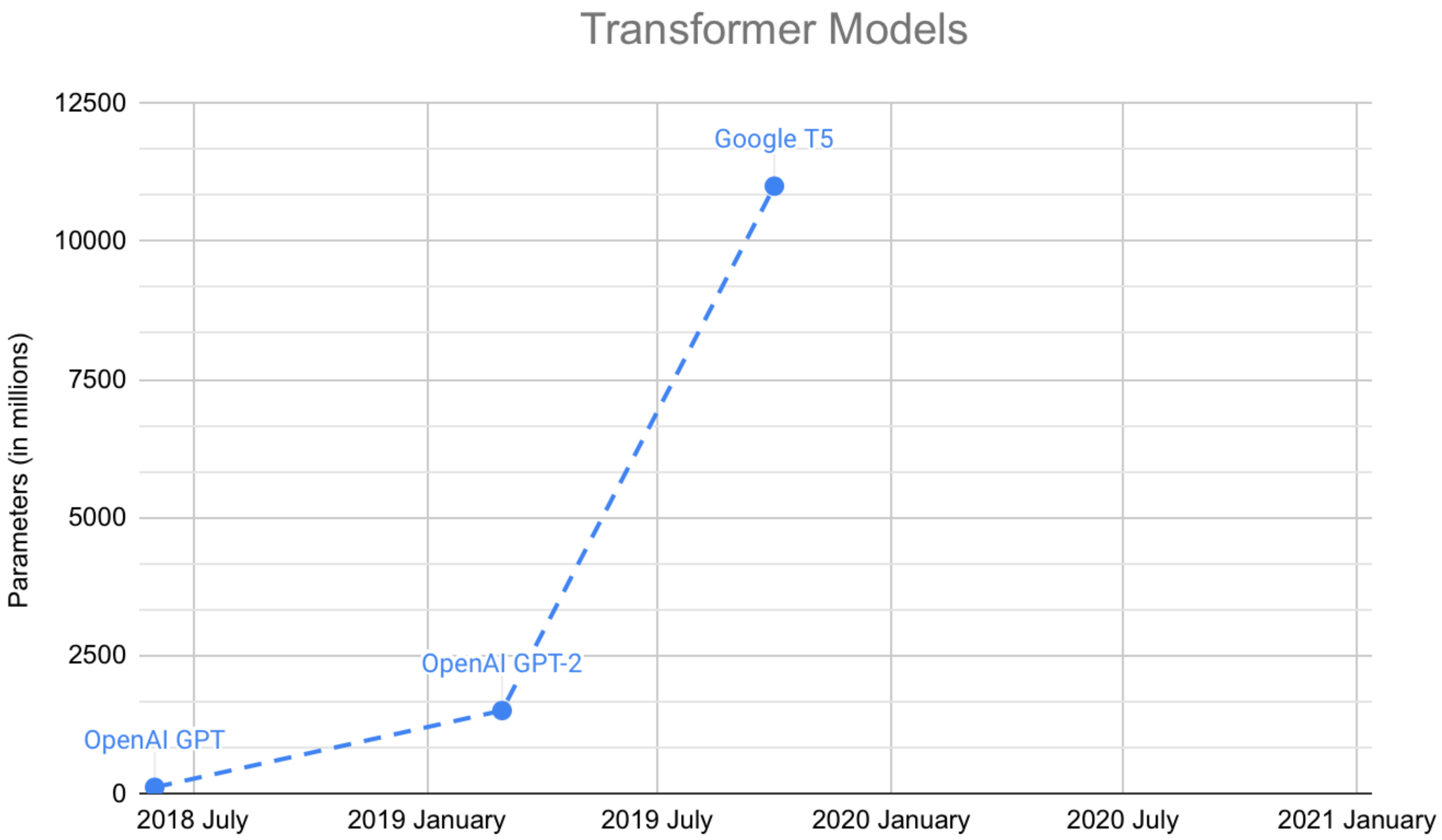
- Feb 2020
- Evaluated most recent transfer learning techniques
- Input and output are both text strings
- Trained on C4 (Colossal Clean Crawled Corpus) - 100x larger than Wikipedia
- 11B parameters
- SOTA on GLUE, SuperGLUE, SQuAD

