Transformers from the Ground Up

Sebastian Raschka
https://sebastianraschka.com

PyData
Jeddah

August 5th, 2021

Google AI Blog

The latest from Google Research

Recent Advances in Google Translate

Monday, June 8, 2020

Posted by Isaac Caswell and Bowen Liang, Software Engineers, Google Research

https://ai.googleblog.com/2020/06/recent-advances-in-google-translate.html

PNAS
Proceedings of the National Academy of Sciences of the United States of America

Log in

Keyword, Author, or DOI

Advanced Search

Home | Articles | Front Matter | News | Podcasts | Authors | Submit

RESEARCH ARTICLE

Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences

Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, D…
+ See all authors and affiliations

PNAS April 13, 2021 118 (15) e2016239118; https://doi.org/10.1073/pnas.2016239118

Edited by David T. Jones, University College London, London, United Kingdom, and accepted by Editorial Board

Article Alerts
Email Article
Citation Tools
© Request Permissions
Share
Tweet
Like 428
Mendeley

ARTICLE CLASSIFICATIONS

Biological Sciences » Biophysics and Computational Biology

https://www.pnas.org/content/118/15/e2016239118.short

https://github.blog/2021-06-29-introducing-github-copilot-ai-pair-programmer/

Topics

1. Augmenting RNNs with attention
2. Self-attention
3. The original transformer architecture
4. Large-scale language models
5. Fine-tuning a pre-trained BERT model in PyTorch
6. Quo vadis, transformers?

Think of this talk as a conceptual overview

That may help to navigate the transformer jungle if you are interested

Please don't worry so much about the mathematical or conceptual details in this talk. These details would take many, many hours to talk about and digest.

Topics

1. Augmenting RNNs with attention

2. Self-attention

3. The original transformer architecture

4. Large-scale language models

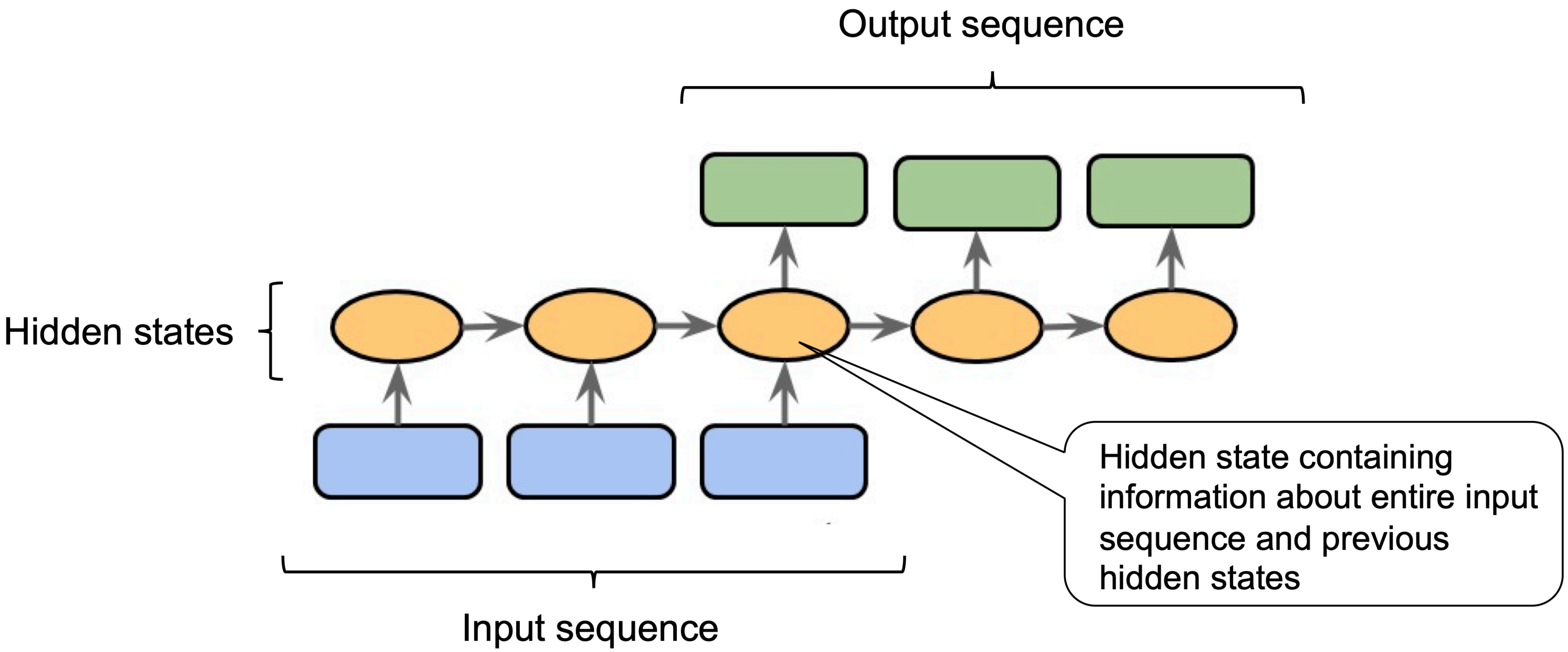
5. Fine-tuning a pre-trained BERT model in PyTorch

6. Quo vadis, transformers?

Regular encoder-decoder RNN for seq2seq tasks

RNN = Recurrent neural network

Seq2seq = sequence-to-sequence (e.g., translation, summarization, ...)



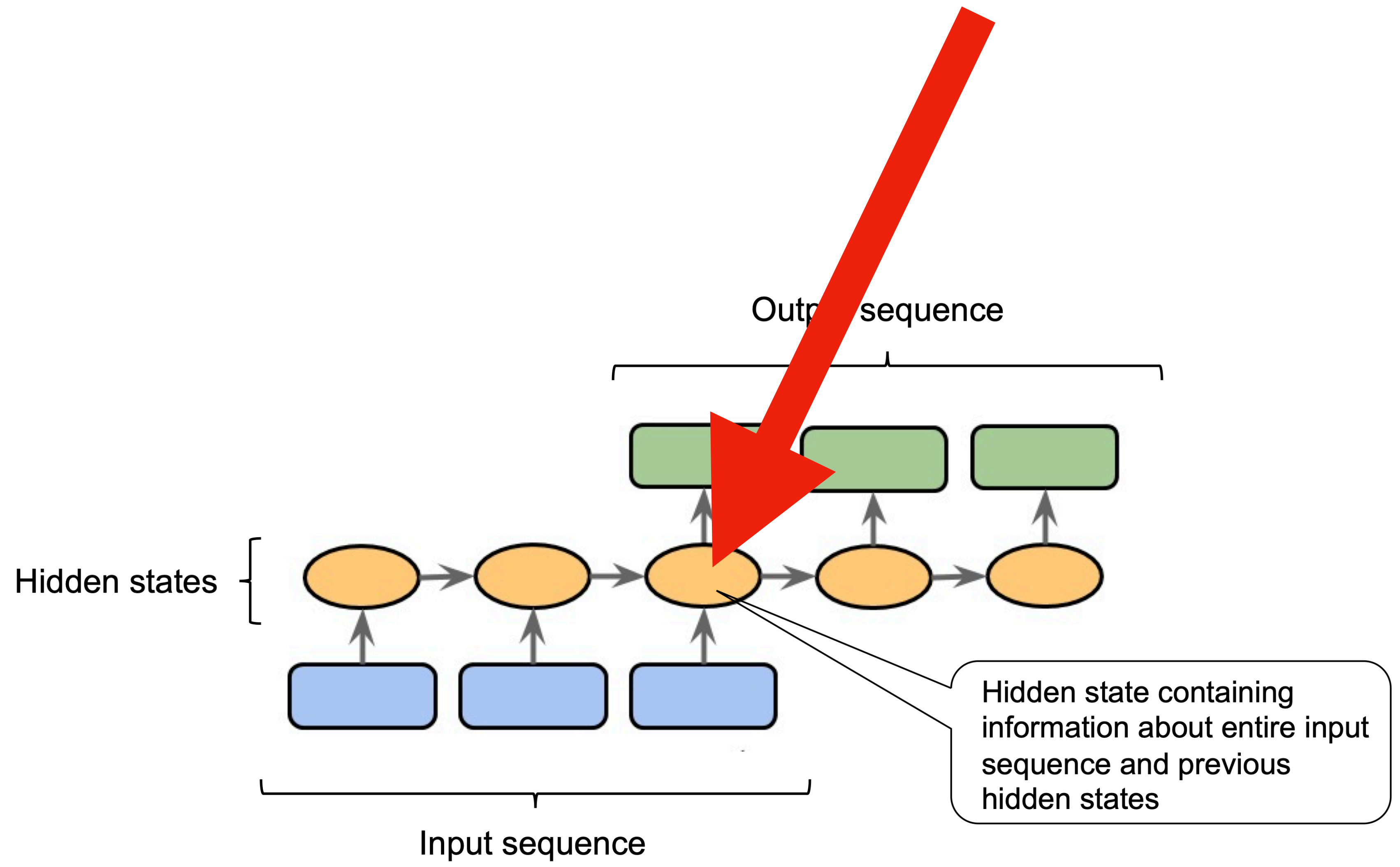
Why parsing all the input before attempting translation?

Because we can't just translate word by word

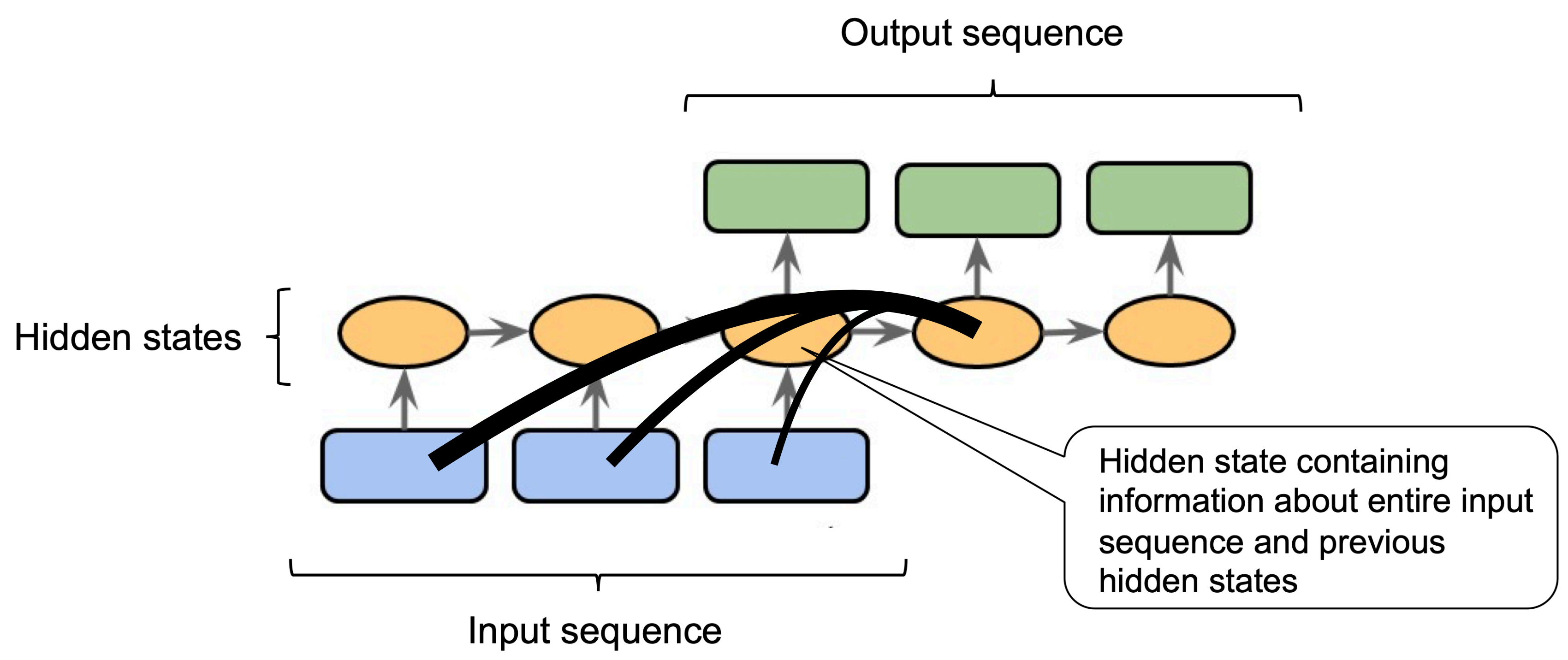
German input sentence:	`Kannst du mir helfen diesen Satz zu uebersetzen ?`
Word-by-word translation (incorrect):	`Can you me help this sentence to translate ?`

