BERT and Friends

CS 780/880 Natural Language Processing Lecture 20

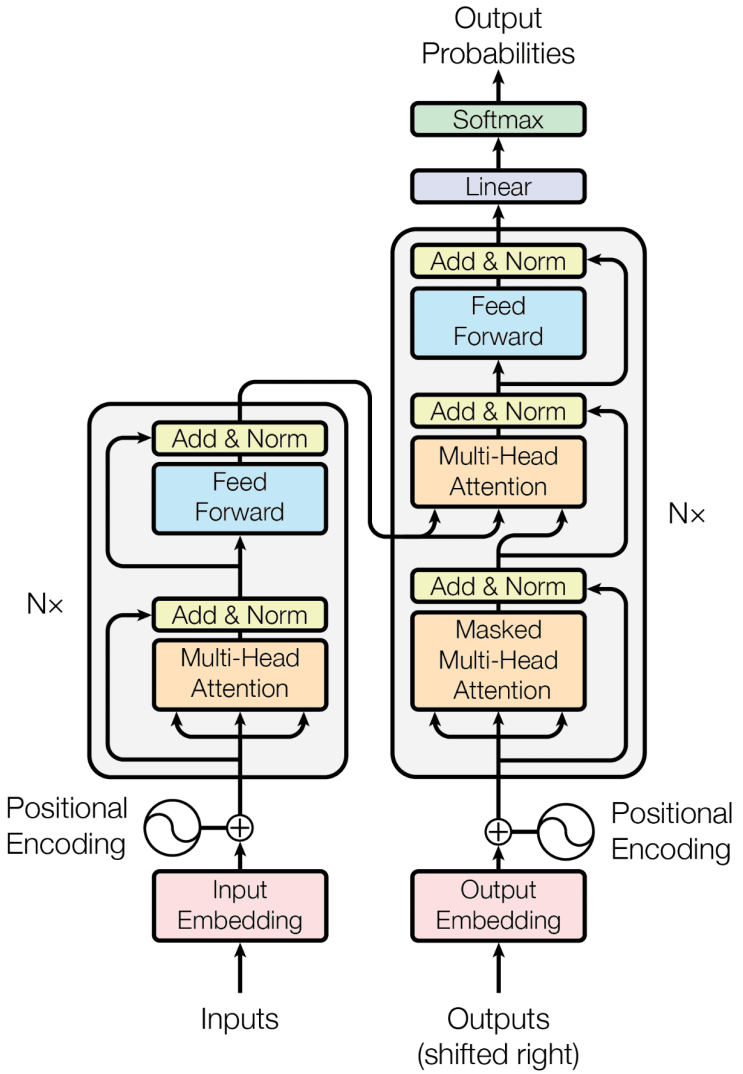
Samuel Carton, University of New Hampshire

Last lecture

Transformer architecture

- Many layers
  - **Self-attention**
  - Feed-forward
  - Residuals
- Encoder-decoder
  - Encoder nonrecurrent
  - Decoder recurrent
- Positional encodings

Pretrained transformer (BERT) is a good starting point for fine-tuning!

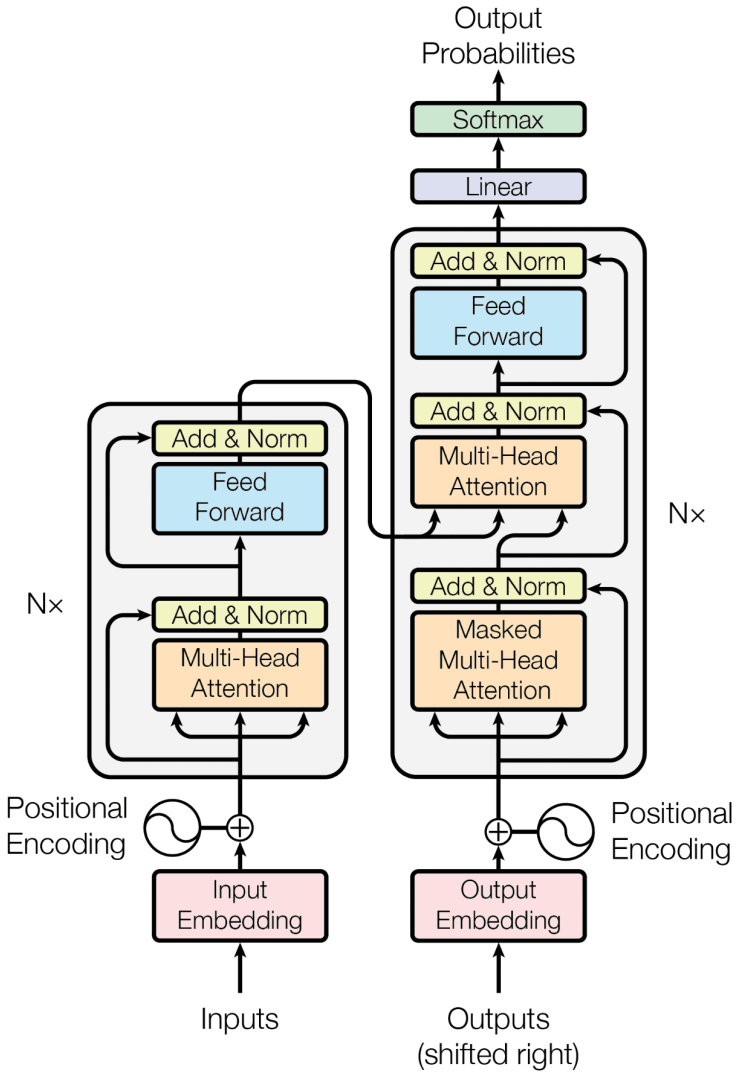


Last lecture

Transformer architecture

- Many layers
  - **Self-attention**
  - Feed-forward
  - Residuals
- Encoder-decoder
  - Encoder nonrecurrent
  - Decoder recurrent
- Positional encodings

Pretrained transformer is a good starting point for fine-tuning!



Pretrained transformers

Every current well-known large language model is a transformer that has been extensively **pretrained** on a large corpus of text, with some language modeling objective

- BERT, RoBERTa, T5, GPT-X, etc.