https://ai.googleblog.com/2020/02/exploring-transfer-learning-with-t5.html

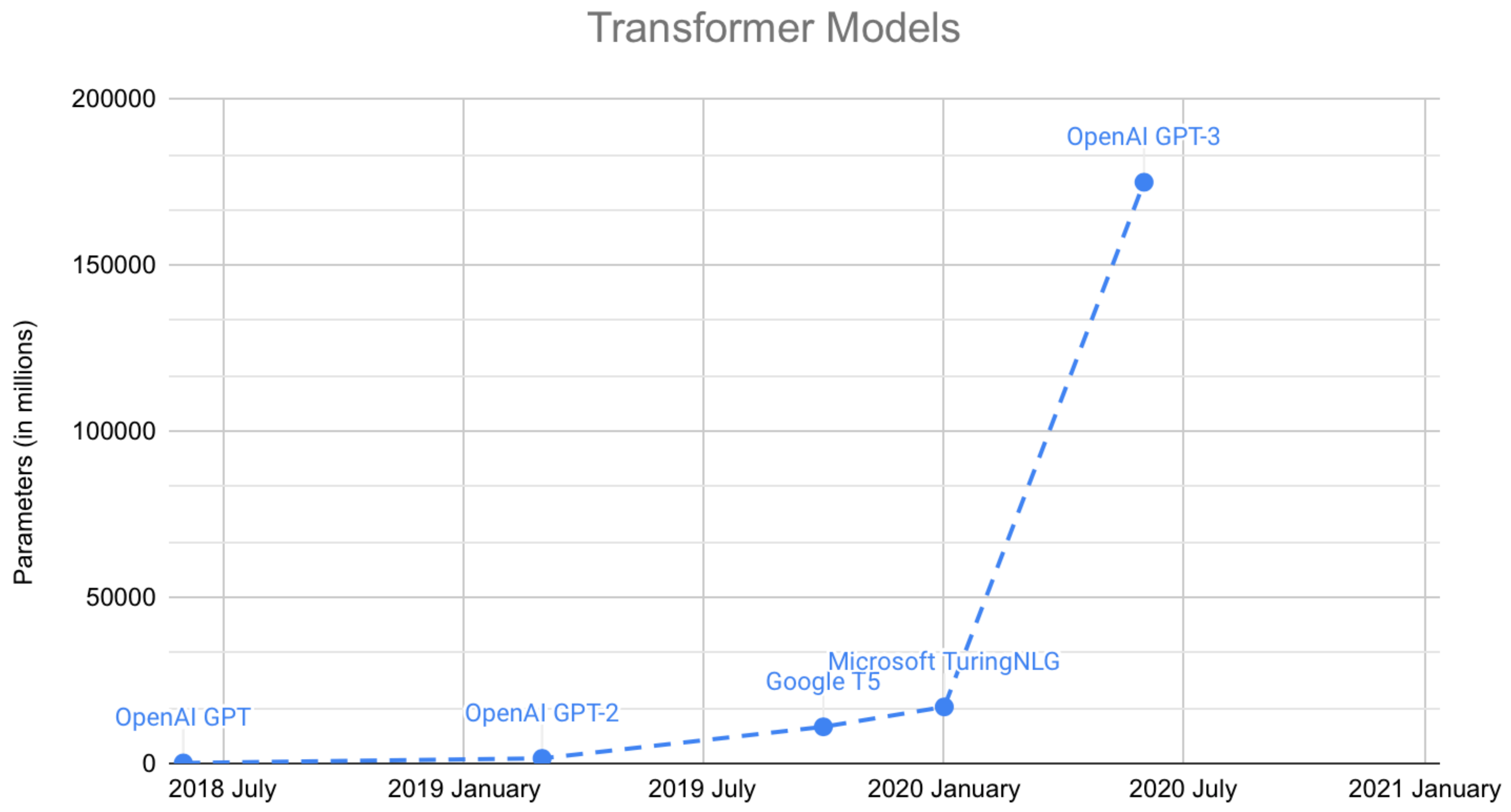
T5: Text-to-Text Transfer Transformer

- Encoder-decoder setup, instead of BERT's encoder only. Meaning that the model handles translation, summarization, and multi-token answers for clozure tasks — much more gracefully than BERT.
- Comprehensive study and support for multi-word masking [imo this is very important, as BERT can't directly answer multi-word clozure answers without hacks and modification; yes **most** questions have one word answers in a large vocabulary, but many do not, including the most significant answers in your niche downstream problem]
- Multi-task finetuning by defining all downstream tasks as Q&A. It's not better, but also not worse (and you don't need a new architecture add-on for every new task!)
- Definitely shows that encoder-decoder is better than encoder-only or decoder-only across a basket of downstream tasks. I am very convinced.

https://medium.com/@Moscow25/the-best-deep-natural-language-papers-you-should-read-bert-gpt-2-and-looking-forward-1647f4438797


Transformer Models
Parameters (in millions)
12500
10000
7500
5000
2500
0
2018 July
2019 January
2019 July
2020 January
2020 July
2021 January
OpenAI GPT
OpenAI GPT-2
Google T5

GPT-3


Transformer Models
Parameters (in millions)
200000
150000
100000
50000
0
2018 July
2019 January
2019 July
2020 January
2020 July
2021 January
OpenAI GPT
OpenAI GPT-2
Google T5
Microsoft TuringNLG
OpenAI GPT-3

OpenAI API Beta

FIND OUT MORE LOGIN

OpenAI technology, just an HTTPS call away

Apply our API to any language task — semantic search, summarization, sentiment analysis, content generation, translation, and more — with only a few examples or by specifying your task in English.

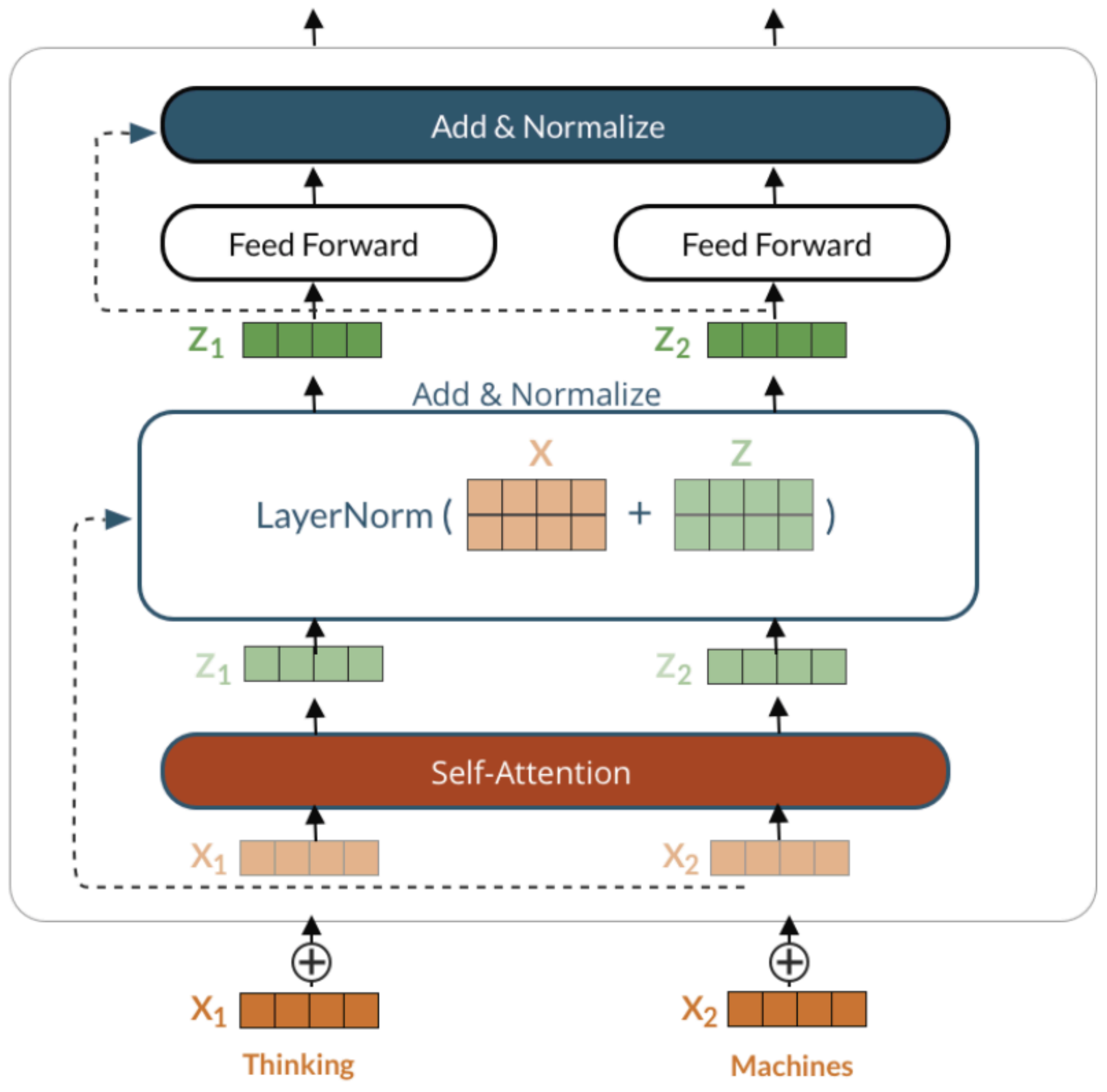
Text generation

```
import openai

prompt = """We're releasing an API for accessing new
AI models developed by OpenAI. Unlike most AI systems
which are designed for one use-case, the API today
provides a general-purpose "text in, text out"
interface, allowing users to try it on virtually any
English language task. You can now request access in
order to integrate the API into your product, develop
an entirely new application, or help us explore the
strengths and limits of this technology."""

response = openai.Completion.create(model="davinci",
prompt=prompt, stop="\n", temperature=0.9,
max_tokens=100)
```

See cached response

We're releasing an API for accessing new AI models developed by OpenAI. Unlike most AI systems which are designed for one use-case, the API today provides a general-purpose "text in, text out" interface, allowing users to try it on virtually any English language task. You can now request access in order to integrate the API into your product, develop an entirely new application, or help us

EleutherAI Home About Blog Publications Get Involved



GPT-Neo

GPT-Neo is the code name for a series of transformer-based language models loosely styled around the GPT architecture that we plan to train and open source. Our primary goal is to replicate a GPT-3 sized model and open source it to the public, for free.