German input sentence:	`Kannst du mir helfen diesen Satz zu uebersetzen ?`
Correct English translation:	`Can you help me to translate this sentence ?`

Problem: RNN has to remember a lot in 1 hidden state



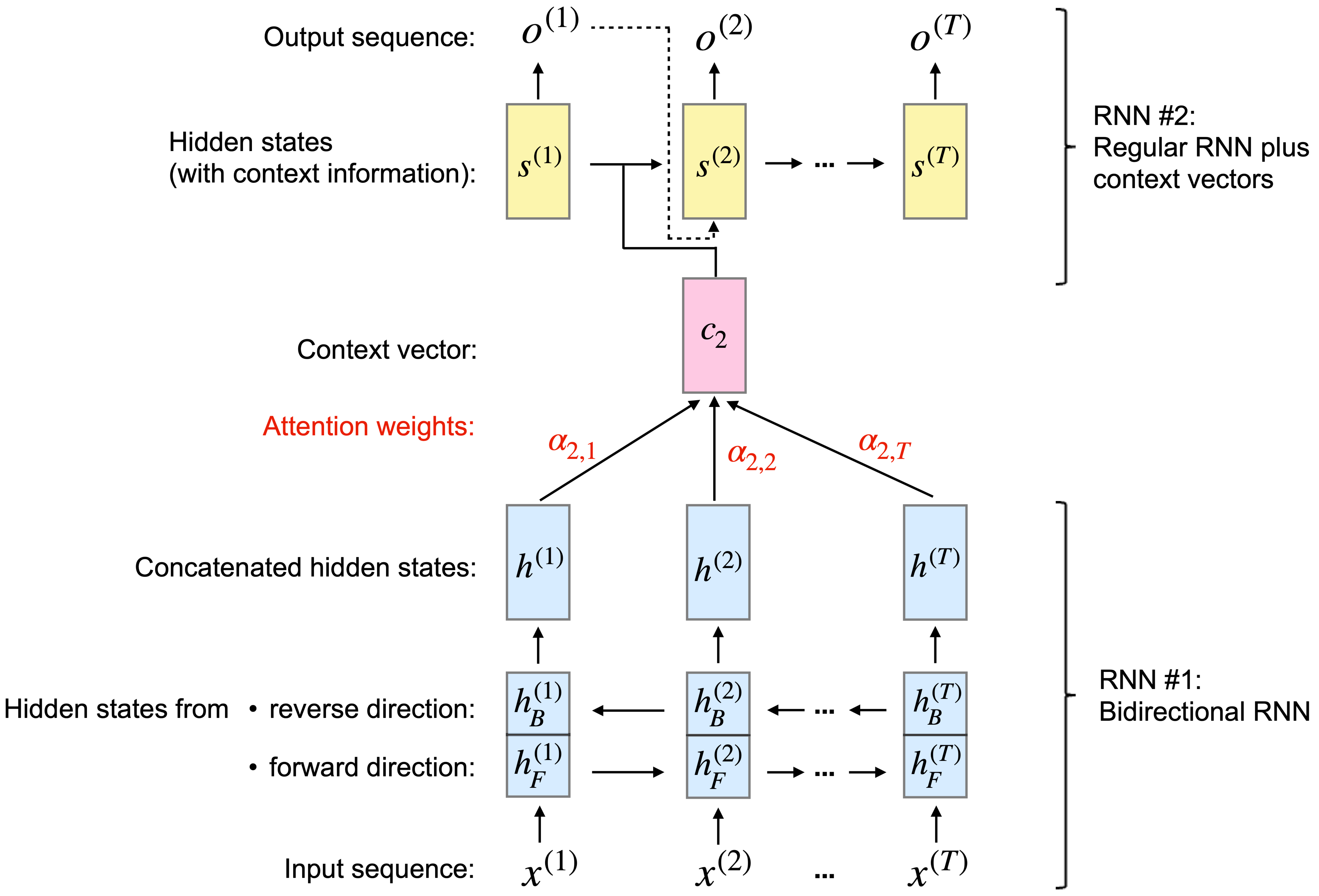
How can we deal with long-range dependencies better and improve language translation?



Idea: attention mechanism that lets each decoder step access relevant inputs

Bahdanau, D., Cho, K., & Bengio, Y. (2014).
Neural Machine Translation by Jointly Learning to Align and Translate
https://arxiv.org/abs/1409.0473

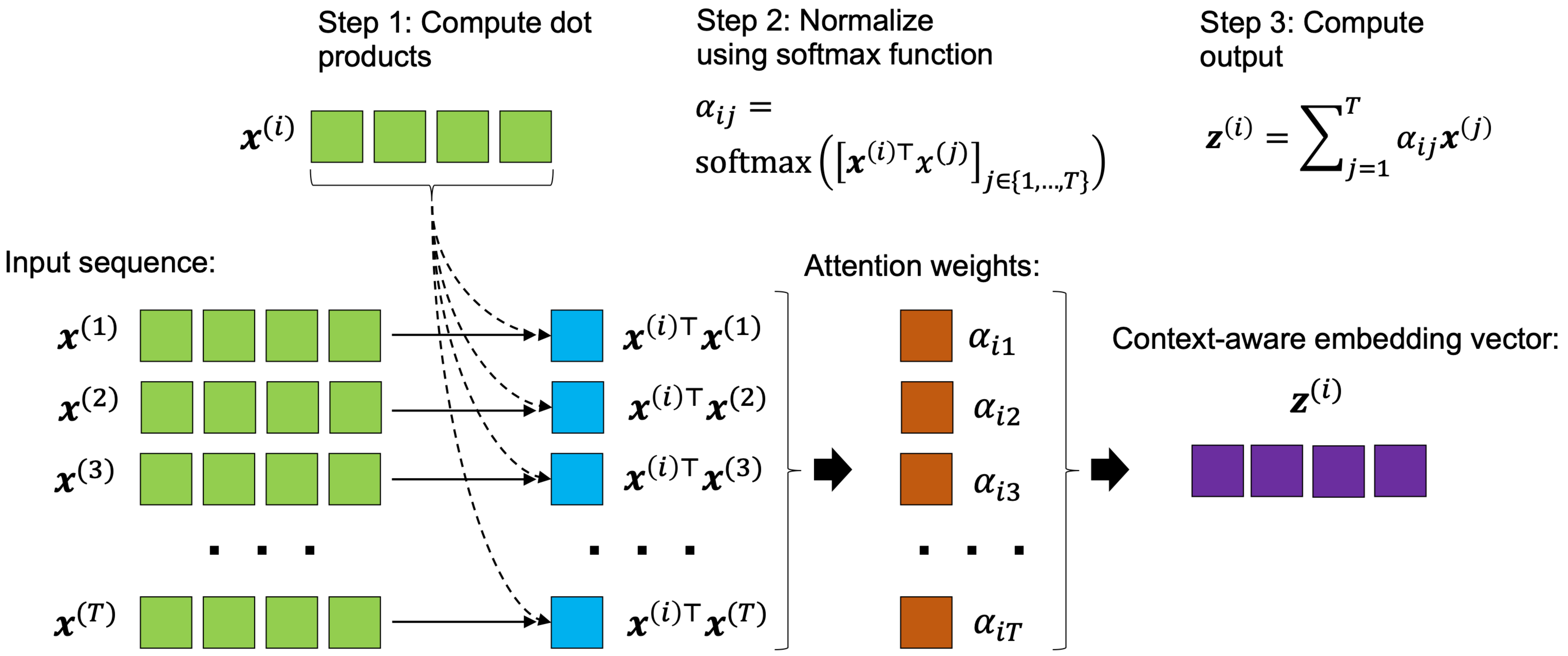
An RNN with attention mechanism



Topics

1. Augmenting RNNs with attention
2. **Self-attention**
3. The original transformer architecture
4. Large-scale language models
5. Fine-tuning a pre-trained BERT model in PyTorch
6. Quo vadis, transformers?