The difference between different models is mostly just:

- Training objective
- Use of encoder only, decoder only, or both
- Model size
- Training set size & composition
- Dataset preprocessing
- Minor architecture differences

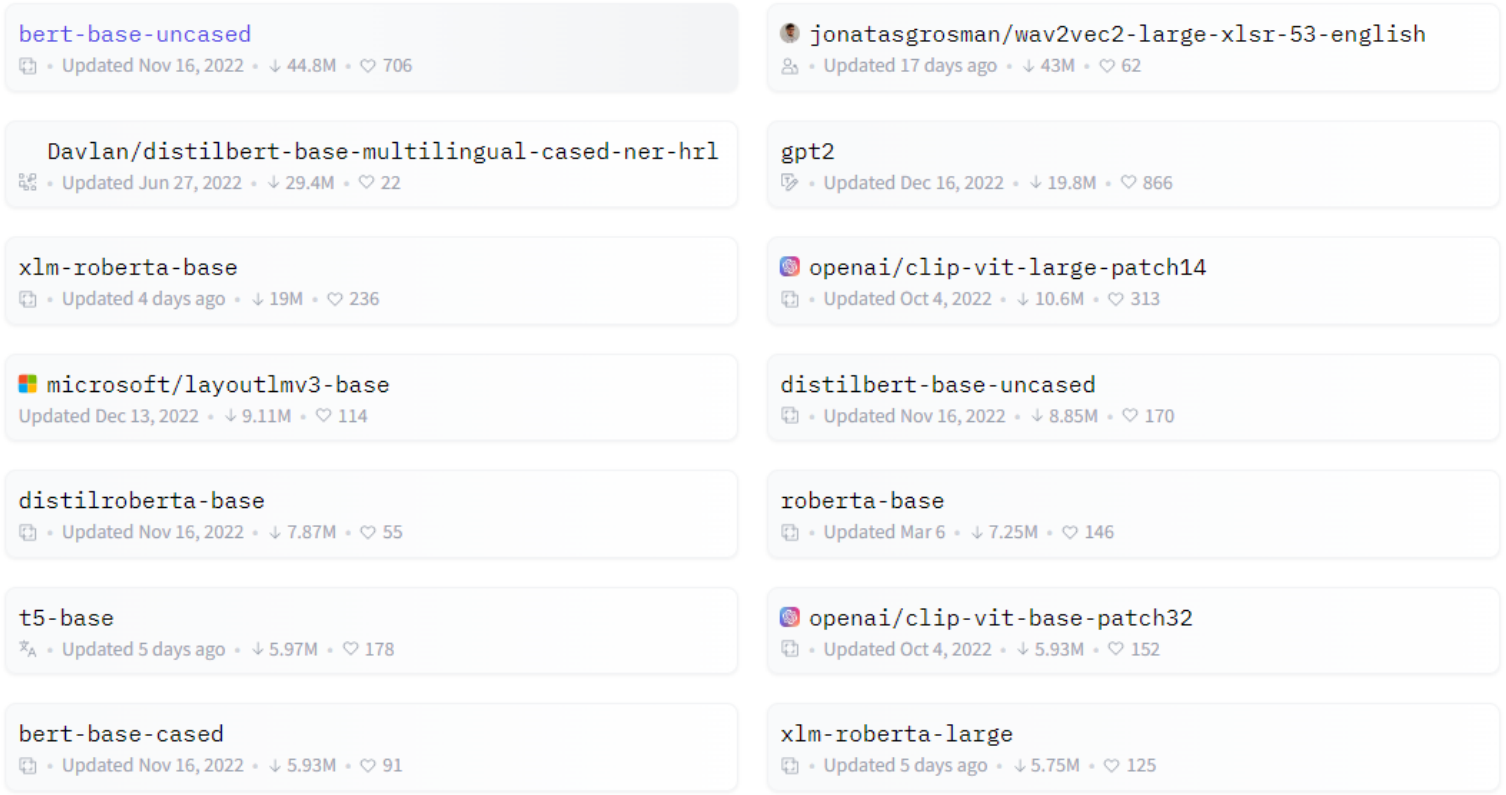
…which seems like a lot, but it's still pretty remarkable that the underlying model is mostly the same (Transformer)

Well-known models

There's a few key models that are in wide use:

- BERT
- RoBERTa
- XLNet
- DistilBERT
- T5
- GPT family

Most can be downloaded at https://huggingface.co/models



BERT

Bidirectional **E**ncoder **R**epresentations from **T**ransformers

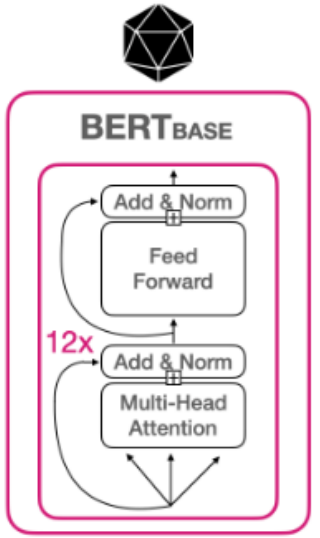
Encoder-only model

Bert-base:

- 12 layers, 12 heads per layer
- 110 million parameters

Two pretraining objectives:

- **Masked language modeling (Mask-LM)**
- **Next sentence prediction (NSP)**

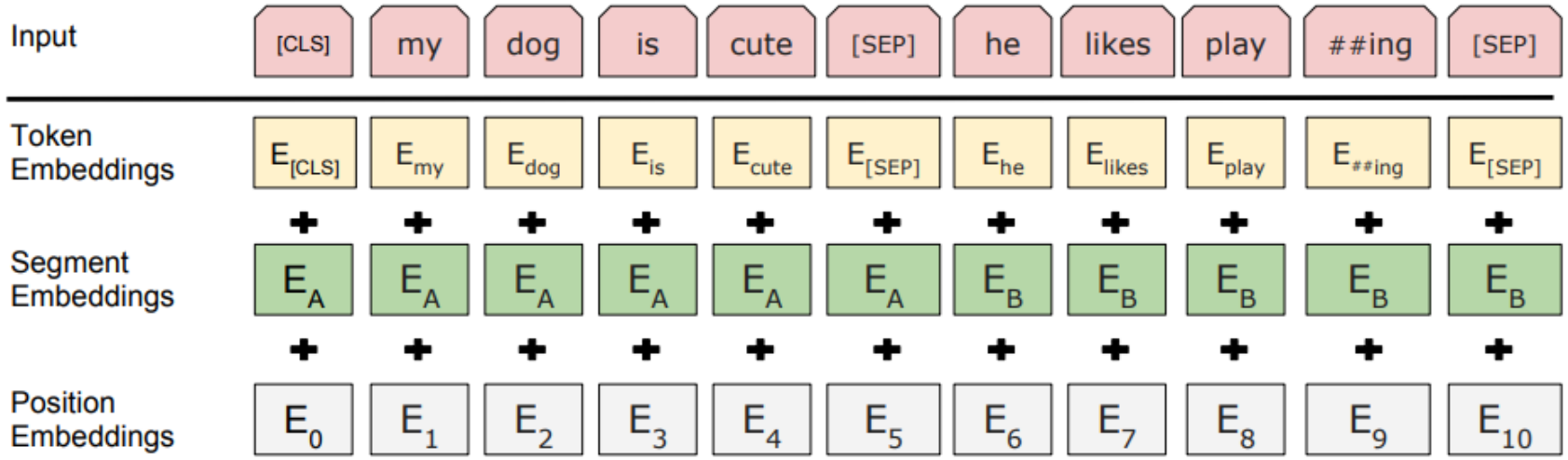


Bert: Pre-training of deep bidirectional transformers for language understanding
J Devlin, MW Chang, K Lee, K Toutanova - arXiv preprint arXiv ..., 2018 - arxiv.org
We introduce a new language representation model called BERT, which stands for Bidirectional Encoder Representations from Transformers. Unlike recent language representation models, BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and ...
☆ Save 99 Cite Cited by 63400 Related articles All 39 versions »