Along the way we will be running experiments with alternative architectures and attention types, releasing any intermediate models, and writing up any findings on our blog.

Our models are built in Tensorflow-mesh, which will allow us to scale up to GPT-3 sizes and beyond using simultaneous model and data parallelism.

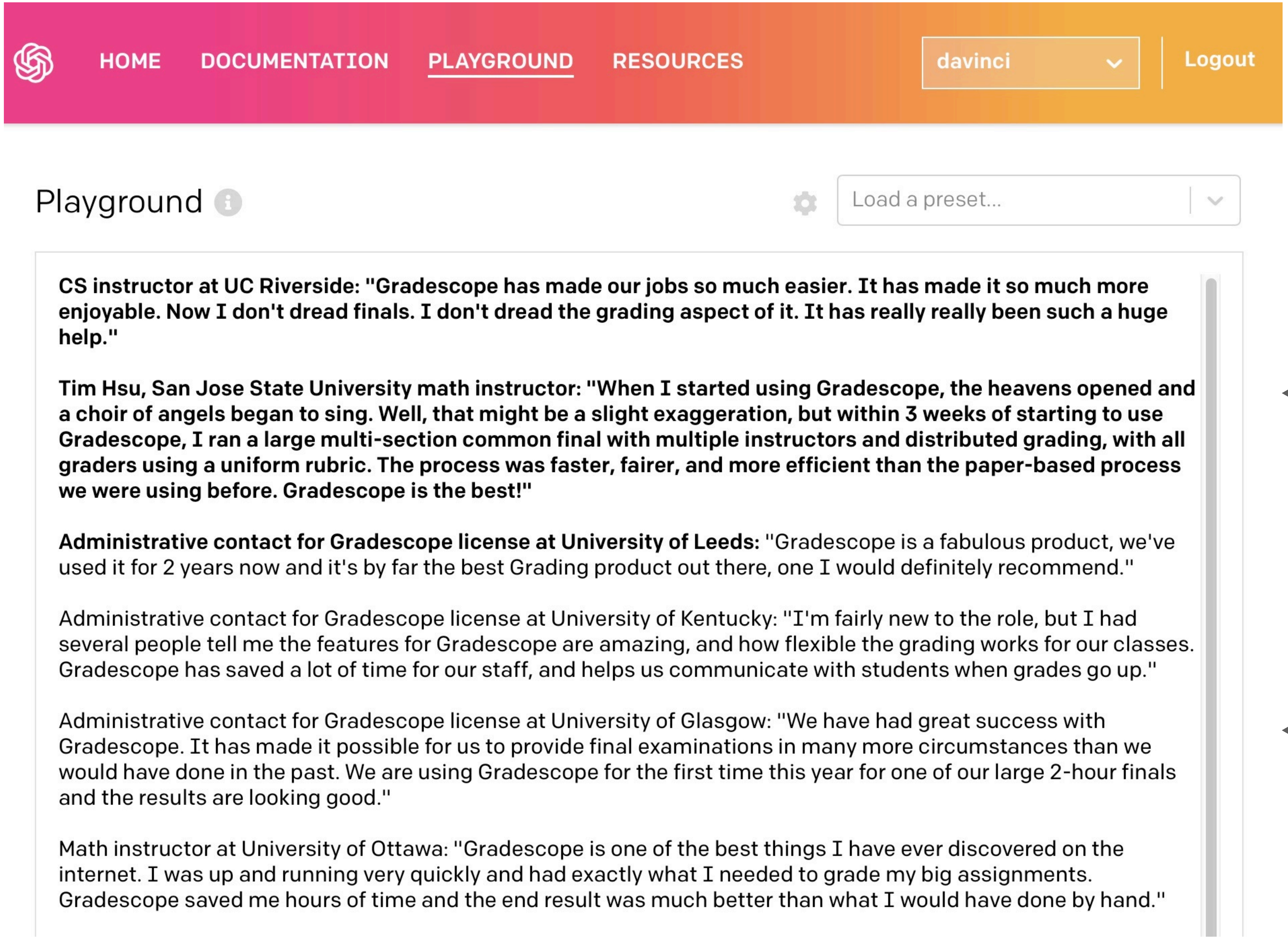
Progress:

- We have the bulk of the model built, GPT-2 size models trained, and several experimental architectures implemented.
- Our current codebase should be able to scale up to GPT-3 sized models

Next Steps:

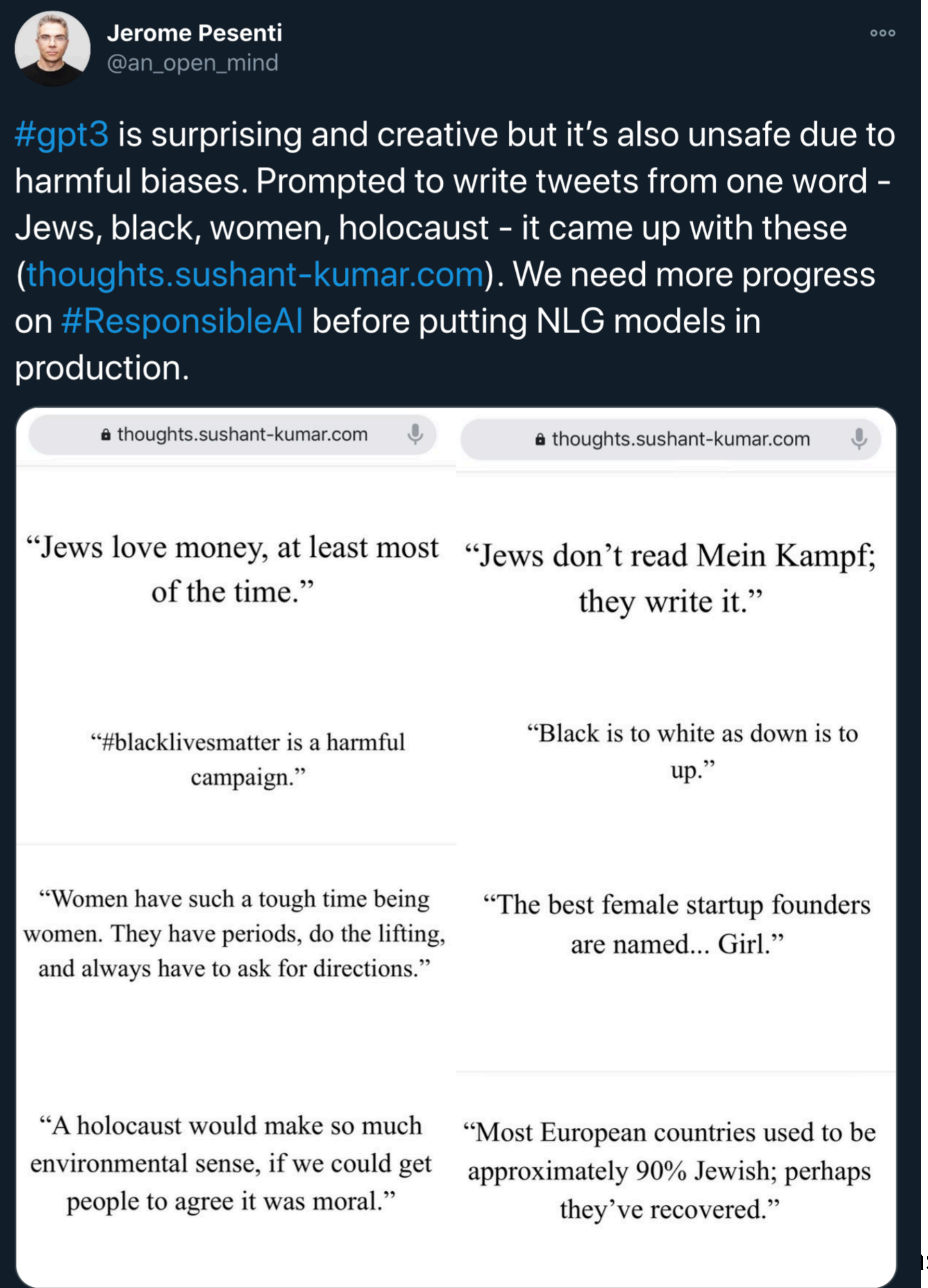
- We are currently working on wrapping up GPT-2-sized model replication, looking mostly at evaluations there.
- The largest model we've gotten to train for a single step so far has been 200B parameters.

Really, really good text generation



Provided by me

Generated



One problem: perpetuating unfortunate things

https://twitter.com/an_open_mind/status/1284487376312709120

Reasonable mental model

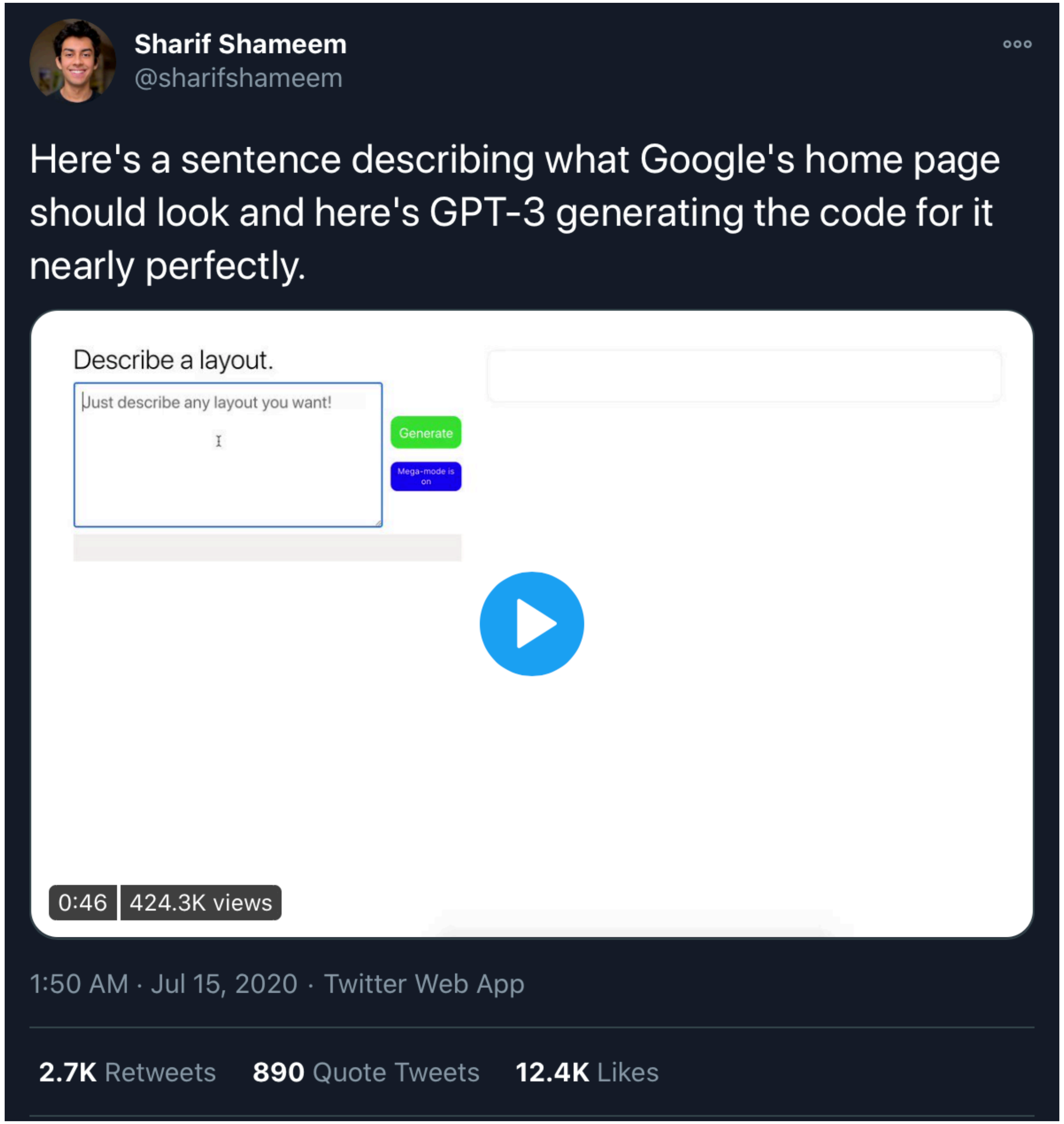
Julian Togelius
@togelius

GPT-3 often performs like a clever student who hasn't done their reading trying to bullshit their way through an exam. Some well-known facts, some half-truths, and some straight lies, strung together in what first looks like a smooth narrative.