A simple form of self-attention*



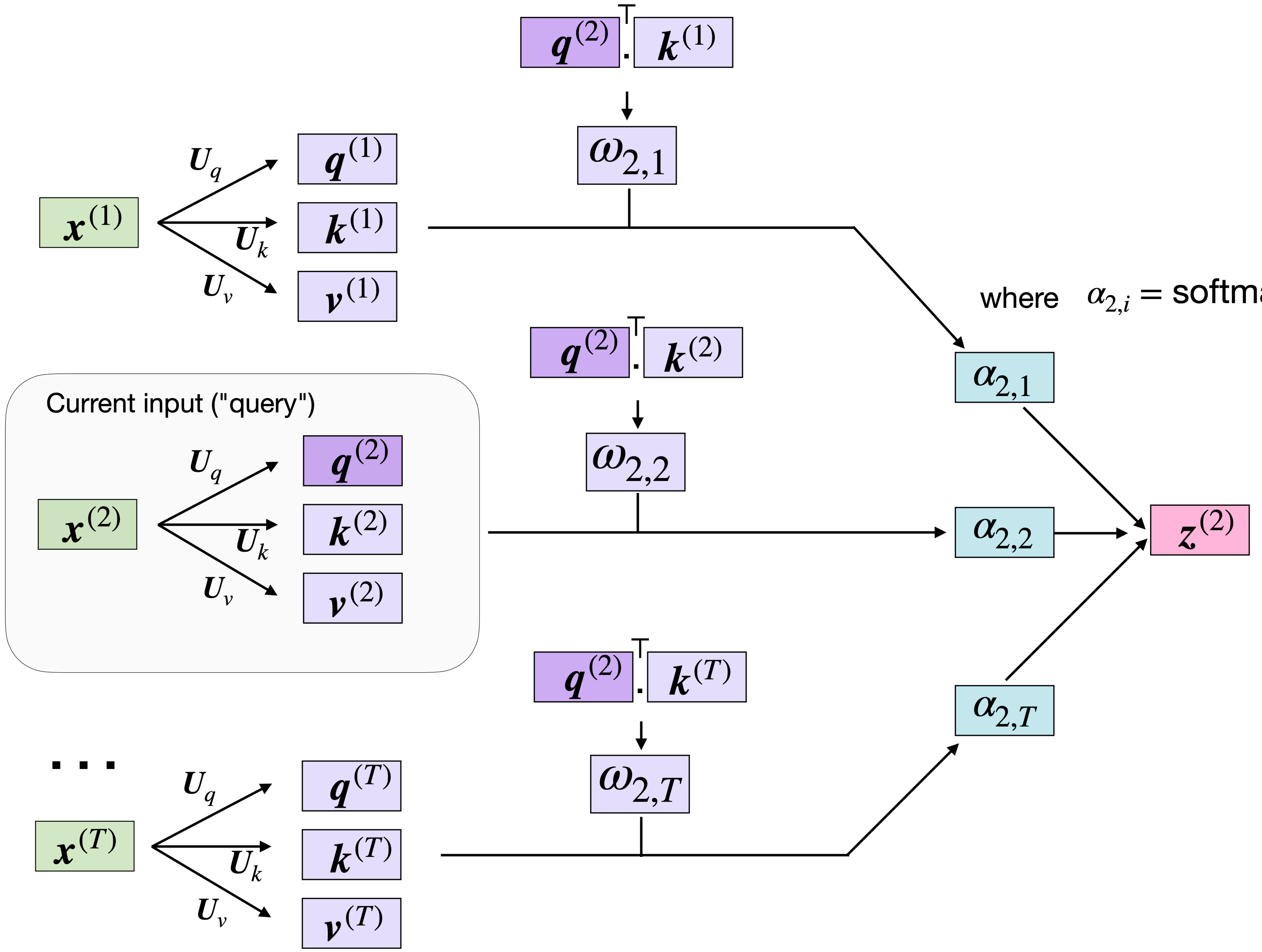
*Minor detail: "self"-attention, because in the attention-based RNN, attention weights are derived from the connection between input & output elements, while self-attention mechanism only focuses on the inputs

Self-attention with learnable weights

"scaled dot product attention"

Where we have three weight matrices $\boldsymbol{U}_q, \boldsymbol{U}_k, \boldsymbol{U}_v$

- Query: $\boldsymbol{q}^{(i)} = \boldsymbol{U}_q \boldsymbol{x}^{(i)}$
- Key: $\boldsymbol{k}^{(i)} = \boldsymbol{U}_k \boldsymbol{x}^{(i)}$
- Value: $\boldsymbol{v}^{(i)} = \boldsymbol{U}_v \boldsymbol{x}^{(i)}$



where $\alpha_{2,i} = \text{softmax}\left(\frac{\omega_{2,i}}{\sqrt{m}}\right) = \frac{\exp(\omega_{2,i}^\top)}{\sum_j \exp(\omega_{2,j}^\top)}$

where $\boldsymbol{z}^{(2)} = \sum_{j=1}^{T} \alpha_{2,j} \boldsymbol{v}^{(j)}$

Topics

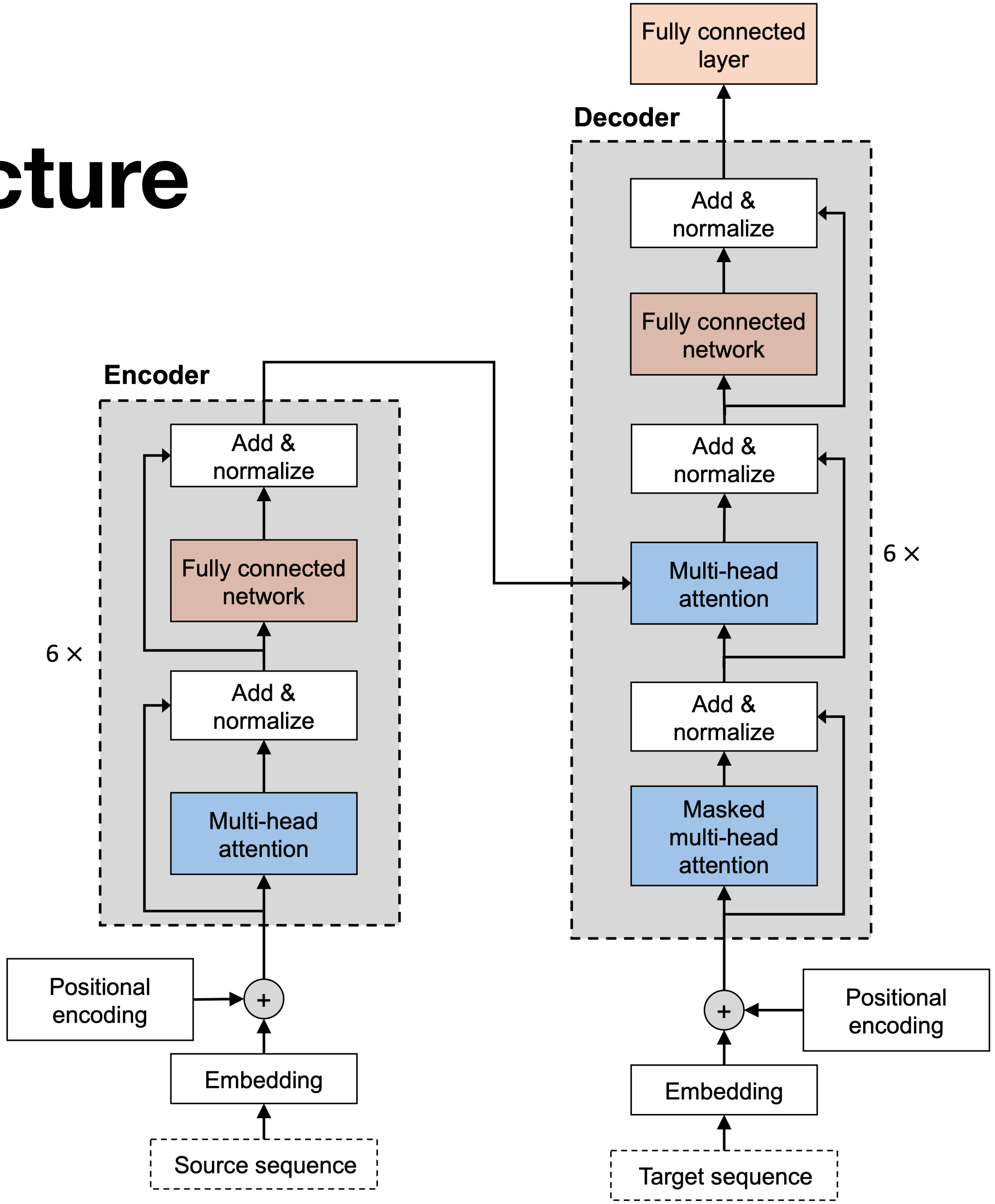
1. Augmenting RNNs with attention
2. Self-attention
3. **The original transformer architecture**
4. Large-scale language models
5. Fine-tuning a pre-trained BERT model in PyTorch
6. Quo vadis, transformers?

Attention Is All You Need

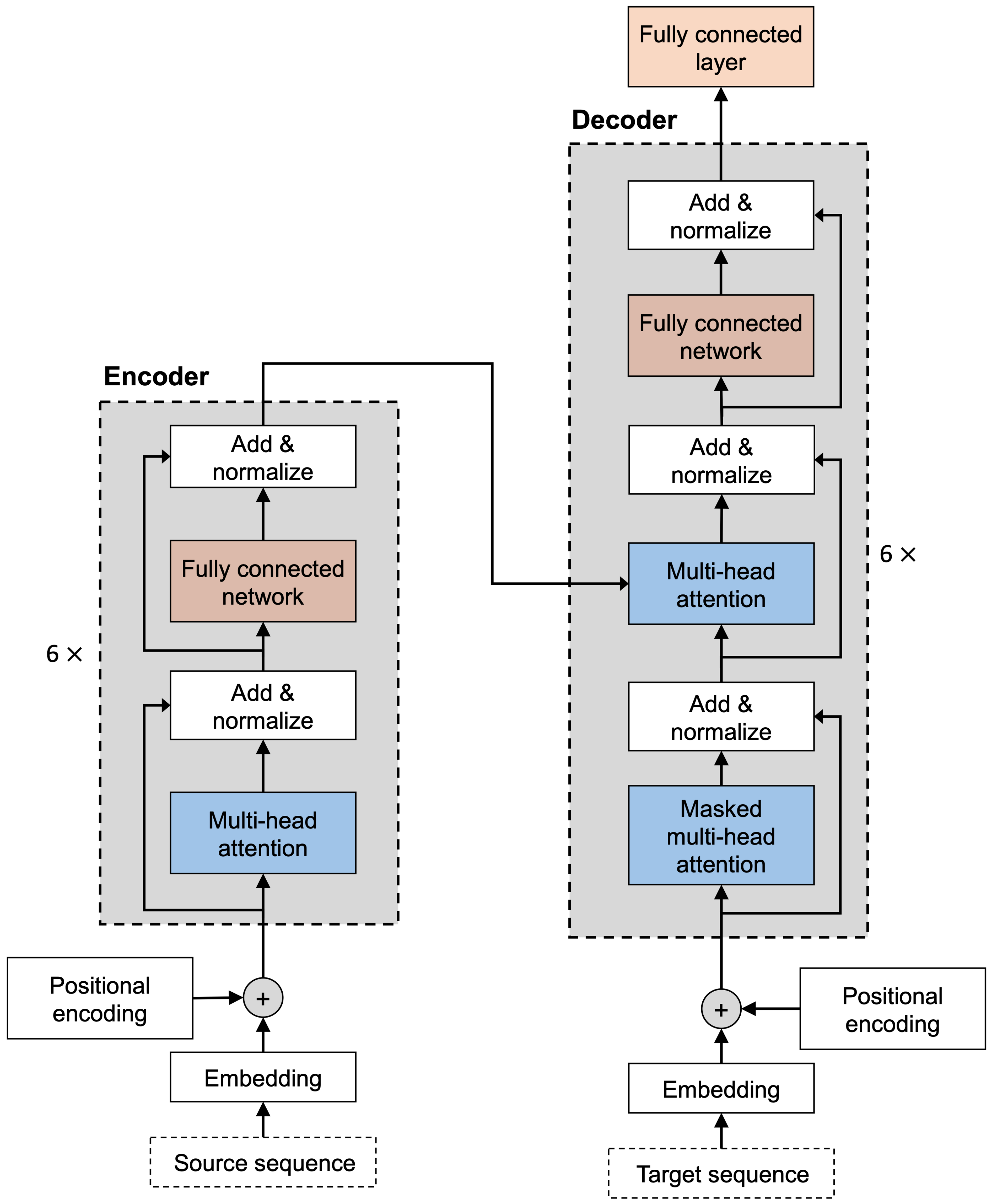
by A. Vaswani and colleagues (2017), https://arxiv.org/abs/1706.03762

- A deep learning architecture for language translation centered around self-attention
- Without any RNN parts

The original transformer architecture



What is multi-head attention?



