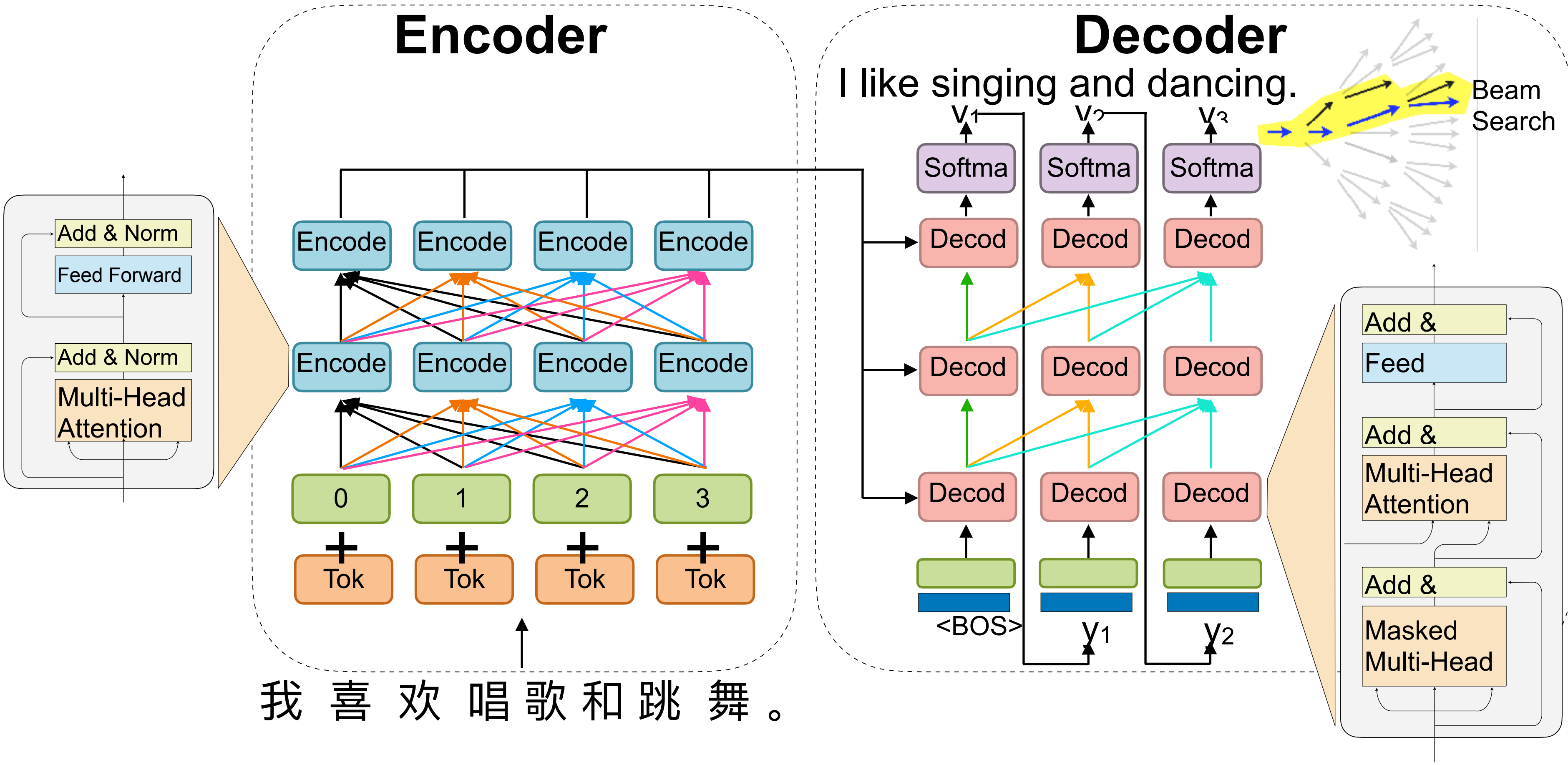
CS 190I Deep Learning Pretrained Language Models