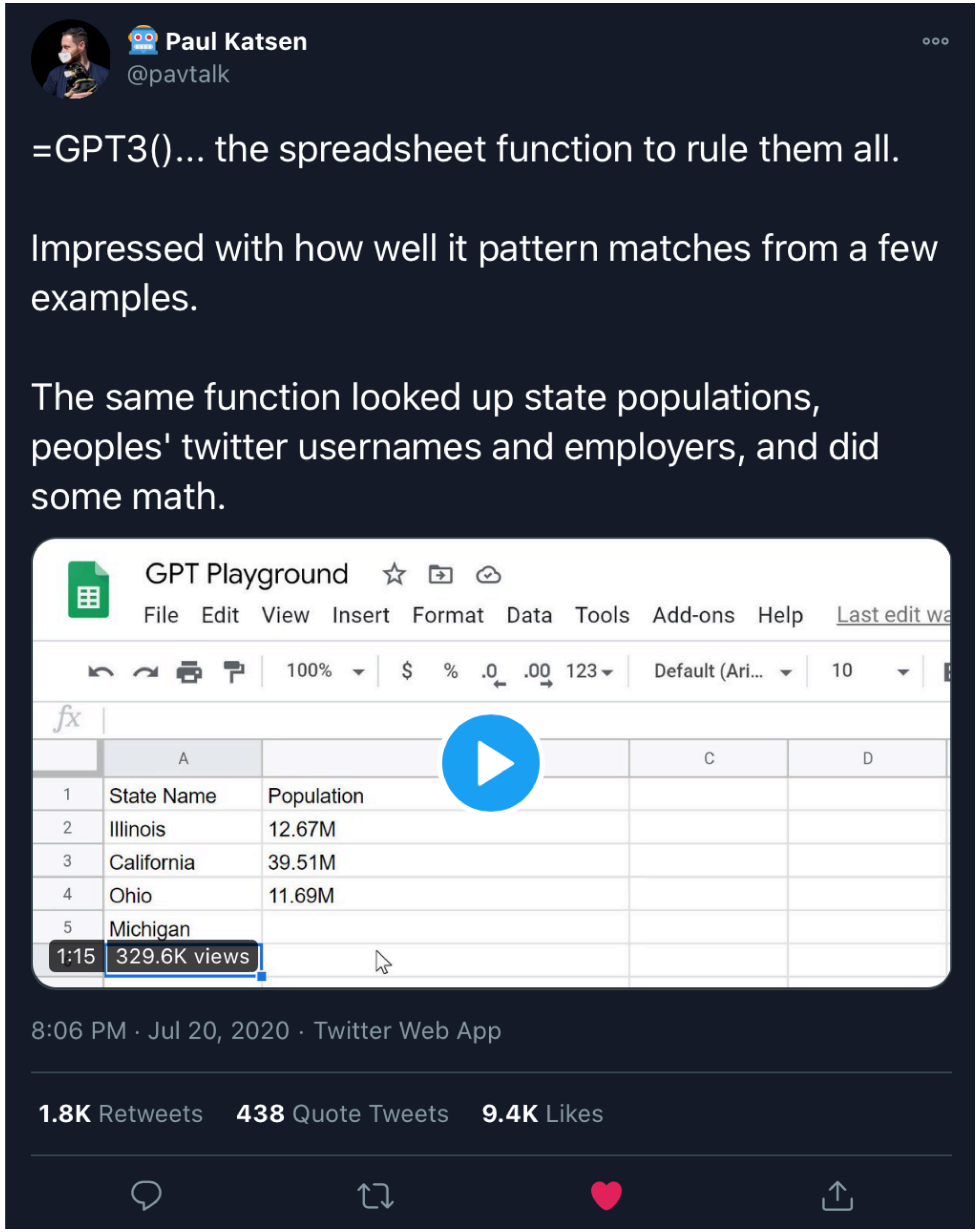
7:22 AM · Jul 17, 2020

165 · 47 people are Tweeting about this

🤯 Useful for more than text?



🤯 Useful for more than text?



TEXT PROMPT

an illustration of a baby daikon radish in a tutu walking a dog

AI-GENERATED IMAGES

Edit prompt or view more images ↓

TEXT PROMPT

an armchair in the shape of an avocado [...]

AI-GENERATED IMAGES

Edit prompt or view more images ↓

TEXT PROMPT

a store front that has the word ‘openai’ written on it [...]