Multiple matrices $\boldsymbol{U}$ to stack the following multiple time (like kernels/ channels in a CNN):

- Query: $\boldsymbol{q}^{(i)} = \boldsymbol{U}_q \boldsymbol{x}^{(i)}$
- Key: $\boldsymbol{k}^{(i)} = \boldsymbol{U}_k \boldsymbol{x}^{(i)}$
- Value: $\boldsymbol{v}^{(i)} = \boldsymbol{U}_v \boldsymbol{x}^{(i)}$

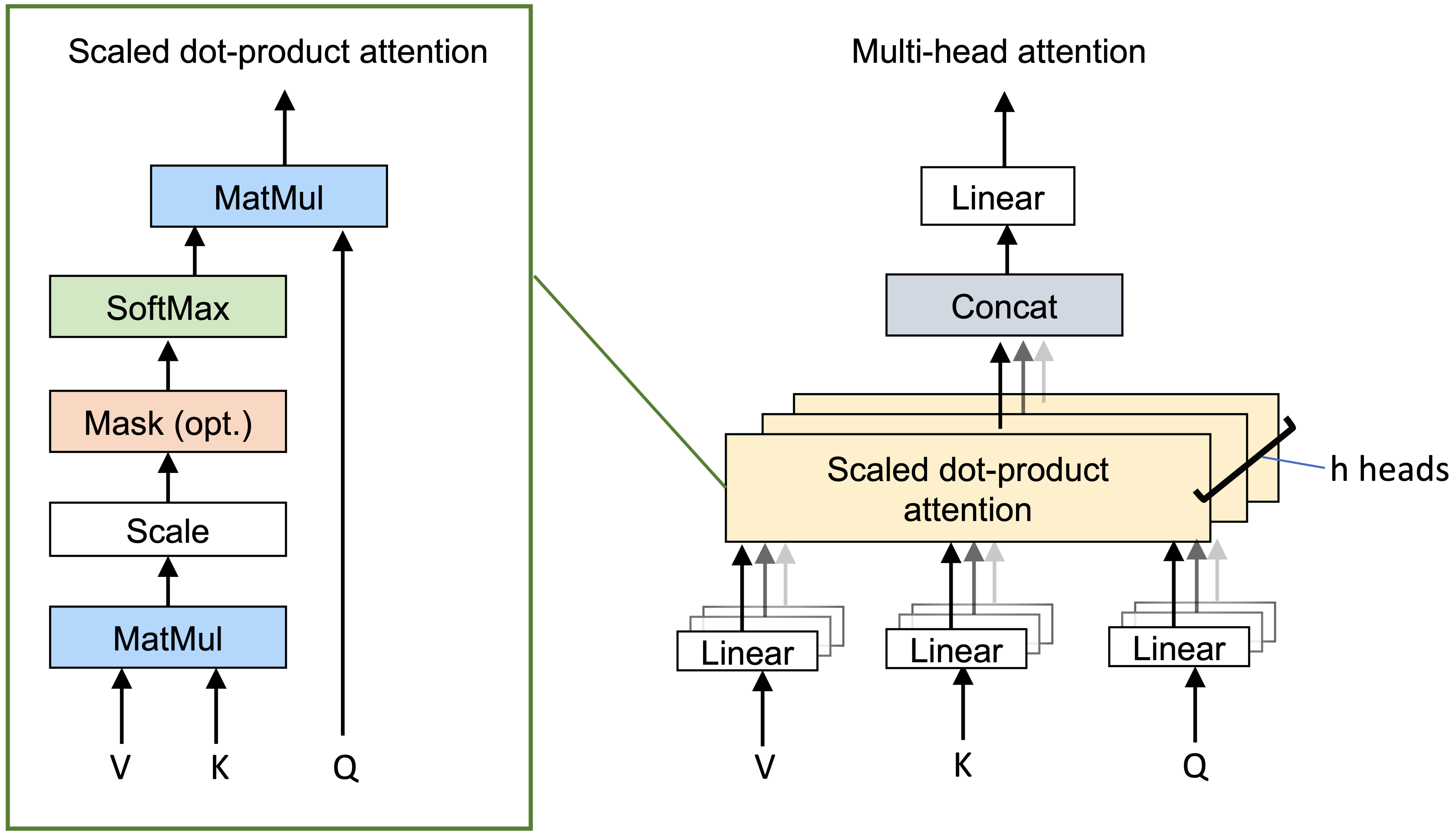
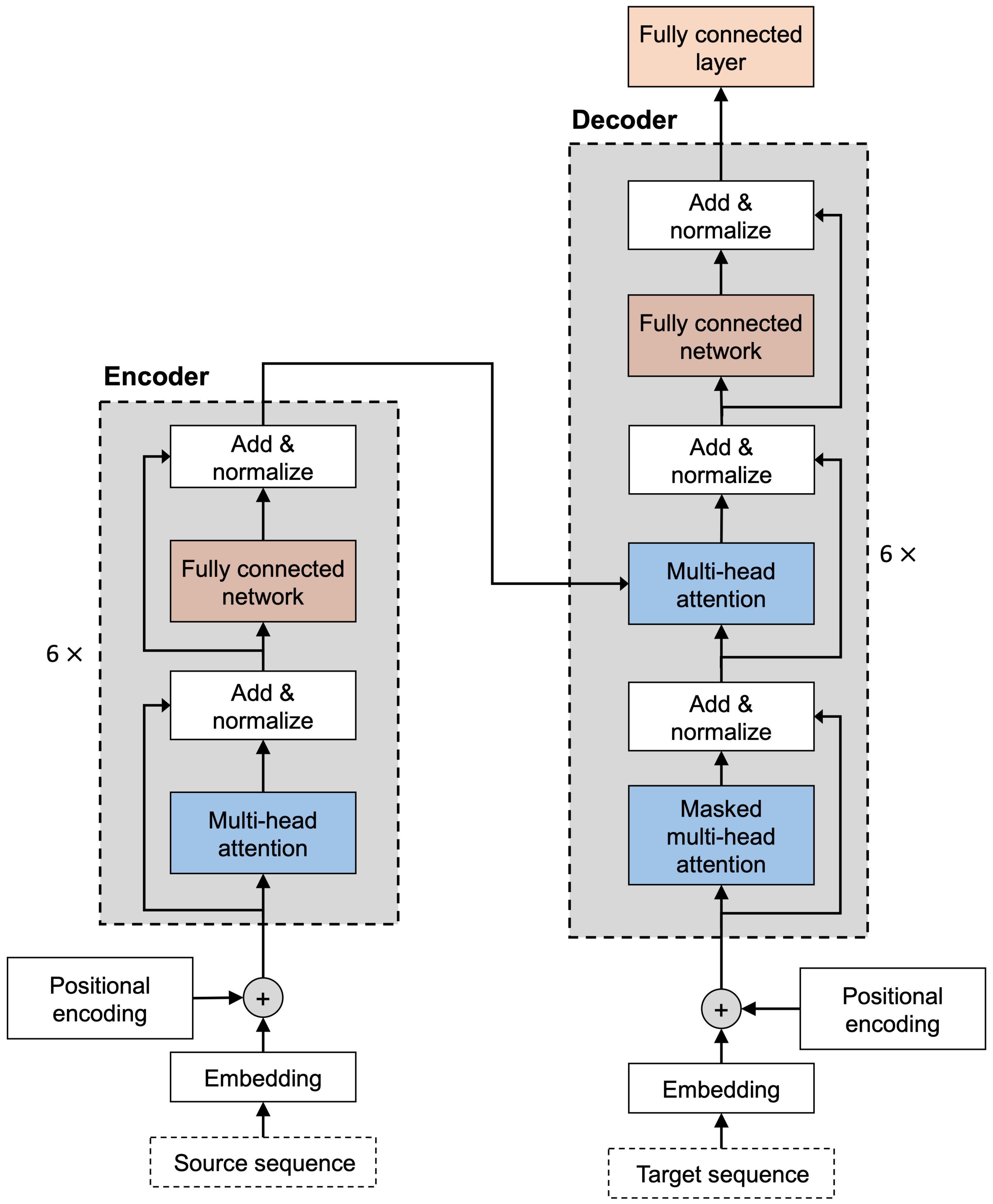


Image credit: Jitian Zhao

What is "masked" multi-head attention?



It masks out words after the current position (those words that the decoder still has to generate)

Can be considered as a form of unidirectional (as opposed to bidirectional) parsing

Topics

1. Augmenting RNNs with attention
2. Self-attention
3. The original transformer architecture
4. **Large-scale language models**
5. Fine-tuning a pre-trained BERT model in PyTorch
6. Quo vadis, transformers?

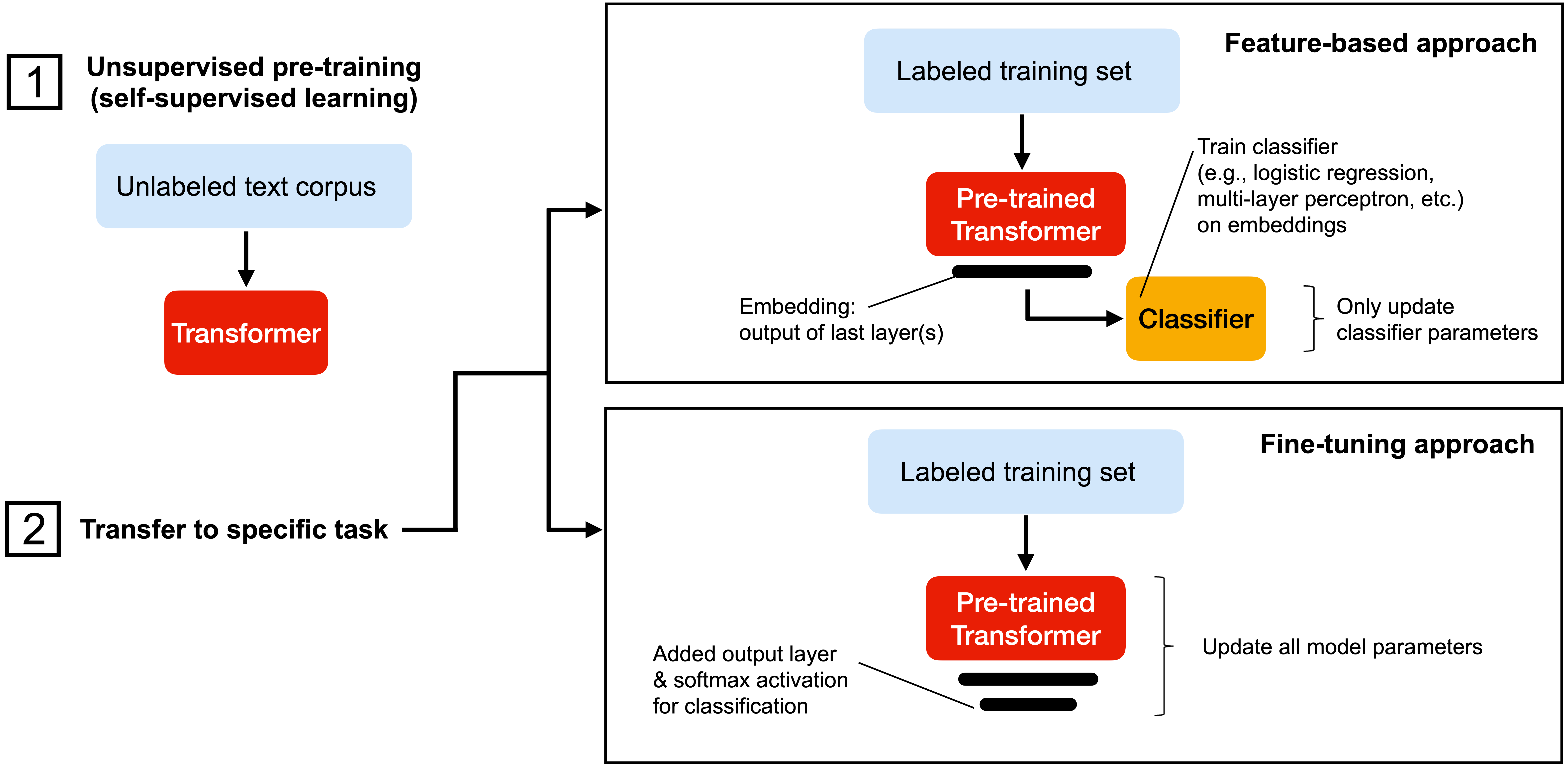
Focus on 2 popular models/approaches:

- **GPT (unidirectional) -> good for generating text**
- **BERT (bidirectional) -> good for prediction (classification)**

Common theme among most transformers:

1. Pre-training on very large, unlabeled datasets
2. Then fine-tuning on labeled dataset for respective target tasks

Training transformers: a 2-step approach

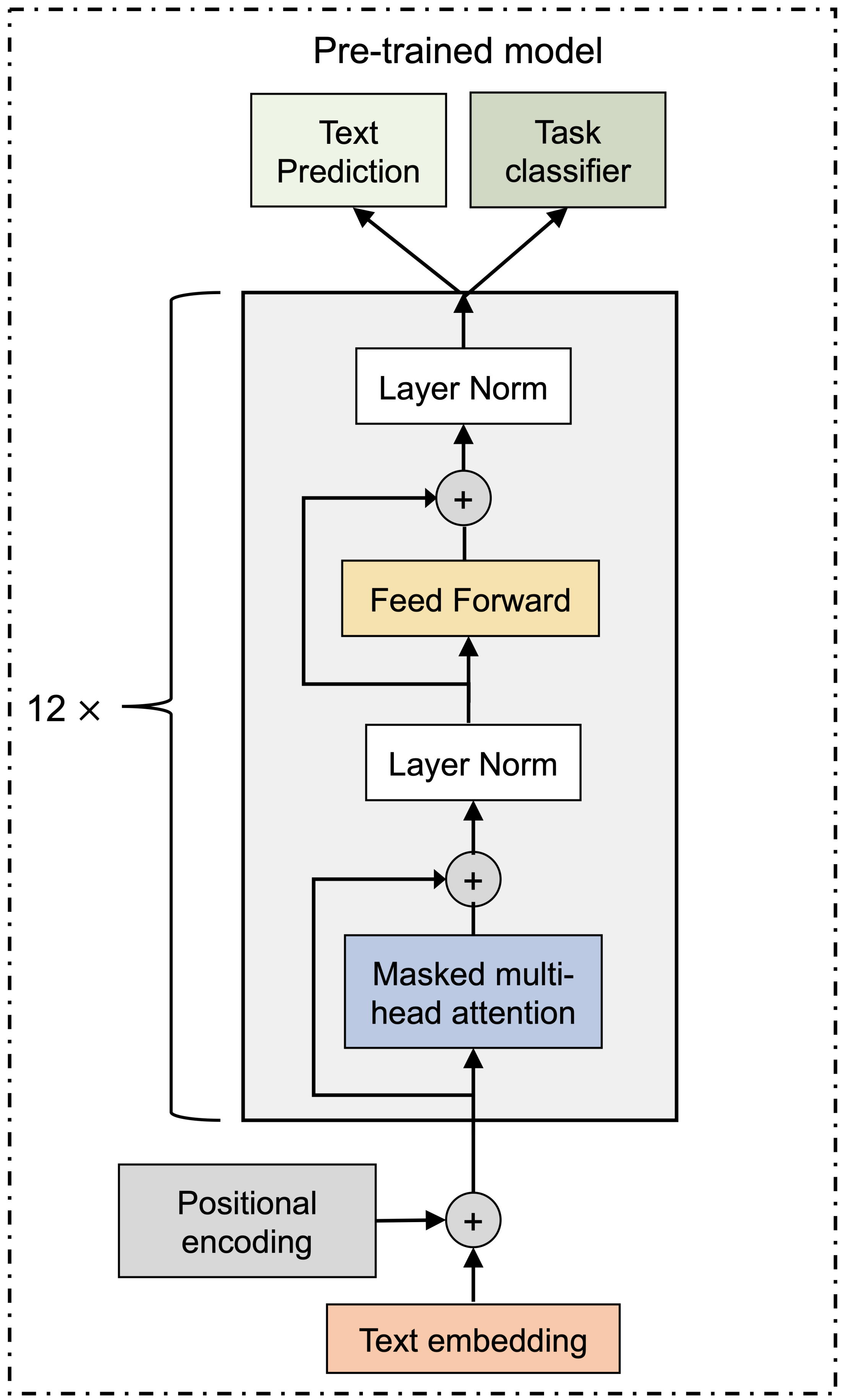


GPT: Generative Pretrained Transformer

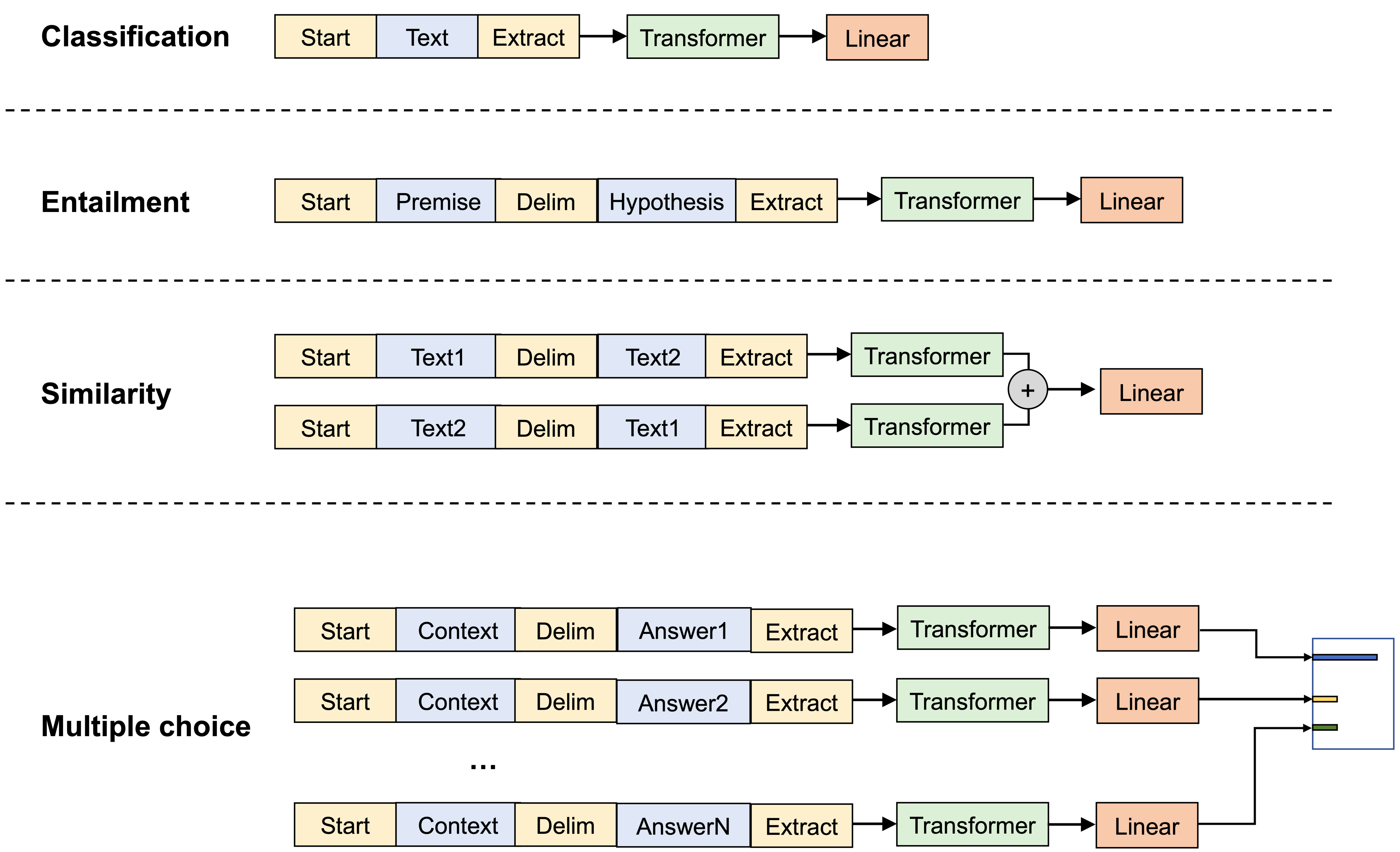
Model	Release year	Number of parameters	Title	Manuscript link
GPT-1	2018	110 million	Improving Language Understanding by Generative Pre-Training	https://www.cs.ubc.ca/~amuham01/LING530/papers/radford2018improving.pdf
GPT-2	2019	1.5 billion	Language Models are Unsupervised Multitask Learners	https://www.semanticscholar.org/paper/Language-Models-are-Unsupervised-Multitask-Learners-Radford-Wu/9405cc0d6169988371b2755e573cc28650d14dfe
GPT-3	2020	175 billion	Language Models are Few-Shot Learners	https://arxiv.org/pdf/2005.14165.pdf

GPT-1

Decoder



Additional layer(s) for fine-tuning



Unidirectional language model: next word prediction

GPT-2: Zero-shot learning

Examples are provided via context (no fine-tuning), e.g.,

“translate to french, english text, french text”

GPT-3: Zero- and few-shot learning

The three settings we explore for in-context learning

1) Zero-shot

The model is only given a natural language description of the task. No weight updates are performed.

```
Translate English to French:    → Task description
cheese =>                       → Prompt
```

2) One-shot

In addition to the task description, the model is also given a simple example of a task. No weight updates are performed.

```
Translate English to French:    → Task description
sea otter => loutre de mer      → Example
cheese =>                       → Prompt
```

3) Few-shot

In addition to the task description, the model is also given a simple example of a task. No gradient updates are performed.

```
Translate English to French:         → Task description
sea otter => loutre de mer
peppermint => menthe poivrée         → Examples
plush giraffe => giraffe peluche
cheese =>                            → Prompt
```

Traditional fine-tuning (not for GPT-3)

Fine-tuning

The model is trained using a large corpus of example tasks.

```
sea otter => loutre de mer          → Example #1
        ↓
Gradient update
        ↓
peppermint => menthe poivrée        → Example #2
        ↓
Gradient update
        ↓
       …
        ↓
plush giraffe => giraffe peluche    → Example #N
        ↓
Gradient update
        ↓
cheese =>                           → Prompt
```

Bidirectional Encoder Representations from Transformers: BERT

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (2018), J. Devlin, M. Chang, K. Lee and K. Toutanova,https://arxiv.org/abs/1810.04805

featuring a bidirectional ("nondirectional") training as it reads all elements at once (as opposed to word by word, left to right)

BERT's Pre-training task 1/2: Masked Language Model

15% of the words are masked (or "marked") and are treated as follows ...