Lei Li (leili@cs)
UCSB

Recap

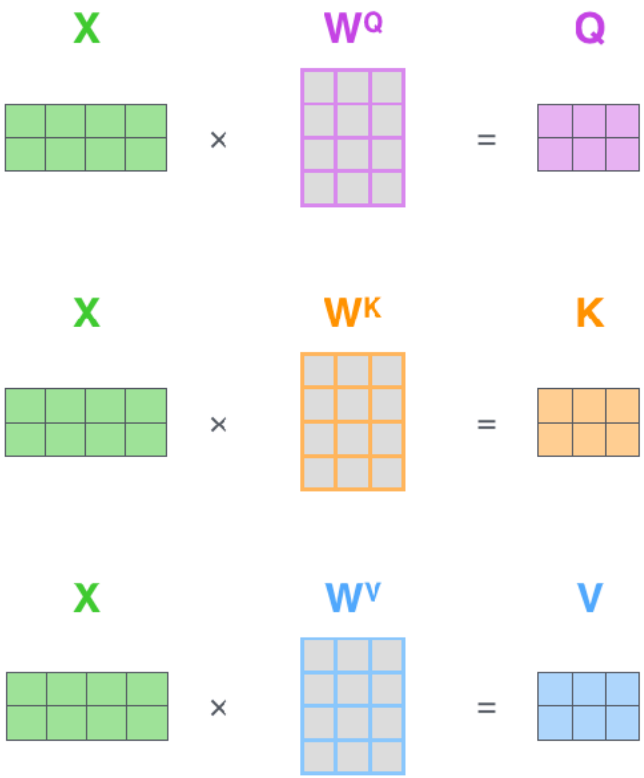
- Key components in Transformer
  - Positional Embedding (to distinguish tokens at different pos)
  - Multihead attention
  - Residual connection
  - layer norm
- Transformer is effective for machine translation, and many other tasks

Transformer

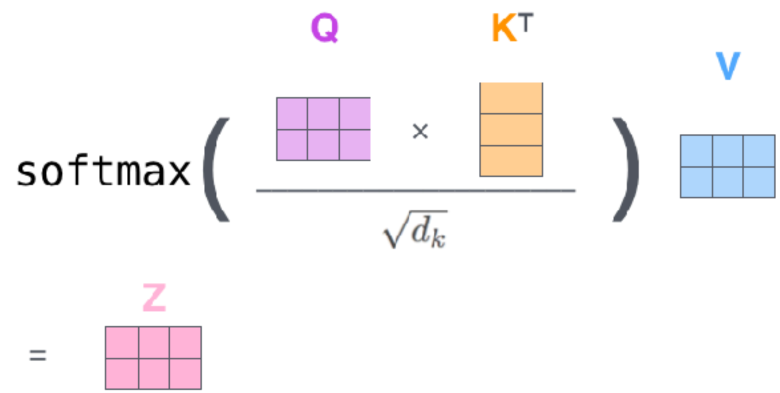


Vaswani et al. Attention is All You Need. 2017

Multi-head Attention



sent len x sent len



sent len x dim

Alammar, The Illustrated Transformer

Outline

- ELMo
- BERT
  - RoBERTa
  - Albert
- GPT

Pre-training in NLP

- Training on a large-scale general domain data before training on a particular task
  - usually raw (unlabelled) and easily available corpus
  - self-supervised: using self-contracted signals.
  - there are also cases with supervised pre-training.
- Two stages:
  - Pre-train
  - Fine-tune

Pre-training Word Embeddings

- Word embeddings are the basis of deep learning for NLP

```
king                      queen
[-0.5, -0.9, 1.4, …]      [-0.6, -0.8, -0.2, …]
```

- Word embeddings (`word2vec`, `GloVe`) are often *pre-trained* on text corpus from co-occurrence statistics

Inner Product   Inner Product

```
the king wore a crown        the queen wore a crown
```

Contextual Representations

- Problem: Word embeddings are applied in a context free manner

open a bank account   on the river bank

[0.3, 0.2, -0.8, …]