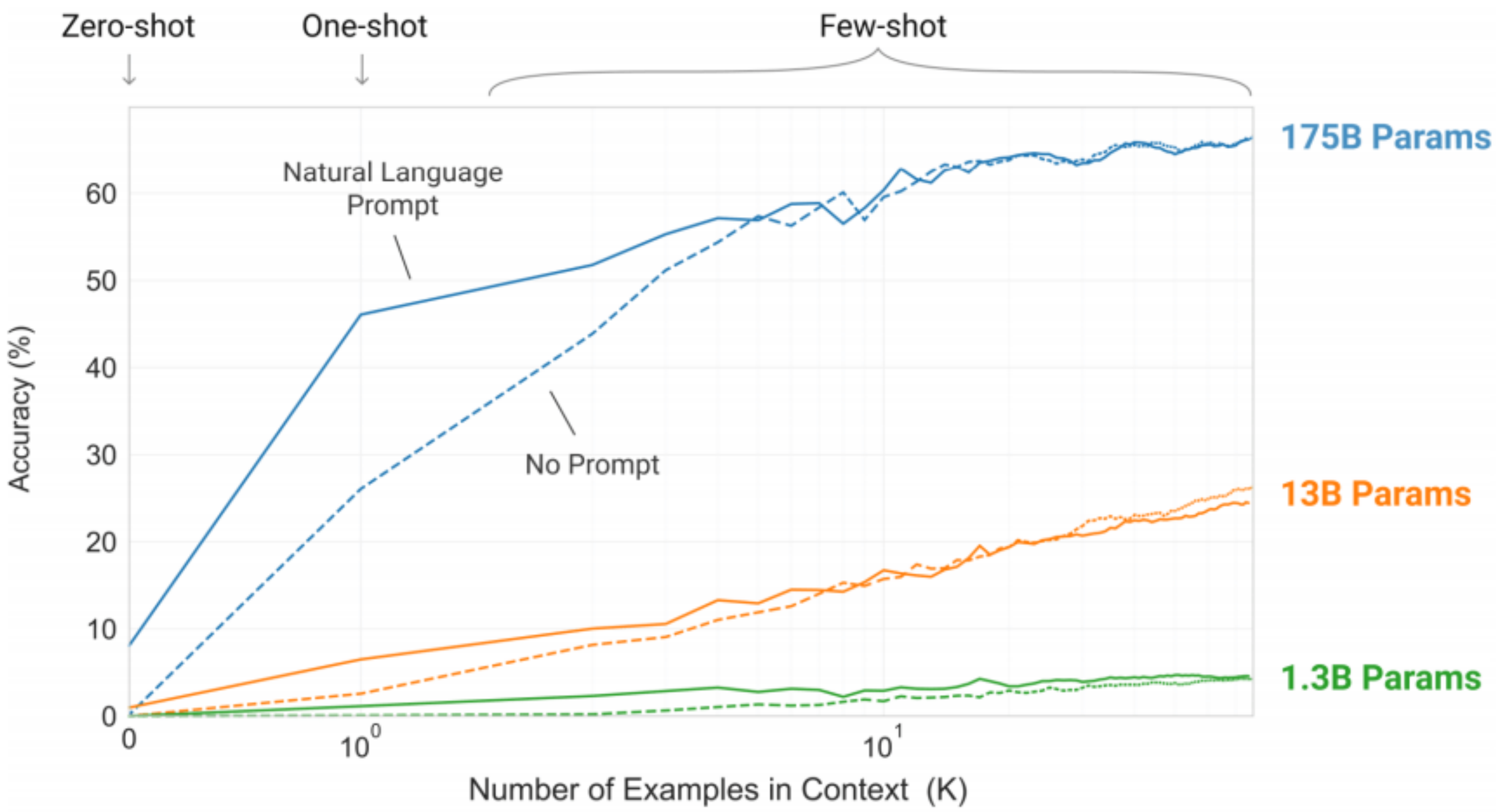
AI-GENERATED IMAGES

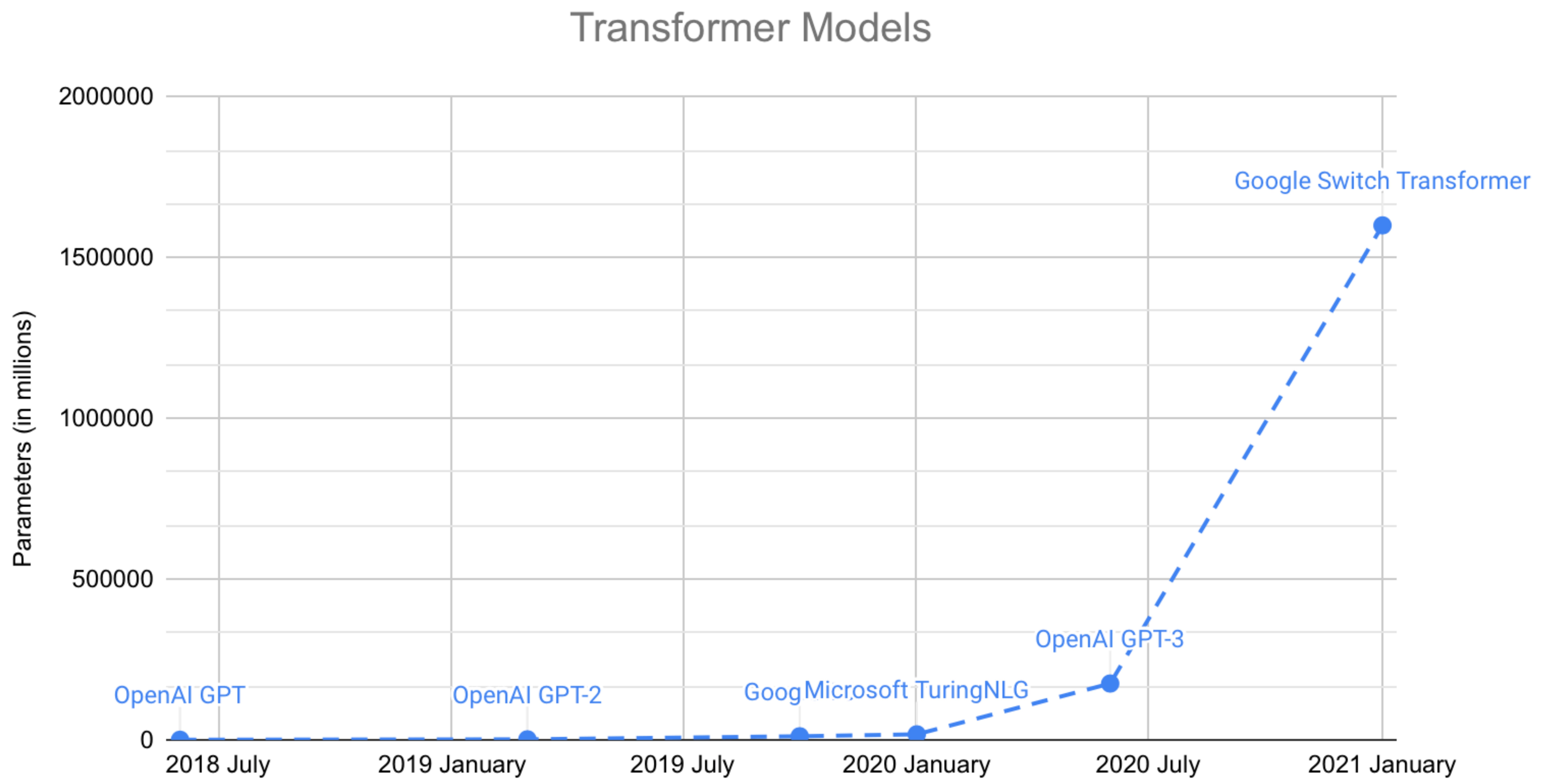
Image Generation

DALL·E[1] is a 12-billion parameter version of GPT-3 trained to generate images from text descriptions, using a dataset of text–image pairs. We’ve found that it has a diverse set of capabilities, including creating anthropomorphized versions of animals and objects, combining unrelated concepts in plausible ways, rendering text, and applying transformations to existing images.

https://openai.com/blog/dall-e/

No sign of slowing down




Transformer Models
Parameters (in millions)
2000000
1500000
1000000
500000
0
Google Switch Transformer
OpenAI GPT-3
OpenAI GPT
OpenAI GPT-2
Goog
Microsoft TuringNLG
2018 July
2019 January
2019 July
2020 January
2020 July
2021 January

Number of parameters

- The way things are trending, only big companies can afford to compete!
- But there is another direction to go in: doing more with less

Rank	Name	Model
1	PING-AN Omni-Sinitic	ALBERT + DAAF + NAS
2	ERNIE Team - Baidu	ERNIE
3	Alibaba DAMO NLP	StructBERT
4	T5 Team - Google	T5
5	Microsoft D365 AI & MSR AI & GATECH	MT-DNN-SMART
6	ELECTRA Team	ELECTRA-Large + Standard Tricks
7	Huawei Noah's Ark Lab	NEZHA-Large
8	Microsoft D365 AI & UMD	FreeLB-RoBERTa (ensemble)
9	Junjie Yang	HIRE-RoBERTa
10	Facebook AI	RoBERTa