Input sentence: A quick brown fox jumps over a lazy dog.

Output sentence:

- 80% Mask token: replace fox with [MASK]
- 10% Random token: replace fox with coffee
- 10% Unchanged: keep fox

BERT's Pre-training task 2/2: Next sentence prediction

Sentence separator token

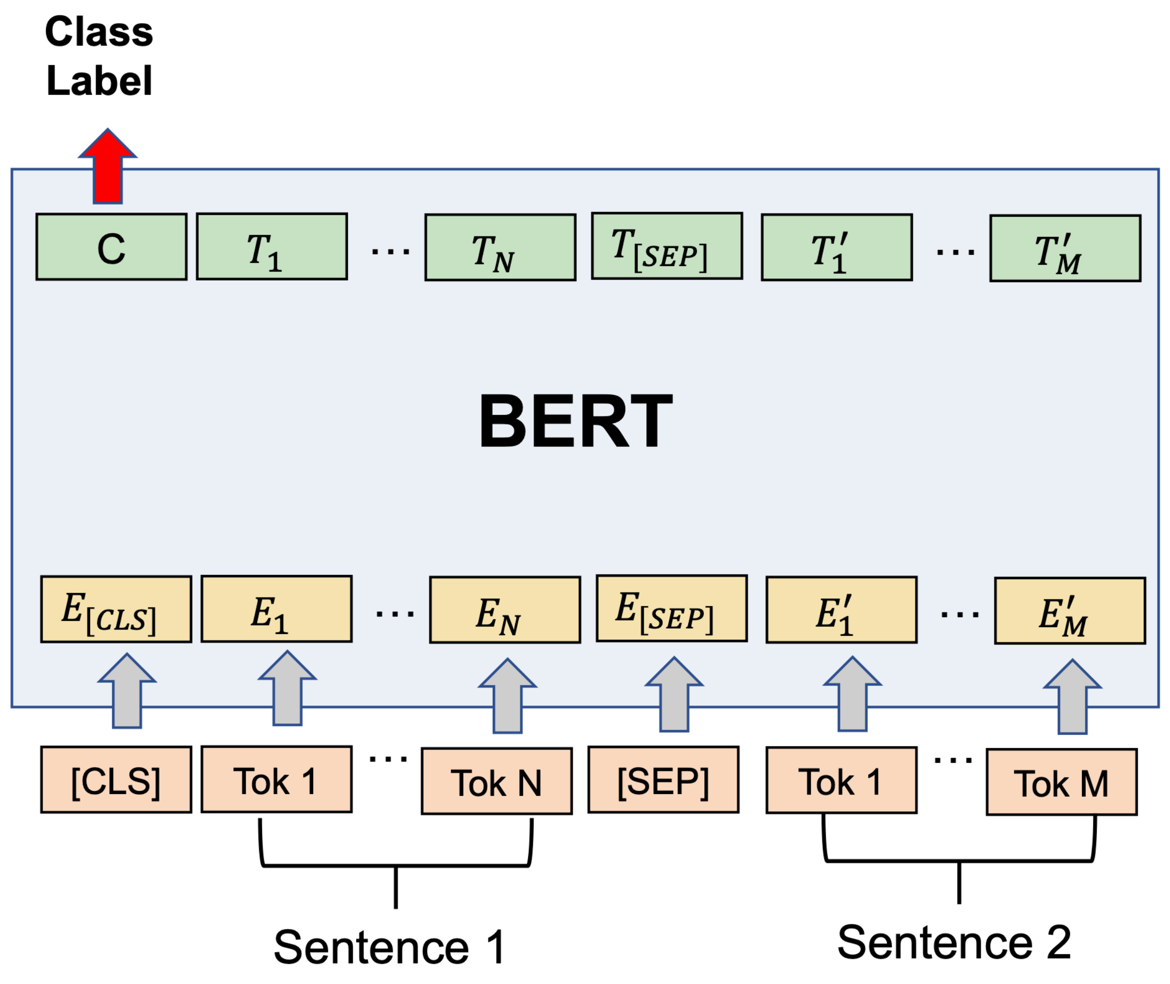
[CLS] A [SEP] B

Classification token:
IsNext, NotNext

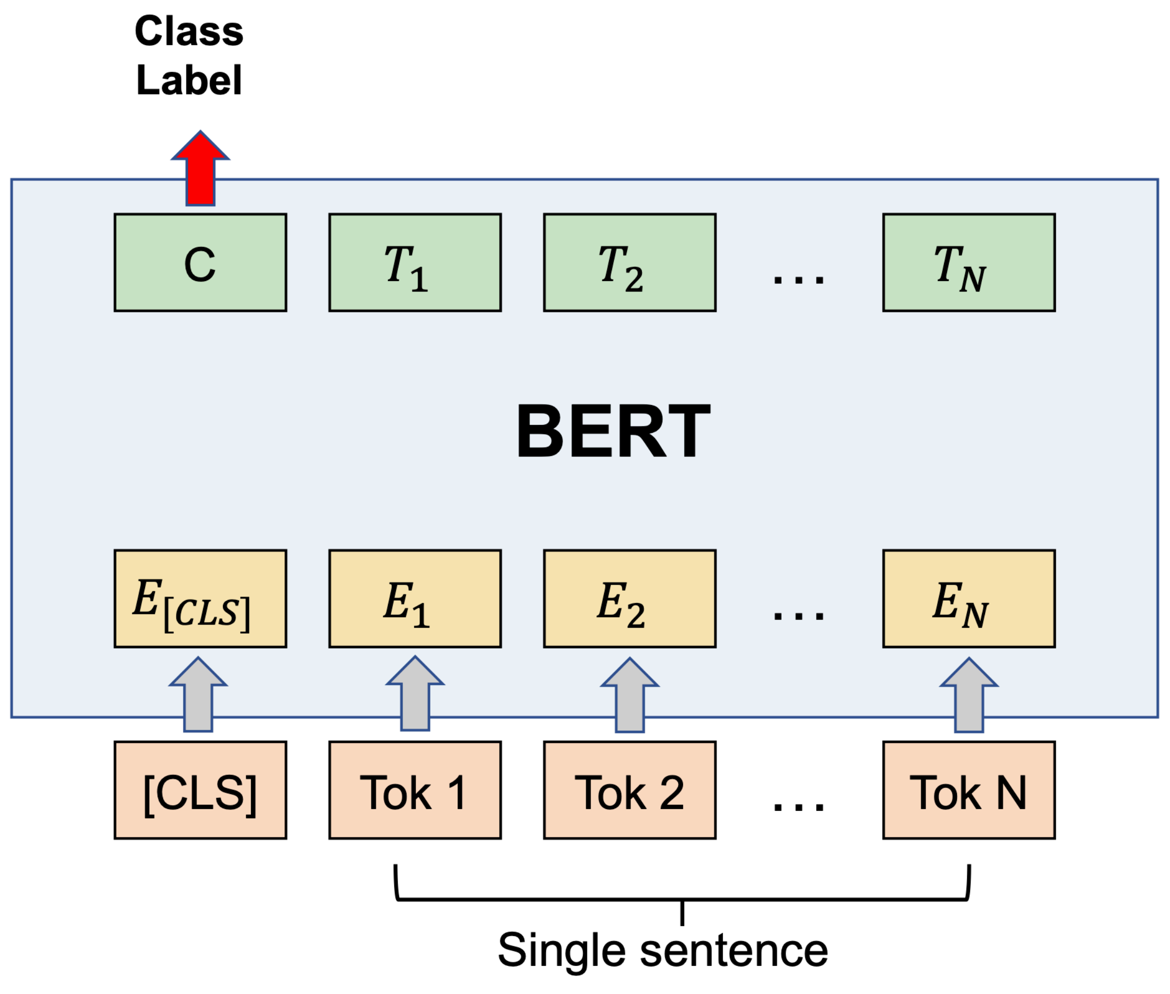
Sentence A

Sentence B

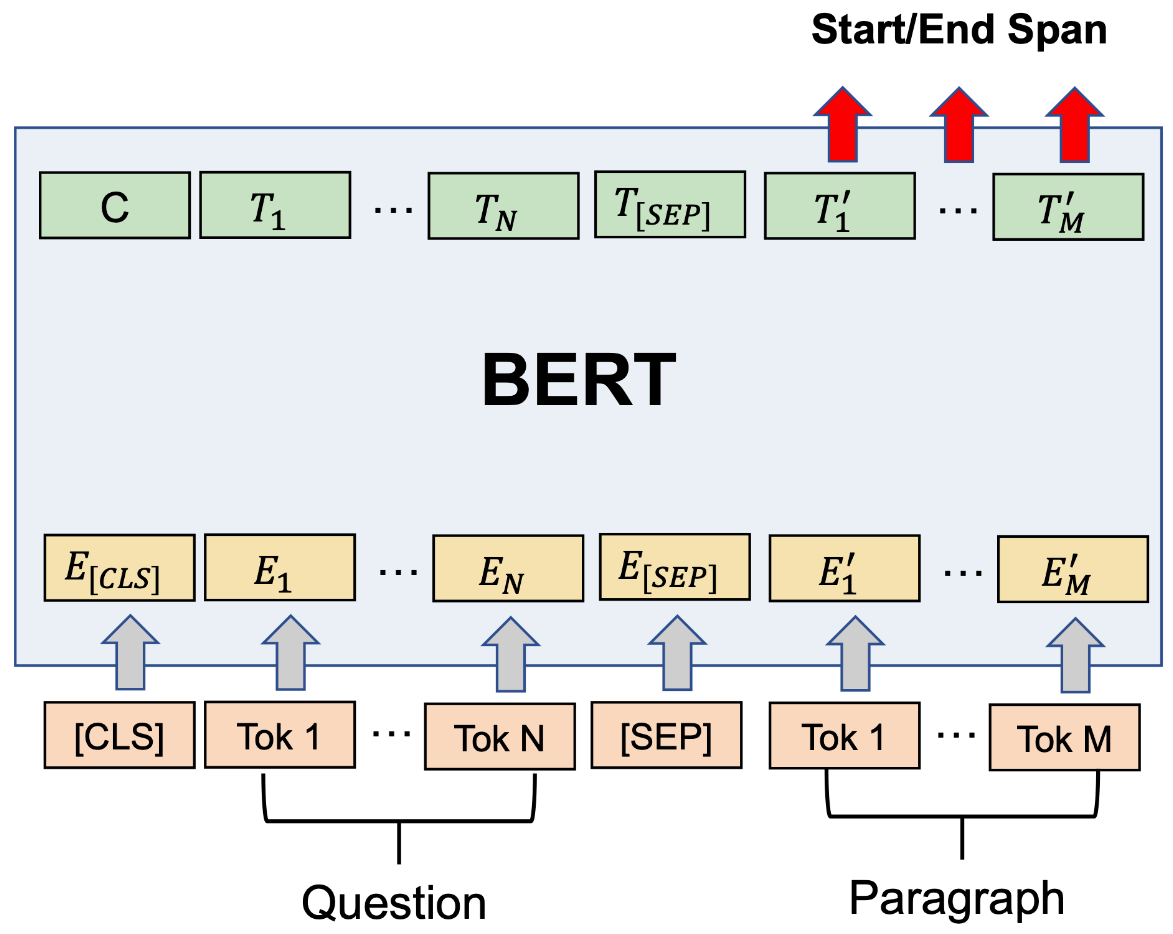
BERT Fine-tuning tasks



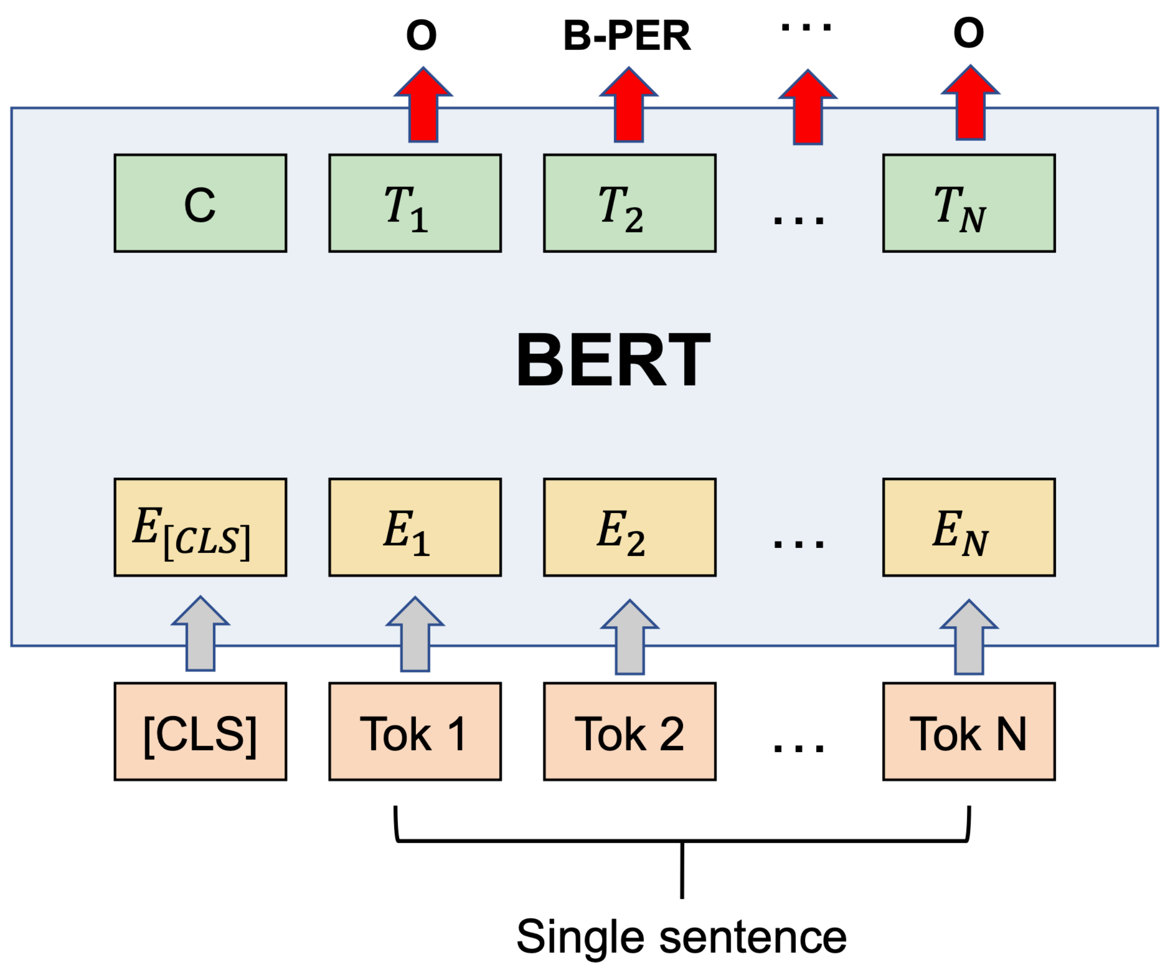
(a) Sentence pair classification tasks



(b) Single sentence classification tasks



(c) Question answering tasks

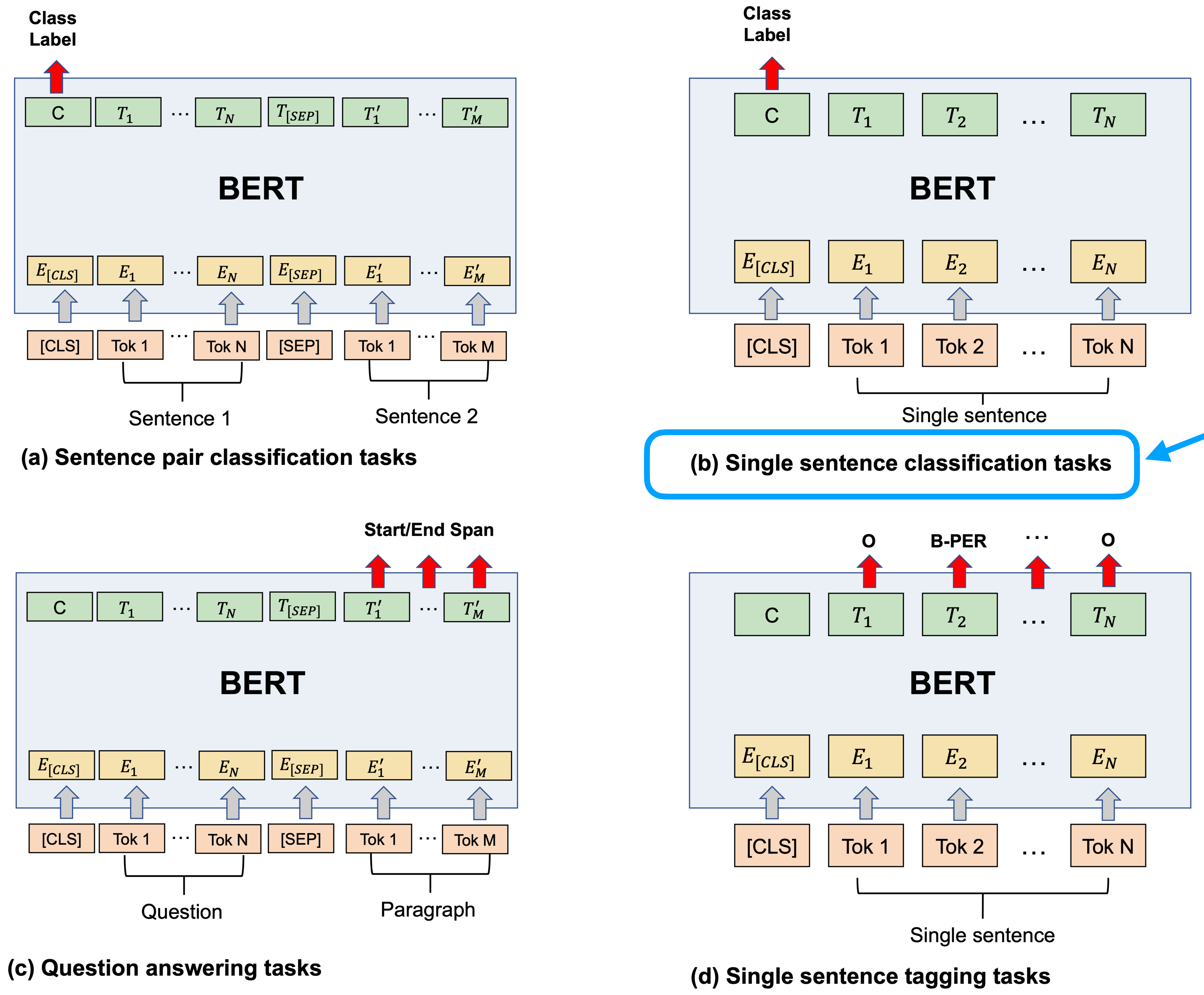


(d) Single sentence tagging tasks

Topics

1. Augmenting RNNs with attention
2. Self-attention
3. The original transformer architecture
4. Large-scale language models
5. **Fine-tuning a pre-trained BERT model in PyTorch**
6. Quo vadis, transformers?

Fine-tuning BERT



Multiple sentences can be concatenated (512 token limit)

Large Movie Review Dataset

This is a dataset for binary sentiment classification containing substantially more data than previous benchmark datasets. We provide a set of 25,000 highly polar movie reviews for training, and 25,000 for testing. There is additional unlabeled data for use as well. Raw text and already processed bag of words formats are provided. See the README file contained in the release for more details.

Large Movie Review Dataset v1.0

When using this dataset, please cite our ACL 2011 paper [bib].

Contact

For comments or questions on the dataset please contact Andrew Maas. As you publish papers using the dataset please notify us so we can post a link on this page.

Publications Using the Dataset

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. (2011). Learning Word Vectors for Sentiment Analysis. *The 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011).*

https://ai.stanford.edu/~amaas/data/sentiment/

Code notebook: https://github.com/rasbt/2021-pydata-jeddah

Topics

1. Augmenting RNNs with attention
2. Self-attention
3. The original transformer architecture
4. Large-scale language models
5. Fine-tuning a pre-trained BERT model in PyTorch
6. **Quo vadis, transformers?**

GPT-3
(175 billion)