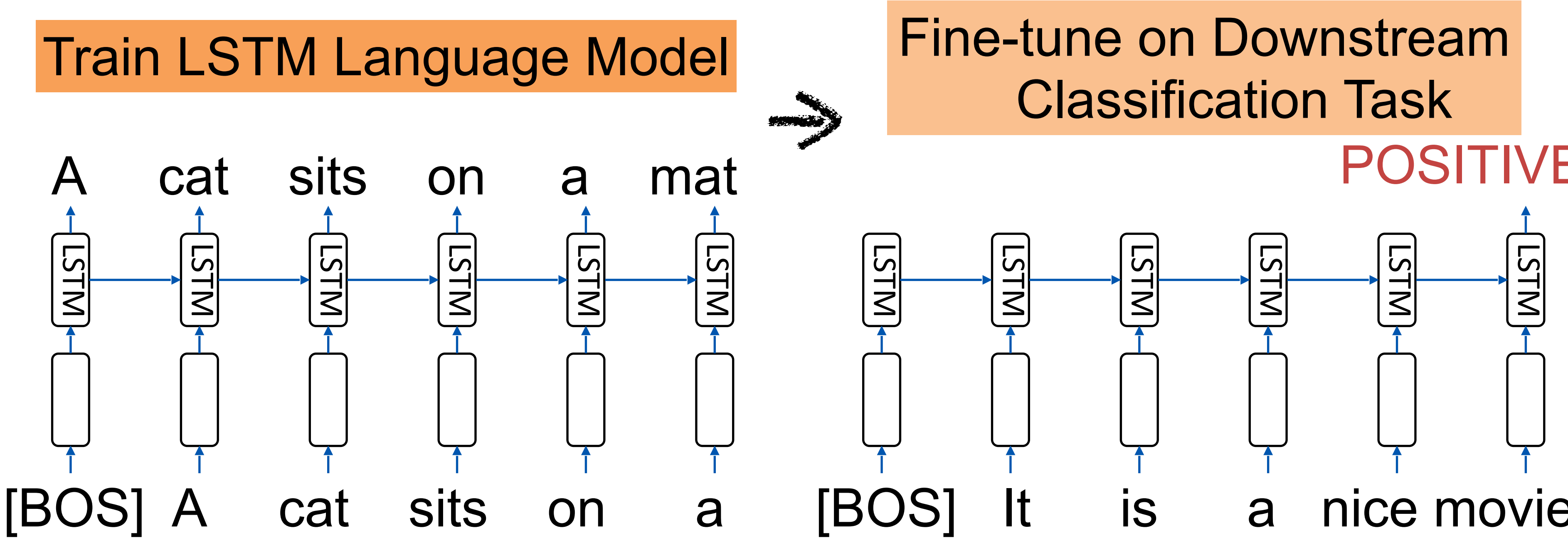
- Solution: Train contextual representations on text corpus

[0.9, -0.2, 1.6, …]   [-1.9, -0.4, 0.1, …]

open a bank account   on the river bank

Pre-train and Fine-tune on LSTM

- Sentiment analysis
  - movie review ==> positive, neutral, negative



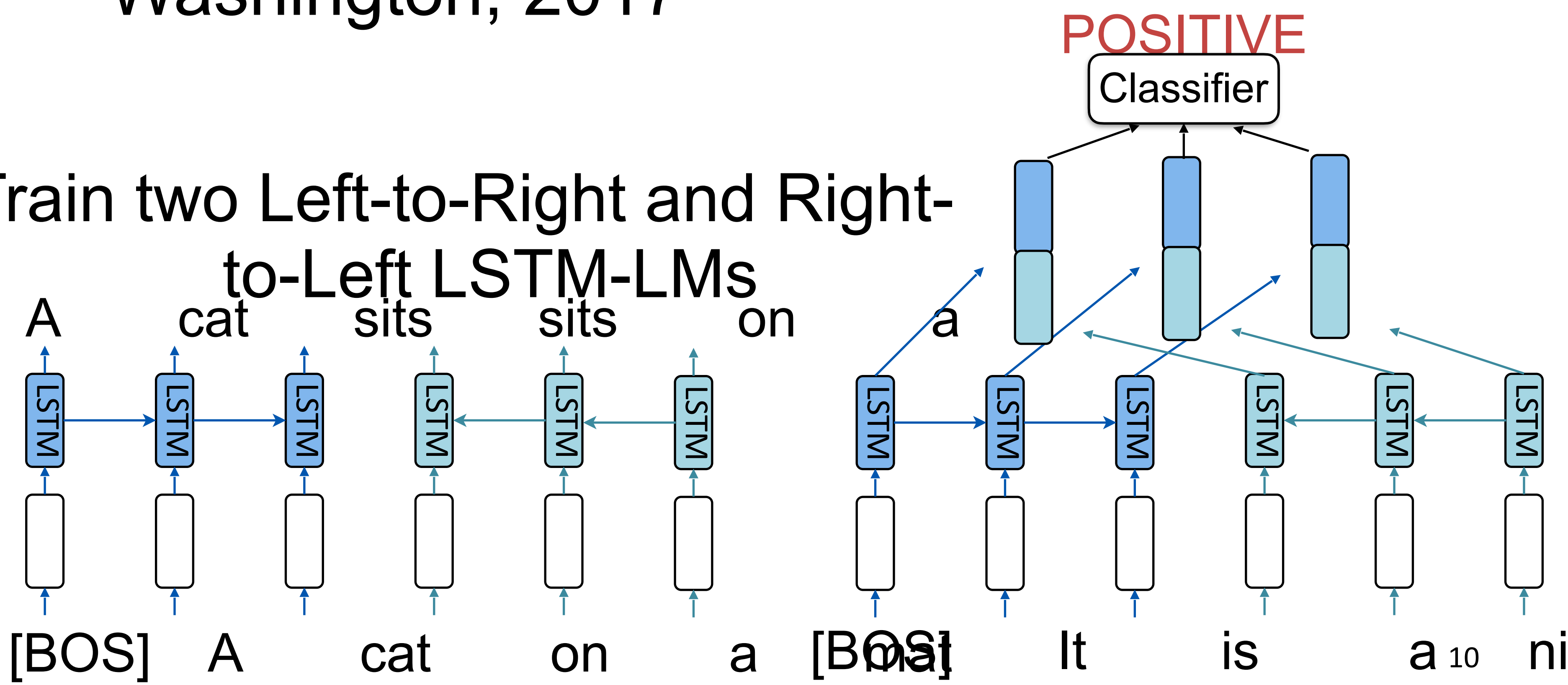
Dai & Le. Semi-Supervised Sequence Learning, 2015