BERT encoder

The BERT encoder:

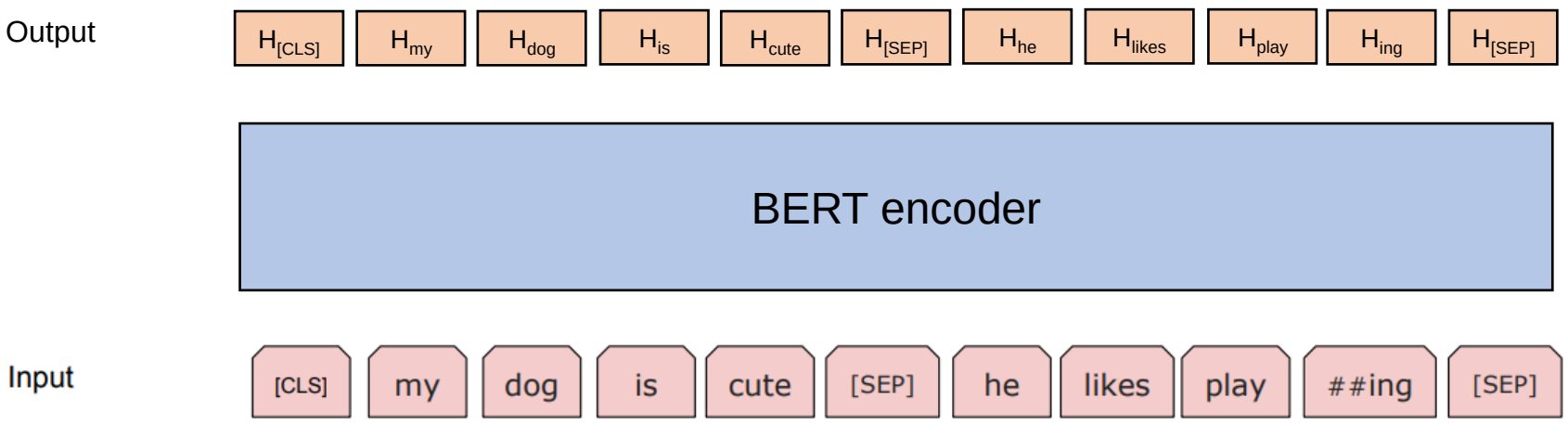
1. Takes in wordpieces
2. With [CLS] at the beginning and [SEP] between sentences
3. Adds positional and segment ID (0 or 1) embeddings
4. Outputs a hidden state vector for each wordpiece (including [CLS] and [SEP]s)



BERT encoder

The BERT encoder:

1. Takes in wordpieces
2. With [CLS] at the beginning and [SEP] between sentences
3. Adds positional and segment ID (0 or 1) embeddings
4. Outputs a hidden state vector for each wordpiece (including [CLS] and [SEP]s)



BERT pretraining

Mask-LM: Randomly mask 15% of tokens and try to predict them from H_{token}

NSP: Randomly sample correct/incorrect sentence pairs, try to predict which is correct from $H_{[CLS]}$

A term used for this overall approach is **denoising autoencoding**

- "Denoising" because it tries to correct missing tokens
- "Autoencoding" because it tries to encode unlabeled text to a vector representation

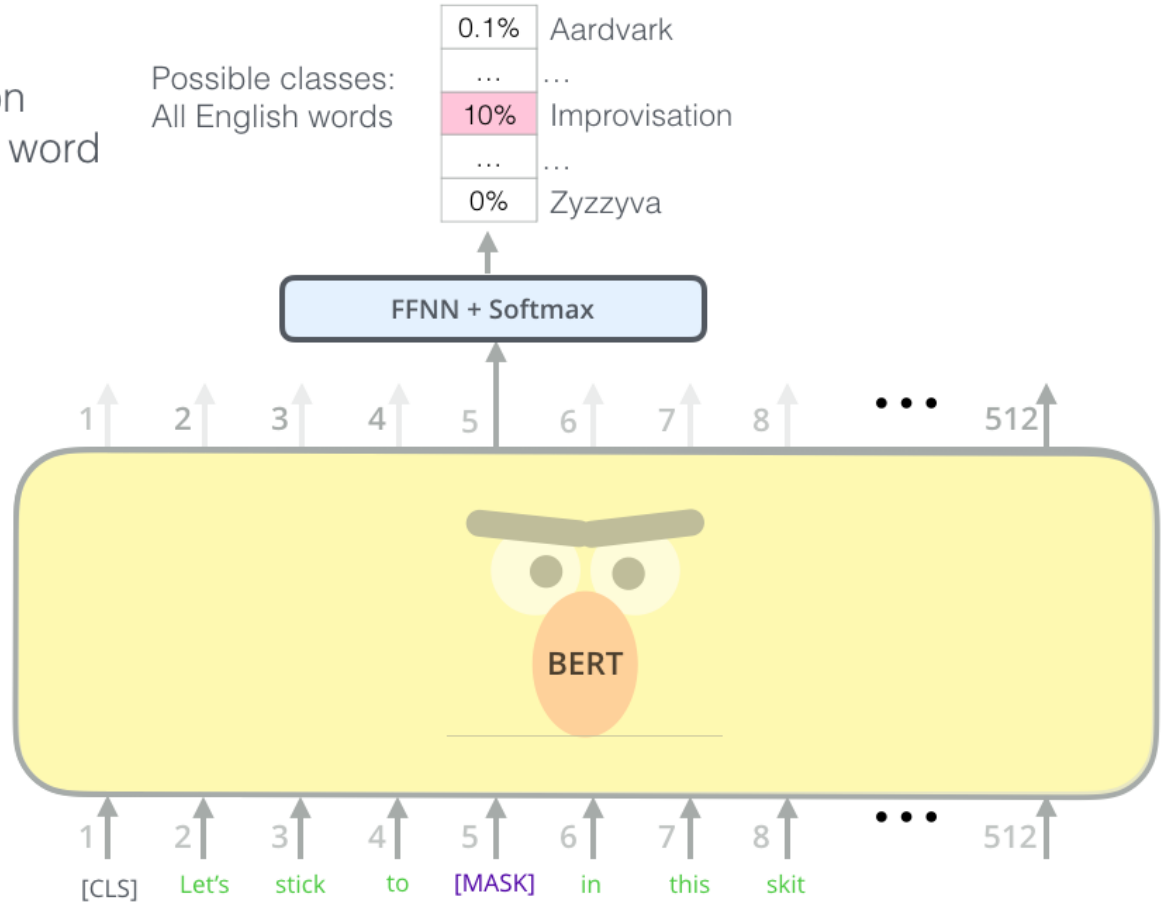
Pretraining corpus:

- BooksCorpus (800M words)
- English Wikipedia (2,500M words)

$BERT_{LARGE}$ is also available (24 layers, 16 heads per layer, 340M params)

Masked language modeling

Use the output of the masked word's position to predict the masked word

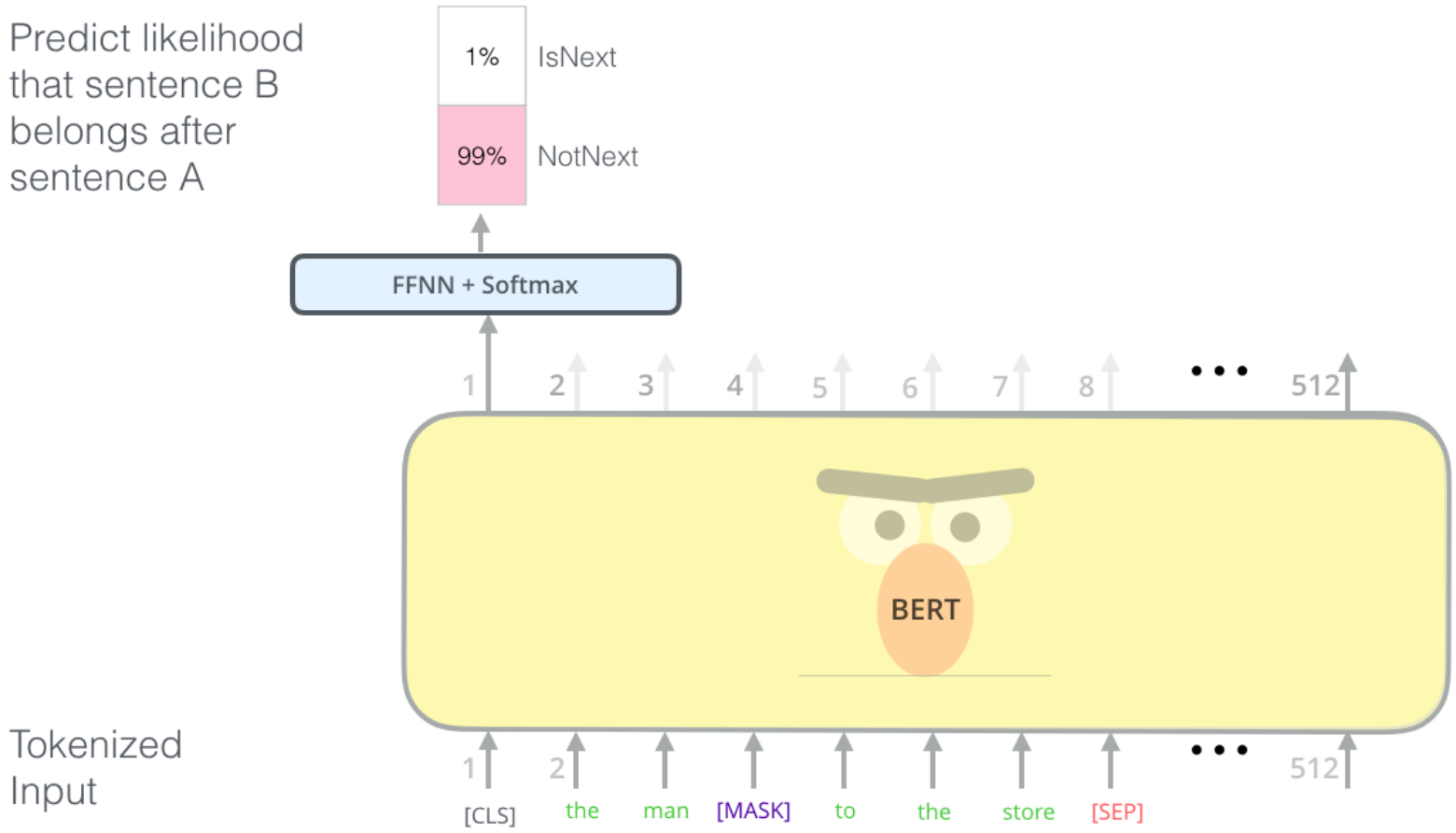


Randomly mask 15% of tokens

Input

https://jalammar.github.io/illustrated-bert/

Next-sentence prediction



https://jalammar.github.io/illustrated-bert/

Deep contextualized representations

A key thing about these models is that they produce **deep contextualized representations** of their input

- A single vector that represents the whole sequence ($H_{[CLS]}$) or an individual token (H_{token})
- The vector reflects the **context** surrounding that token.
  - So H_{jerk} will be different for “You are a jerk.” versus “I like jerk chicken”
  - Compare and contrast to word vectors

With large scale pretraining, we have models which can produce useful representations of input, which we can then fine-tune to do specific things

- Kind of like teaching someone English before trying to teach them to grade papers

RoBERTa

Essentially a refinement/exploration of BERT

- Same architecture & training data
  - Also encoder-only
- **Ditches NSP**
- Does “dynamic” mask-LM
- Improved performance on NLP benchmarks

Roberta: A robustly optimized bert pretraining approach
Y Liu, M Ott, N Goyal, J Du, M Joshi, D Chen... - arXiv preprint arXiv ..., 2019 - arxiv.org
... configuration **RoBERTa** for **Robustly optimized BERT approach**. Specifically, **RoBERTa** is
... (eg, the **pretraining** objective), we begin by training **RoBERTa** following the BERTLARGE ...
☆ Save 99 Cite Cited by 6469 Related articles All 5 versions »

	MNLI	QNLI	QQP	RTE	SST	MRPC	CoLA	STS	WNLI	Avg
Single-task single models on dev										
$\text{BERT}_{\text{LARGE}}$	86.6/-	92.3	91.3	70.4	93.2	88.0	60.6	90.0	-	-
$\text{XLNet}_{\text{LARGE}}$	89.8/-	93.9	91.8	83.8	95.6	89.2	63.6	91.8	-	-
RoBERTa	**90.2/90.2**	**94.7**	**92.2**	**86.6**	**96.4**	**90.9**	**68.0**	**92.4**	**91.3**	-

Comparable size to $\text{BERT}_{\text{LARGE}}$

- 24 layers, 16 heads per layer, 355M params total

Probably a better default choice than BERT, if you have the GPU memory

XLNet

Another competitor of BERT that occasionally shows up in the literature

Also uses **only** the encoder

Pretrains using a variant of autoregressive language modeling called **permutation language modeling**

Comparison with BERT

- Same size
- Additional training data:
  - ClueWeb
  - Common Crawl
- Broadly improved performance

Autoregressive language modeling

Basic idea: train the model to be most likely to reproduce the training data

We've learned this before (a couple times), but this is alternative terminology.