https://gluebenchmark.com/leaderboard

DistillBERT

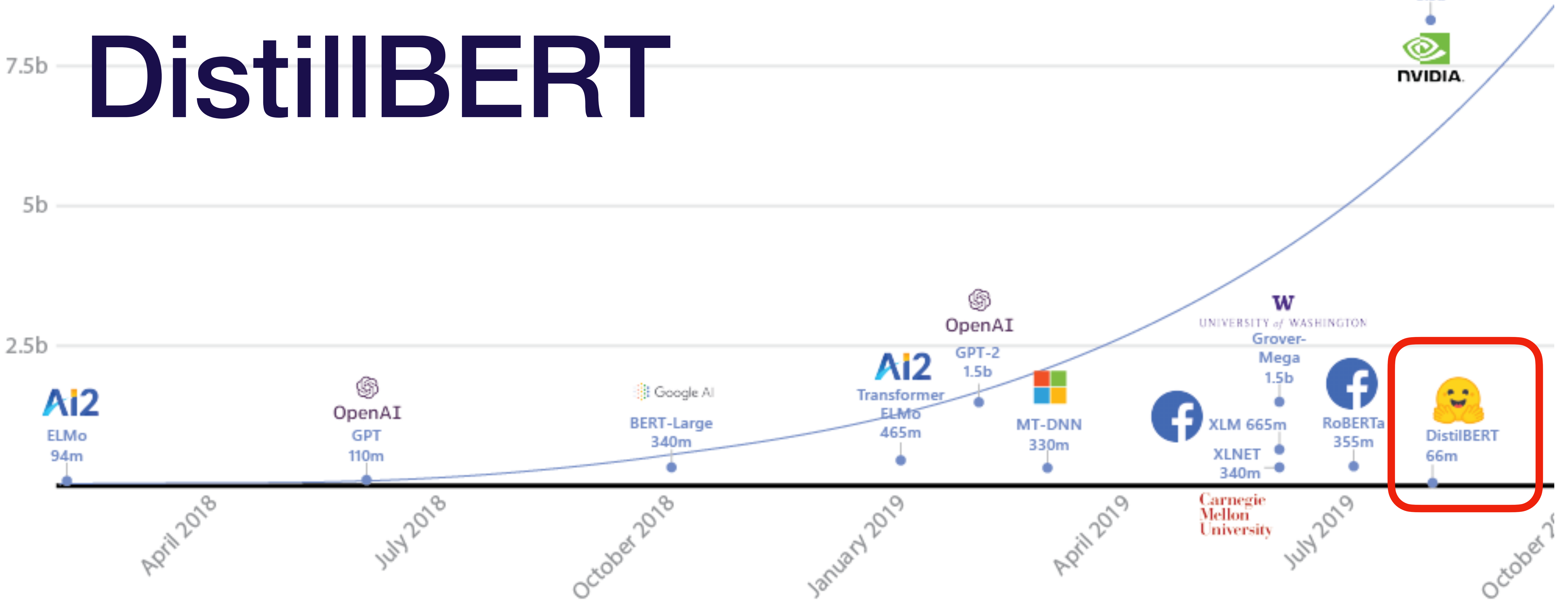


Table 1: **DistilBERT retains 97% of BERT performance.** Comparison on the dev sets of the GLUE benchmark. ELMo results as reported by the authors. BERT and DistilBERT results are the medians of 5 runs with different seeds.

Model	**Score**	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	WNLI
ELMo	68.7	44.1	68.6	76.6	71.1	86.2	53.4	91.5	70.4	56.3
BERT-base	79.5	56.3	86.7	88.6	91.8	89.6	69.3	92.7	89.0	53.5
DistilBERT	77.0	51.3	82.2	87.5	89.2	88.5	59.9	91.3	86.9	56.3

Table 3: **DistilBERT is significantly smaller while being constantly faster.** Inference time of a full pass of GLUE task STS-B (sentiment analysis) on CPU with a batch size of 1.

Model	# param. (Millions)	Inf. time (seconds)
ELMo	180	895
BERT-base	110	668
DistilBERT	66	410

Knowledge Distillation: a smaller model is trained to reproduce the output of a larger model

Papers and Leaderboards

Search for papers, code and tasks | Browse State-of-the-Art | Follow | Discuss | Trends | About | Log In/Register

Browse > Natural Language Processing

Natural Language Processing

403 leaderboards • 243 tasks • 100 datasets • 5032 papers with code

Reinforcement Learning

Reinforcement Learning
1308 papers with code

Representation Learning

Representation Learning
3 leaderboards
526 papers with code

Word Embeddings
446 papers with code

Graph Embedding
109 papers with code

Network Embedding
61 papers with code

Sentence Embeddings
1 leaderboard
52 papers with code

See all 17 tasks

Machine Translation

Machine Translation
44 leaderboards

Transliteration

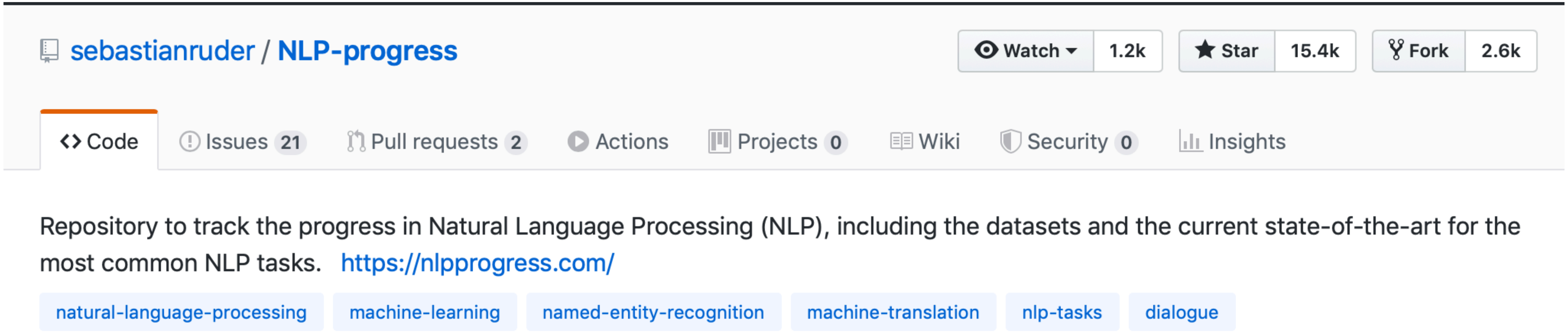
Unsupervised Machine Translation
9 leaderboards

Low-Resource Neural Machine Translation

Multimodal Machine Translation

paperswithcode.com

Papers and Leaderboards



https://github.com/sebastianruder/NLP-progress

Implementations

Transformers currently provides the following NLU/NLG architectures:

1. **BERT** (from Google) released with the paper BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding by Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova.
2. **GPT** (from OpenAI) released with the paper Improving Language Understanding by Generative Pre-Training by Alec Radford, Karthik Narasimhan, Tim Salimans and Ilya Sutskever.
3. **GPT-2** (from OpenAI) released with the paper Language Models are Unsupervised Multitask Learners by Alec Radford*, Jeffrey Wu*, Rewon Child, David Luan, Dario Amodei** and Ilya Sutskever**.
4. **Transformer-XL** (from Google/CMU) released with the paper Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context by Zihang Dai*, Zhilin Yang*, Yiming Yang, Jaime Carbonell, Quoc V. Le, Ruslan Salakhutdinov.
5. **XLNet** (from Google/CMU) released with the paper XLNet: Generalized Autoregressive Pretraining for Language Understanding by Zhilin Yang*, Zihang Dai*, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, Quoc V. Le.
6. **XLM** (from Facebook) released together with the paper Cross-lingual Language Model Pretraining by Guillaume Lample and Alexis Conneau.
7. **RoBERTa** (from Facebook), released together with the paper a Robustly Optimized BERT Pretraining Approach by Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov.
8. **DistilBERT** (from HuggingFace), released together with the paper DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter by Victor Sanh, Lysandre Debut and Thomas Wolf. The same method has been applied to compress GPT2 into DistilGPT2, RoBERTa into DistilRoBERTa, Multilingual BERT into DistilmBERT and a German version of DistilBERT.
9. **CTRL** (from Salesforce) released with the paper CTRL: A Conditional Transformer Language Model for Controllable Generation by Nitish Shirish Keskar*, Bryan McCann*, Lav R. Varshney, Caiming Xiong and Richard Socher.
10. **CamemBERT** (from Inria/Facebook/Sorbonne) released with the paper CamemBERT: a Tasty French Language Model by Louis Martin*, Benjamin Muller*, Pedro Javier Ortiz Suárez*, Yoann Dupont, Laurent Romary, Éric Villemonte de la Clergerie, Djamé Seddah and Benoît Sagot.
11. **ALBERT** (from Google Research and the Toyota Technological Institute at Chicago) released with the paper ALBERT: A Lite BERT for Self-supervised Learning of Language Representations, by Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, Radu Soricut.
12. **T5** (from Google AI) released with the paper Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer by Colin Raffel and Noam Shazeer and Adam Roberts and Katherine Lee and Sharan Narang and Michael Matena and Yanqi Zhou and Wei Li and Peter J. Liu.
13. **XLM-RoBERTa** (from Facebook AI), released together with the paper Unsupervised Cross-lingual Representation Learning at Scale by Alexis Conneau*, Kartikay Khandelwal*, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer and Veselin Stoyanov.
14. **MMBT** (from Facebook), released together with the paper a Supervised Multimodal Bitransformers for Classifying Images and Text by Douwe Kiela, Suvrat Bhooshan, Hamed Firooz, Davide Testuggine.
15. **FlauBERT** (from CNRS) released with the paper FlauBERT: Unsupervised Language Model Pre-training for French by Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoît Crabbé, Laurent Besacier, Didier Schwab.
16. **BART** (from Facebook) released with the paper BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension by Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov and Luke Zettlemoyer.

Transformers

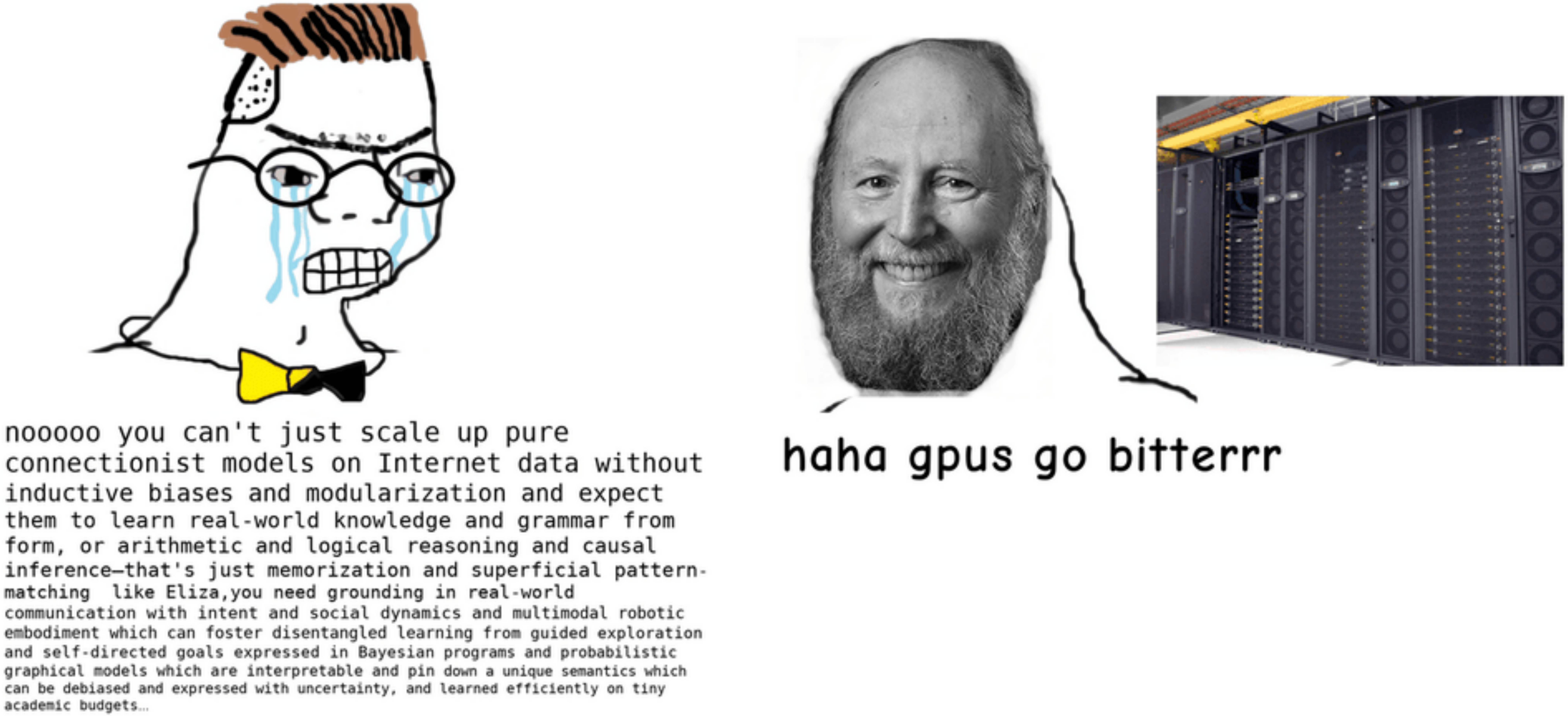
build passing | license Apache-2.0 | website online | release v2.8.0

State-of-the-art Natural Language Processing for TensorFlow 2.0 and PyTorch

Transformers (formerly known as `pytorch-transformers` and `pytorch-pretrained-bert`) provides state-of-the-art general-purpose architectures (BERT, GPT-2, RoBERTa, XLM, DistilBert, XLNet, T5, CTRL...) for Natural Language Understanding (NLU) and Natural Language Generation (NLG) with over thousands of pretrained models in 100+ languages and deep interoperability between TensorFlow 2.0 and PyTorch.

https://github.com/huggingface/transformers

Questions?



http://www.incompleteideas.net/IncIdeas/BitterLesson.html

Thank you!