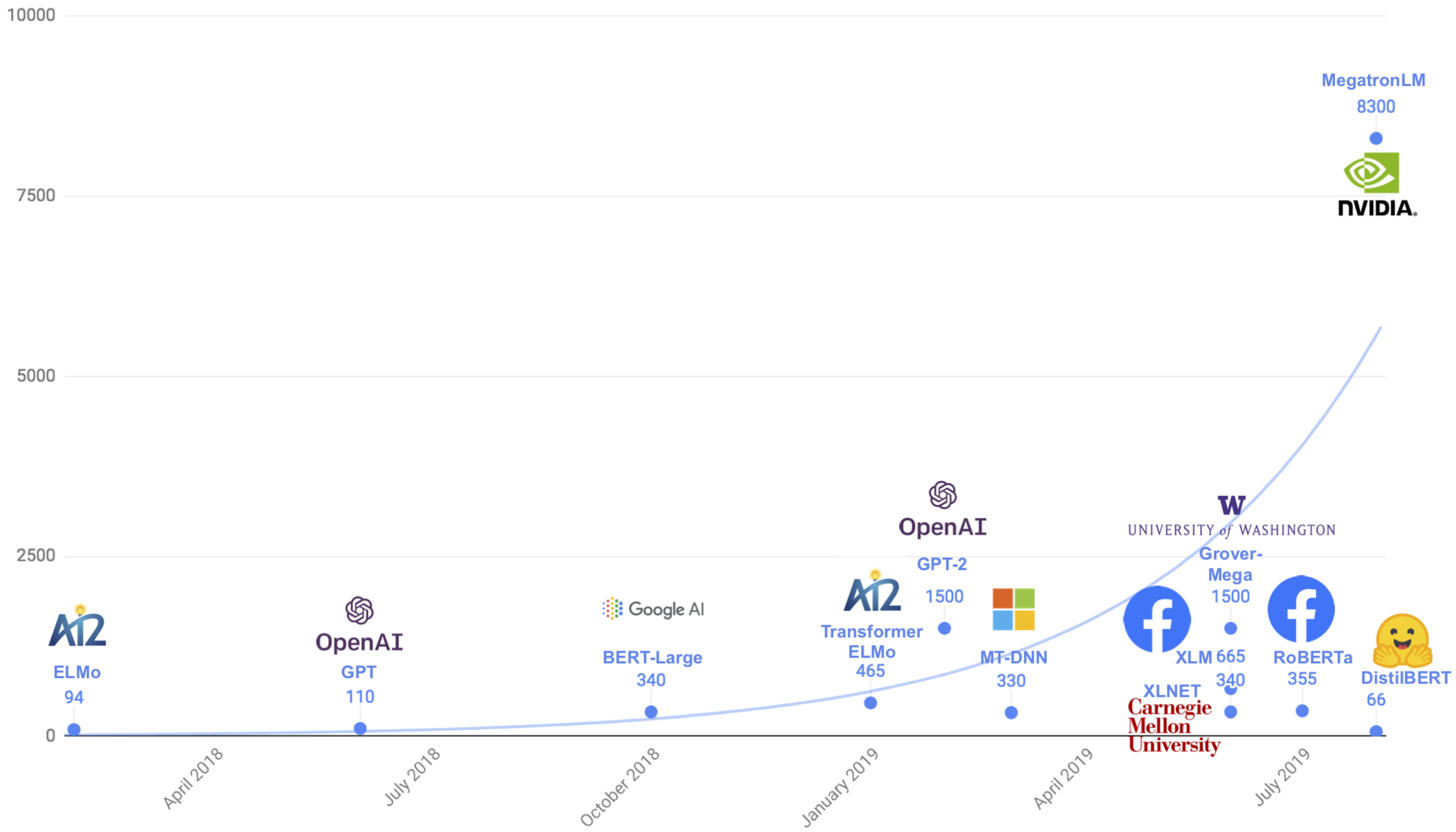


Image Source: https://medium.com/huggingface/distilbert-8cf3380435b5

THE COST OF TRAINING NLP MODELS
A CONCISE OVERVIEW

Or Sharir
AI21 Labs
ors@ai21.com

Barak Peleg
AI21 Labs
barakp@ai21.com

Yoav Shoham
AI21 Labs
yoavs@ai21.com

April 2020

http://arxiv.org/abs/2004.08900

- $2.5k - $50k (110 million parameter model)
- $10k - $200k (340 million parameter model)
- $80k - $1.6m (1.5 billion parameter model)

TECH \ ARTIFICIAL INTELLIGENCE

OpenAI's text-generating system GPT-3 is now spewing out 4.5 billion words a day

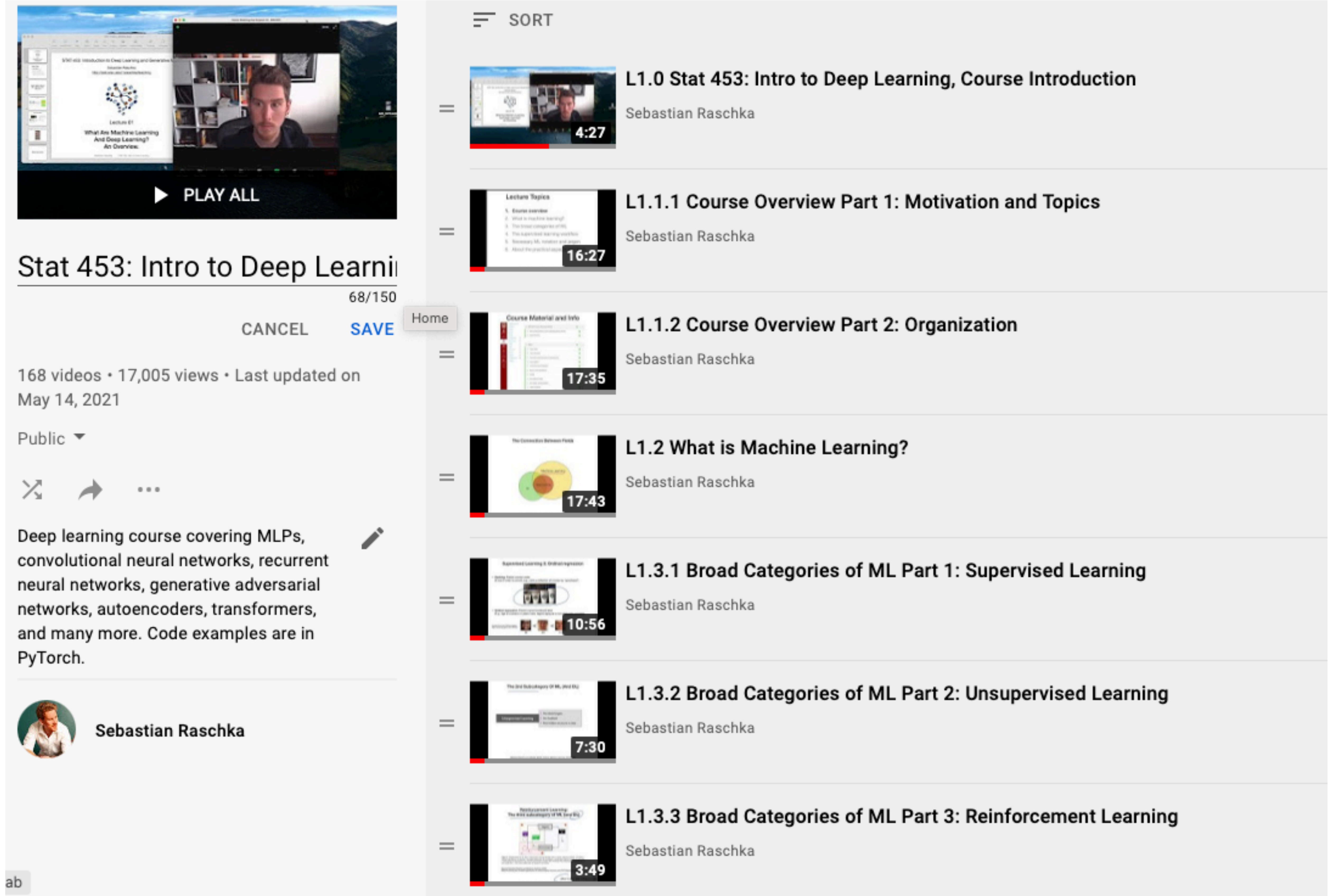
Robot-generated writing looks set to be the next big thing

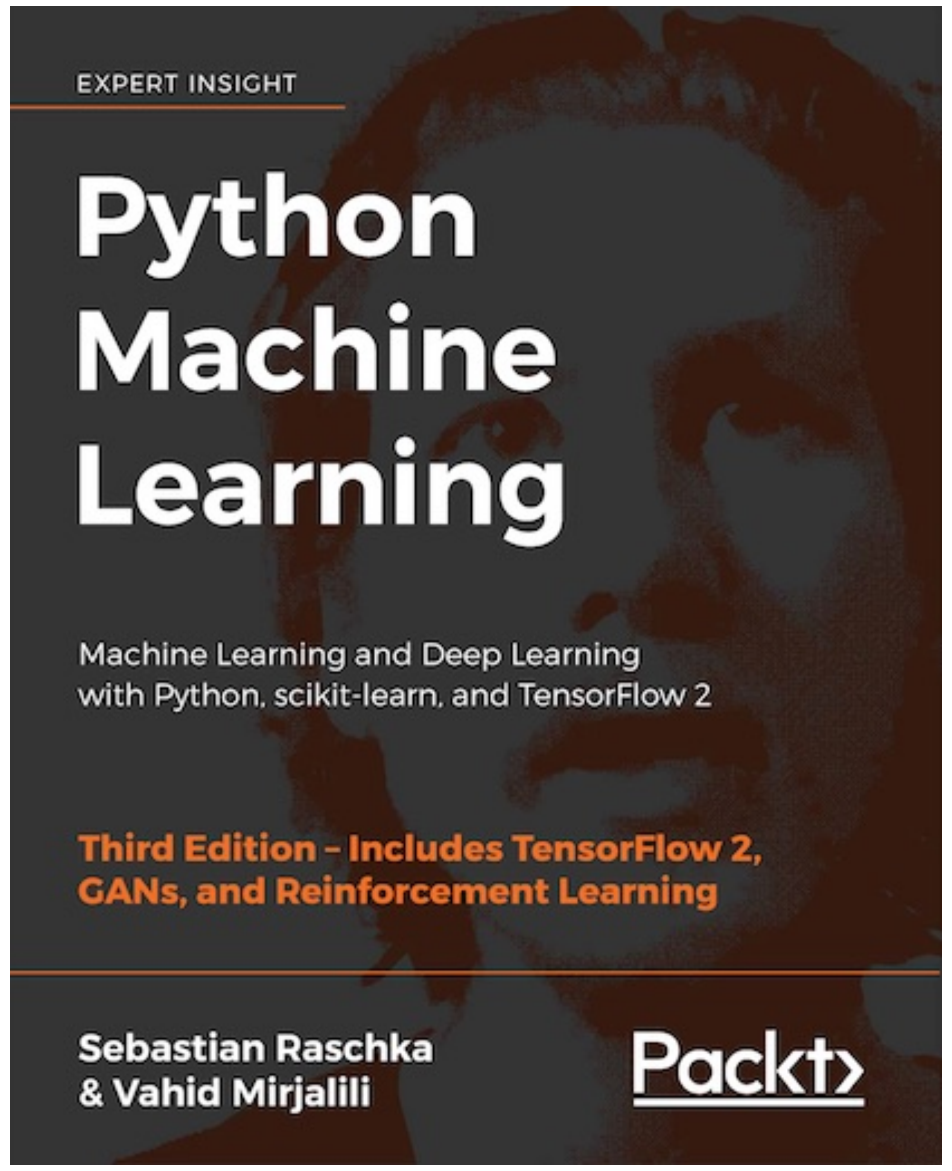
By James Vincent | Mar 29, 2021, 8:24am EDT

https://www.theverge.com/2021/3/29/22356180/openai-gpt-3-text-generation-words-day

Figure credits:

Python Machine Learning book chapter on Transformers
by Jitian Zhao and Sebastian Raschka





https://sebastianraschka.com/books/

YouTube https://www.youtube.com/playlist?list=PLTKMiZHVd_2KJtIXOW0zFhFfBaJJilH51

@rasbt

https://sebastianraschka.com/blog/2021/dl-course.html