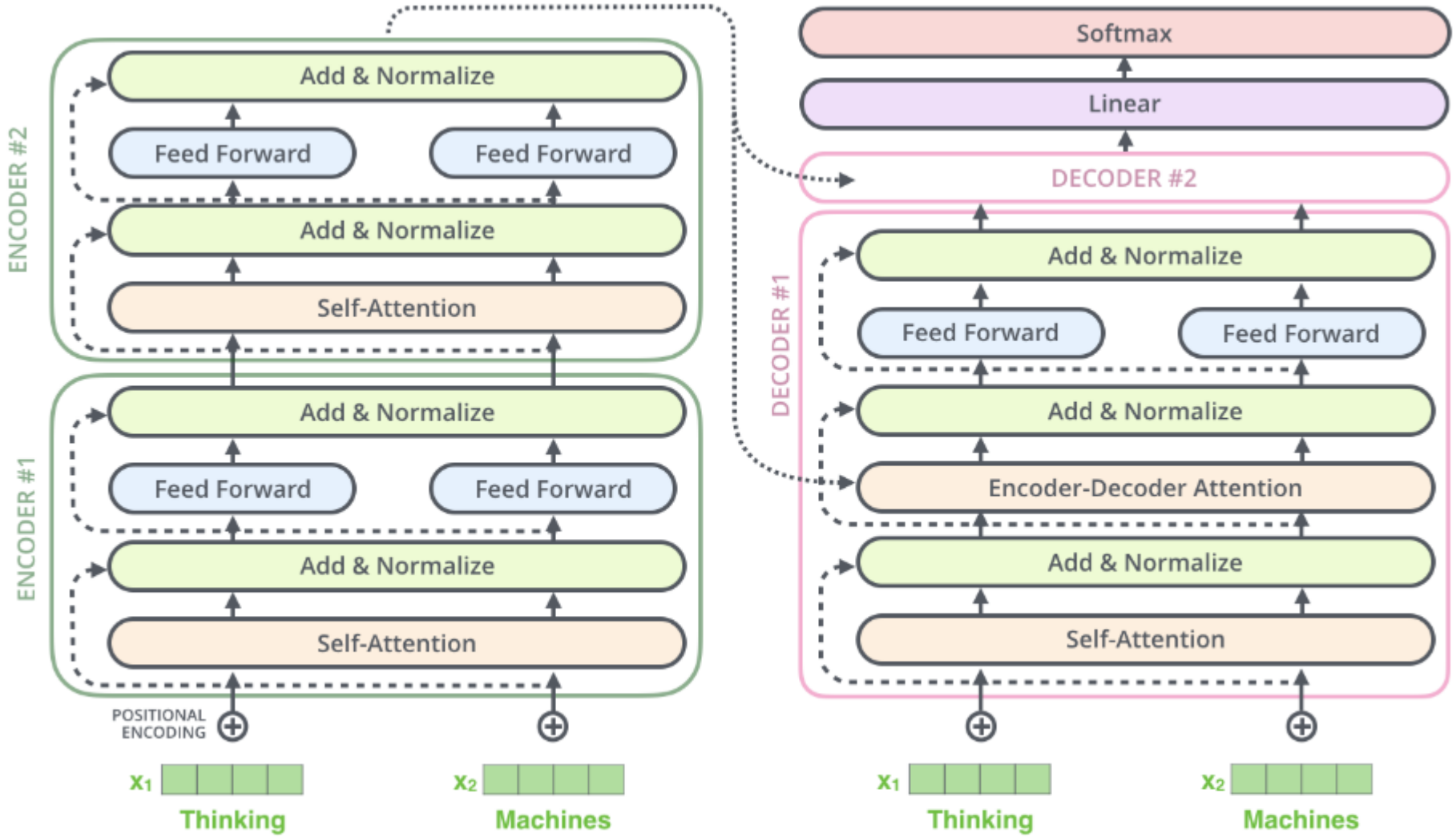
Based on a **forward factorization** of the text where each x_t is dependent on $\{x_0 \ldots x_{t-1}\}$, so we can factorize the overall likeliness of x as a sum of log-probabilities of each individual x_t:

$$\max_{\theta} \quad \log p_\theta(\mathbf{x}) = \sum_{t=1}^{T} \log p_\theta(x_t \mid \mathbf{x}_{<t}) = \sum_{t=1}^{T} \log \frac{\exp\left(h_\theta(\mathbf{x}_{1:t-1})^\top e(x_t)\right)}{\sum_{x'} \exp\left(h_\theta(\mathbf{x}_{1:t-1})^\top e(x')\right)},$$

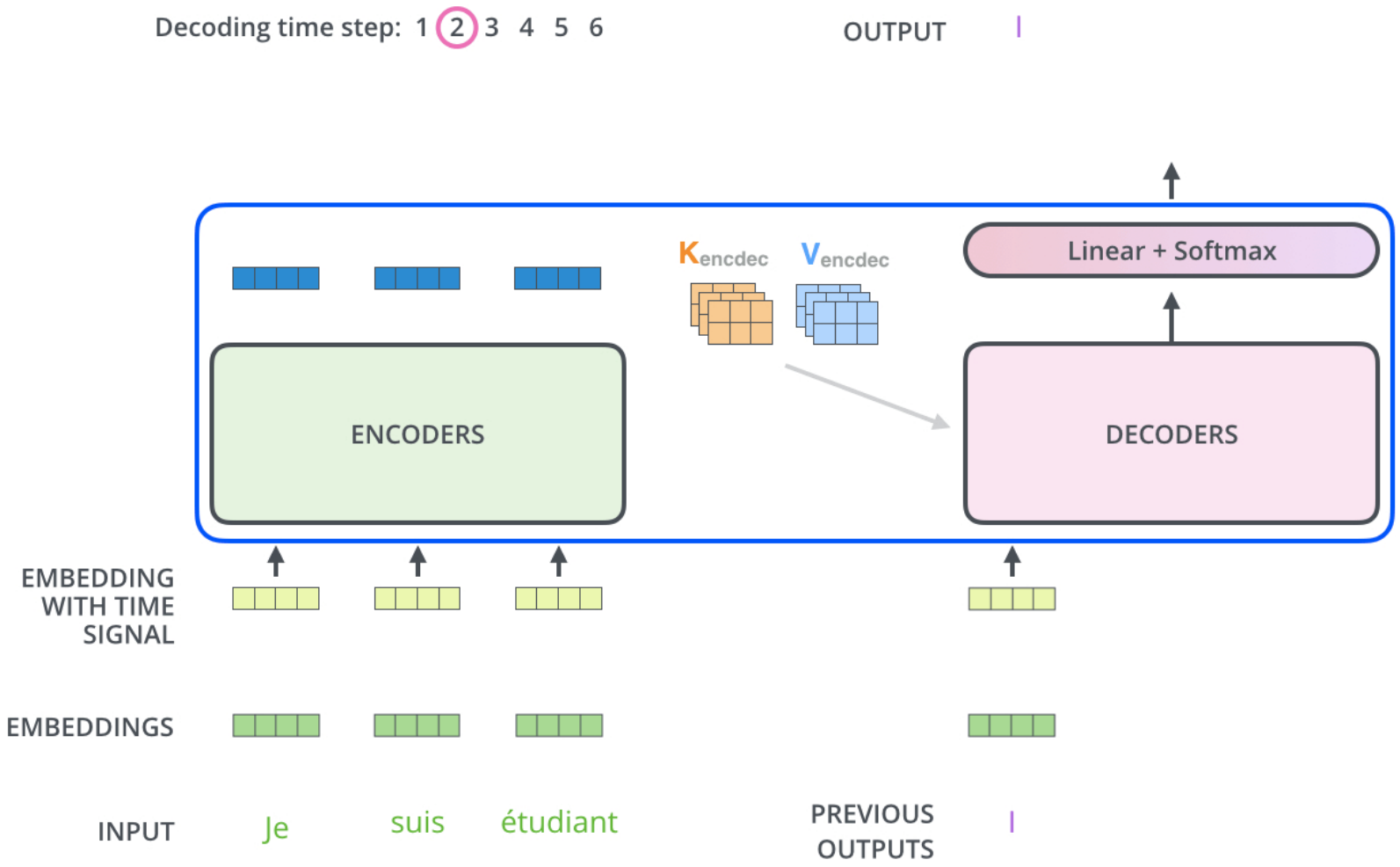
Several options for exactly how to do this:

- **Teacher forcing**: each $x_{<t}$ is drawn from the true data
- **Naïve autoregression**: each $x_{<t}$ is the one generated by the model

Autoregressive language modeling



Autoregressive decoding



XLNET: Permutation language modeling

Rather than only optimizing for token likelihood in forward factorization, XLNet optimizes for every possible permutation of the text

So not just $P(X_3 \mid X_1, X_2)$, but also $P(X_3)$, $P(X_3|X_4)$, $P(X_3|X_1, X_4, X_2)$, etc..

But doesn't use decoder!

- Instead manipulates model attention and positional encodings to erase and reorder tokens from input

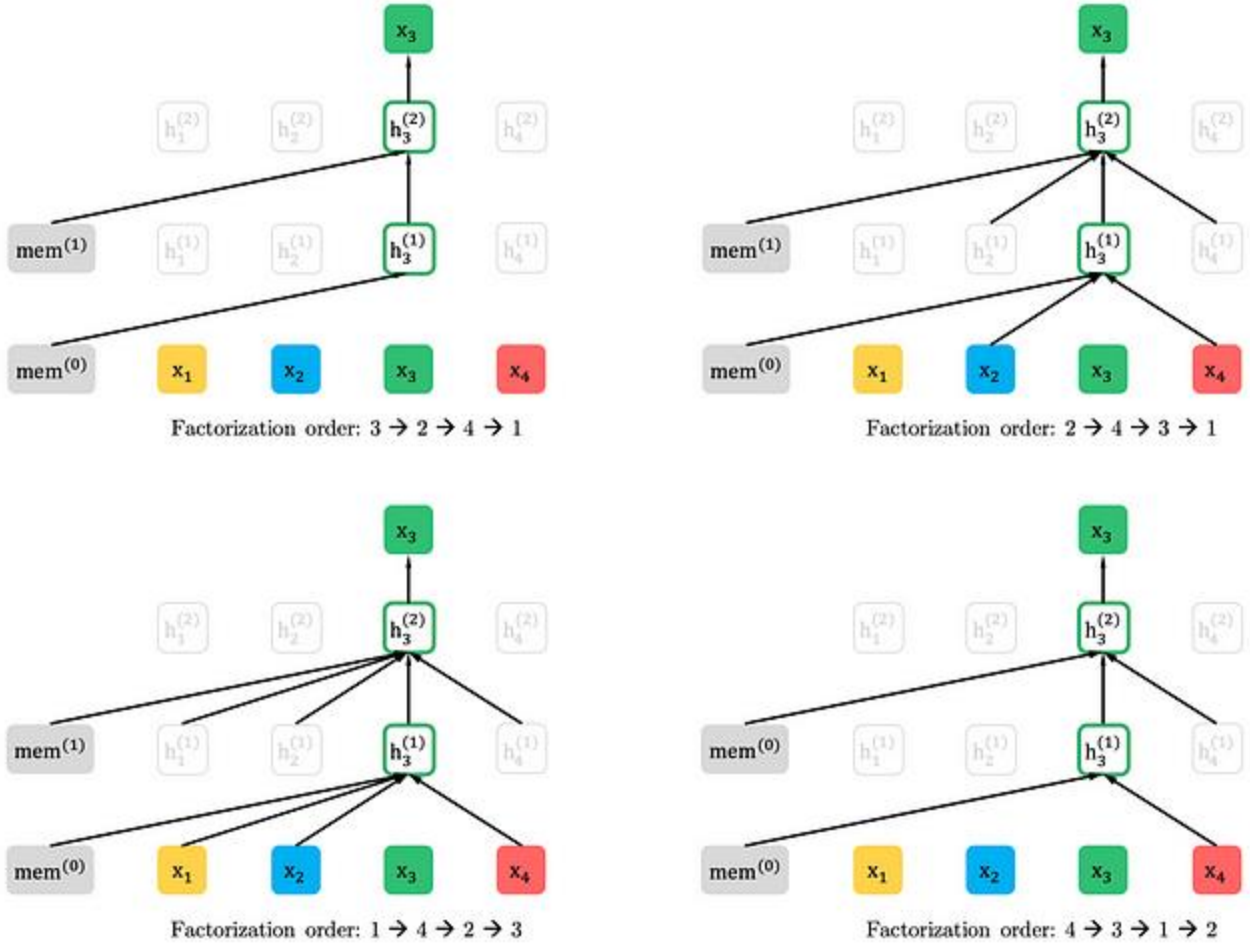
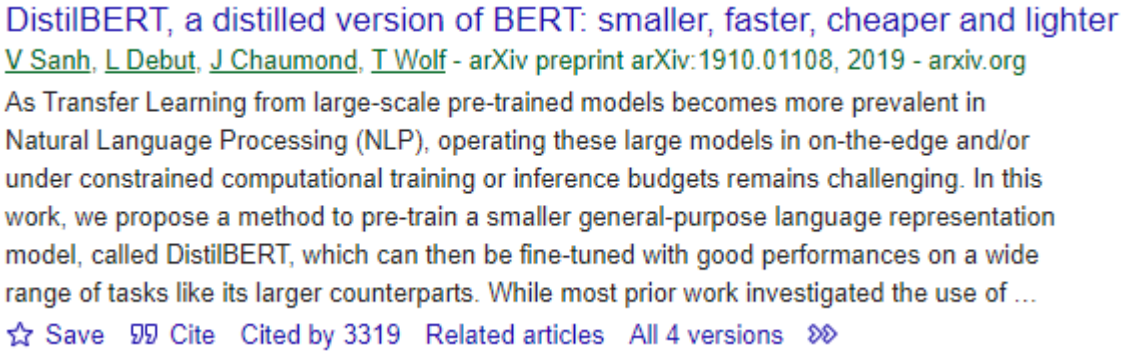


Figure 1: Illustration of the permutation language modeling objective for predicting x_3 given the same input sequence x but with different factorization orders.

DistilBERT

A version of BERT that has been reduced in size from BERT by a process called **knowledge distillation**


DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter
V Sanh, L Debut, J Chaumond, T Wolf - arXiv preprint arXiv:1910.01108, 2019 - arxiv.org
As Transfer Learning from large-scale pre-trained models becomes more prevalent in Natural Language Processing (NLP), operating these large models in on-the-edge and/or under constrained computational training or inference budgets remains challenging. In this work, we propose a method to pre-train a smaller general-purpose language representation model, called DistilBERT, which can then be fine-tuned with good performances on a wide range of tasks like its larger counterparts. While most prior work investigated the use of ...
Save Cite Cited by 3319 Related articles All 4 versions

Knowledge distillation:

- Big (trained) teacher model and small student model
- Train student model to emulate teacher model
- Different from regular training because teacher model produces nonzero probabilities over other possible classes, which is richer training data than 1's and 0's
  - Kind of like explaining that a shape in a CT scan is a tumor, but also looks like a cyst, rather than "it's just a tumor and not a cyst"

97% of the performance of BERT, but 40% smaller and 60% faster

- 6 layers, 12 heads per layer, 66M parameters

T5

Important model. Really the first big improvement from the BERT variants.

Uses the full encoder-decoder apparatus of the Transformer architecture

Does a combination of unsupervised language modeling and supervised text-to-text modeling

Fine-tuned T5 is still pretty close to SoTA for many NLP tasks

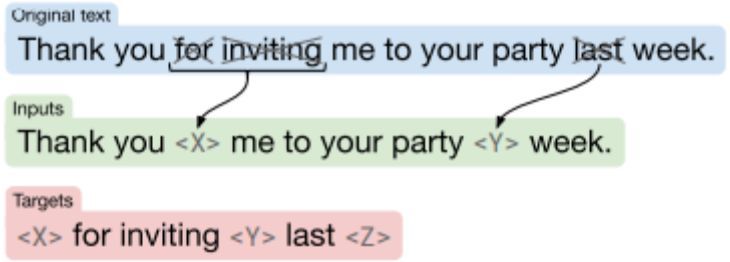
Exploring the **limits** of **transfer learning** with a **unified** text-to-text transformer
C Raffel, N Shazeer, A Roberts, K Lee... - ... of Machine **Learning** ..., 2020 - dl.acm.org
... The effectiveness of **transfer learning** has given rise to a diversity of approaches, ... In this paper, we **explore** the landscape of **transfer learning** techniques for NLP by introducing a **unified** ...
☆ Save 99 Cite Cited by 6961 Related articles All 14 versions Web of Science: 1361

T5 pretraining—unsupervised

Creates a big, cleaned-up unsupervised training corpus:
“Colossal Clean Crawled Corpus”: cleaned-up version of Common Crawl

Uses variant of masked-LM objective from BERT, mapping corrupted text to true text

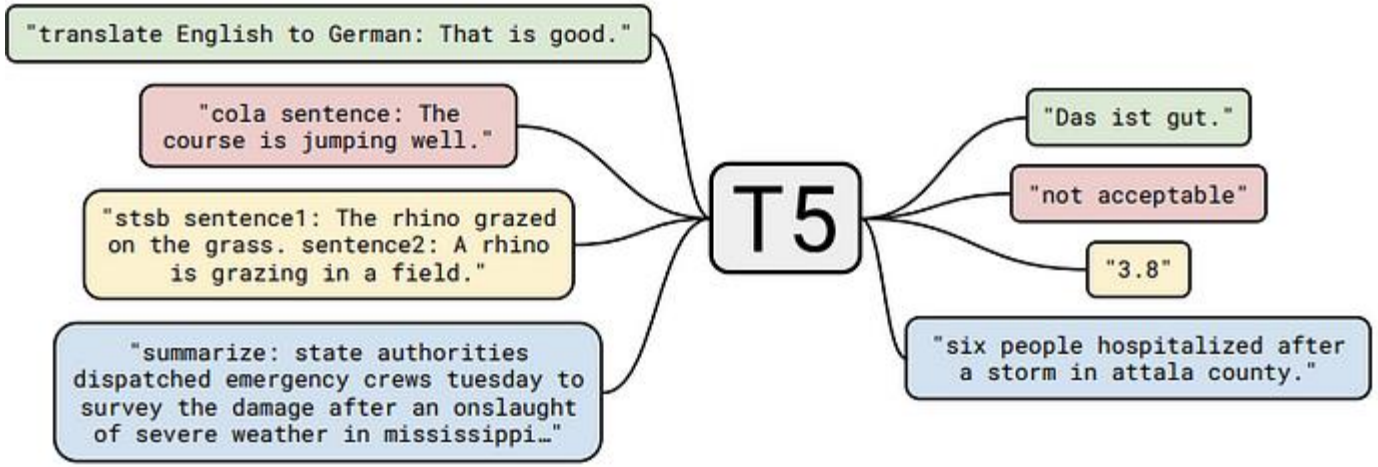
- Can mask out contiguous sequences of tokens at once
- Uses teacher-forcing to train decoder



T5 pretraining—supervised

Also converts a diverse set of supervised learning datasets into text-to-text tasks, and trains on them

- GLUE and SuperGLUE



T5

Most common model is T5-11b (11 billion parameters), but smaller variants also exist

Fine-tuned T5-11b is still pretty competitive in NLP benchmarks

Leaderboard Version: **2.0**

Rank	Name	Model	URL	Score	BoolQ	CB	COPA	MultiRC	ReCoRD	RTE	WiC	WSC	AX-b	AX-g
1	JDExplore d-team	Vega v2		91.3	90.5	98.6/99.2	99.4	88.2/62.4	94.4/93.9	96.0	77.4	98.6	-0.4	100.0/50.0
2	Liam Fedus	ST-MoE-32B		91.2	92.4	96.9/98.0	99.2	89.6/65.8	95.1/94.4	93.5	77.7	96.6	72.3	96.1/94.1
3	Microsoft Alexander v-team	Turing NLR v5		90.9	92.0	95.9/97.6	98.2	88.4/63.0	96.4/95.9	94.1	77.1	97.3	67.8	93.3/95.5
4	ERNIE Team - Baidu	ERNIE 3.0		90.6	91.0	98.6/99.2	97.4	88.6/63.2	94.7/94.2	92.6	77.4	97.3	68.6	92.7/94.7
5	Yi Tay	PaLM 540B		90.4	91.9	94.4/96.0	99.0	88.7/63.6	94.2/93.3	94.1	77.4	95.9	72.9	95.5/90.4
6	Zirui Wang	T5 + UDG, Single Model (Google Brain)		90.4	91.4	95.8/97.6	98.0	88.3/63.0	94.2/93.5	93.0	77.9	96.6	69.1	92.7/91.9
7	DeBERTa Team - Microsoft	DeBERTa / TuringNLRv4		90.3	90.4	95.7/97.6	98.4	88.2/63.7	94.5/94.1	93.2	77.5	95.9	66.7	93.3/93.8
8	SuperGLUE Human Baselines	SuperGLUE Human Baselines		89.8	89.0	95.8/98.9	100.0	81.8/51.9	91.7/91.3	93.6	80.0	100.0	76.6	99.3/99.7
9	T5 Team - Google	T5		89.3	91.2	93.9/96.8	94.8	88.1/63.3	94.1/93.4	92.5	76.9	93.8	65.6	92.7/91.9

https://super.gluebenchmark.com/leaderboard

GPT-1

Precursor model to GPT-2, GPT-3, and GPT-4

Decoder-only. Does not encode entire input sequence—rather, just encodes input sequence token-by-token

Trained using standard autoregressive language modeling objective

- Uses teacher forcing (I believe)

Trained on BooksCorpus dataset

12 layers, 12 heads per layer, 120M parameters

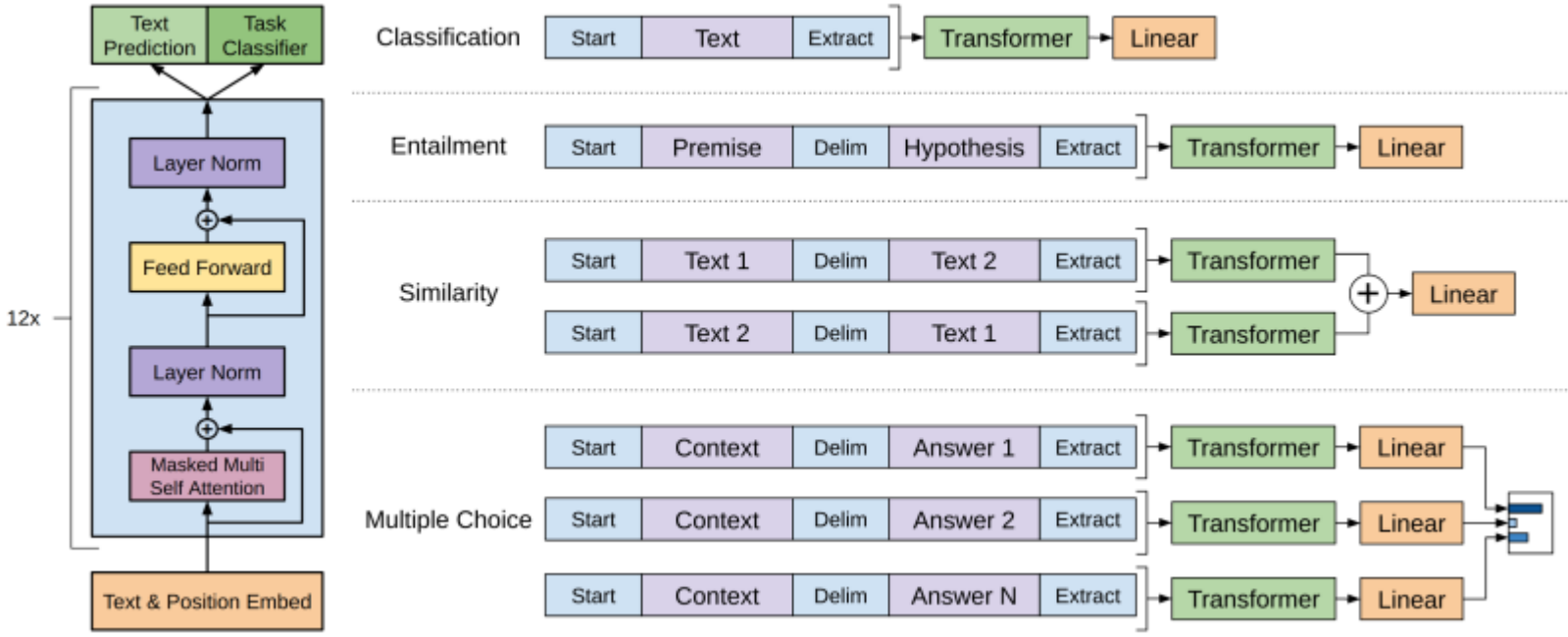
[PDF] Improving language understanding by generative pre-training

A Radford, K Narasimhan, T Salimans, I Sutskever - 2018 - cs.ubc.ca

Natural language understanding comprises a wide range of diverse tasks such as textual entailment, question answering, semantic similarity assessment, and document classification. Although large unlabeled text corpora are abundant, labeled data for learning these specific tasks is scarce, making it challenging for discriminatively trained models to perform adequately. We demonstrate that large gains on these tasks can be realized by generative pre-training of a language model on a diverse corpus of unlabeled text, followed ...

☆ Save 99 Cite Cited by 5179 Related articles All 9 versions »

GPT-1 fine-tuning



GPT-2,3,4

All larger and more capable versions of GPT-1

Same model with slight modifications

Same or larger datasets

GPT-2: 1.5B parameters
GPT-3: 175B parameters
GPT-4: ???????????

How to choose?

For general-purpose NLP fine-tuning, use the biggest model you can train:

- T5-11b → RoBERTa-Large → BERT-base → DistilBERT

For text generation, GPT-2 or GPT-Neo

For specific domains, try to find domain-specific versions of models

- E.g. MatSciBERT for materials-science specific NLP tasks
- Hugging Face has a nice search interface

Important to try multiple models

Concluding thoughts

Pretrained transformer models

- BERT, RoBERTa, XLNet, RoBERTa, DistilBERT, T5, GPT-X

Encoder-decoder, encoder-only, decoder-only

How to choose?

Looking forward:

- Zero- and few-shot learning
- Prompt engineering