ELMo

- ELMo: Deep Contextual Word Embeddings, AI2 & University of Washington, 2017

Fine-tune

Train two Left-to-Right and Right-to-Left LSTM-LMs



Peters et al. Deep Contextualized Word Representations. 2018

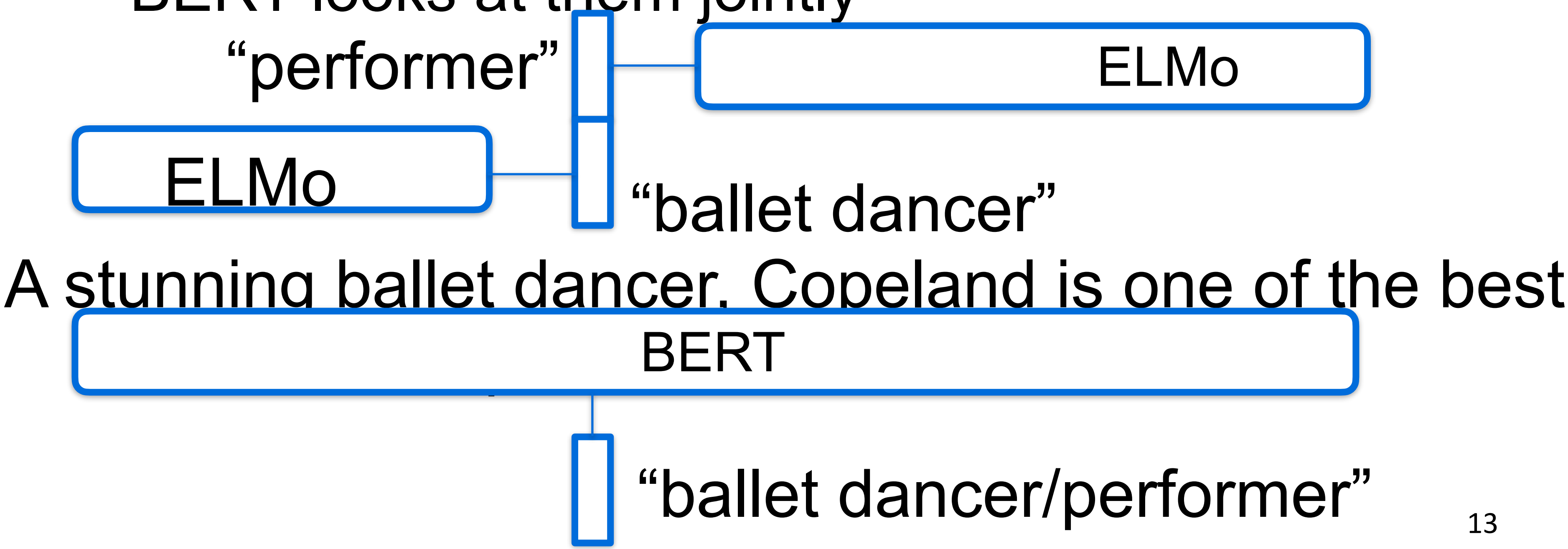
BERT

BERT

- AI2 released ELMo in spring 2018, GPT was released in summer 2018, BERT came out October 2018
- Major changes compared to ELMo:
  - Transformers instead of LSTMs (transformers in GPT as well)
  - Truly bidirectional context => Masked LM objective instead of standard LM

From Unidirectional to Bidirectional Context

- ELMo is a unidirectional model (as is GPT): we can concatenate two unidirectional models, but is this the right thing to do?
- ELMo reprs look at each direction in isolation; BERT looks at them jointly

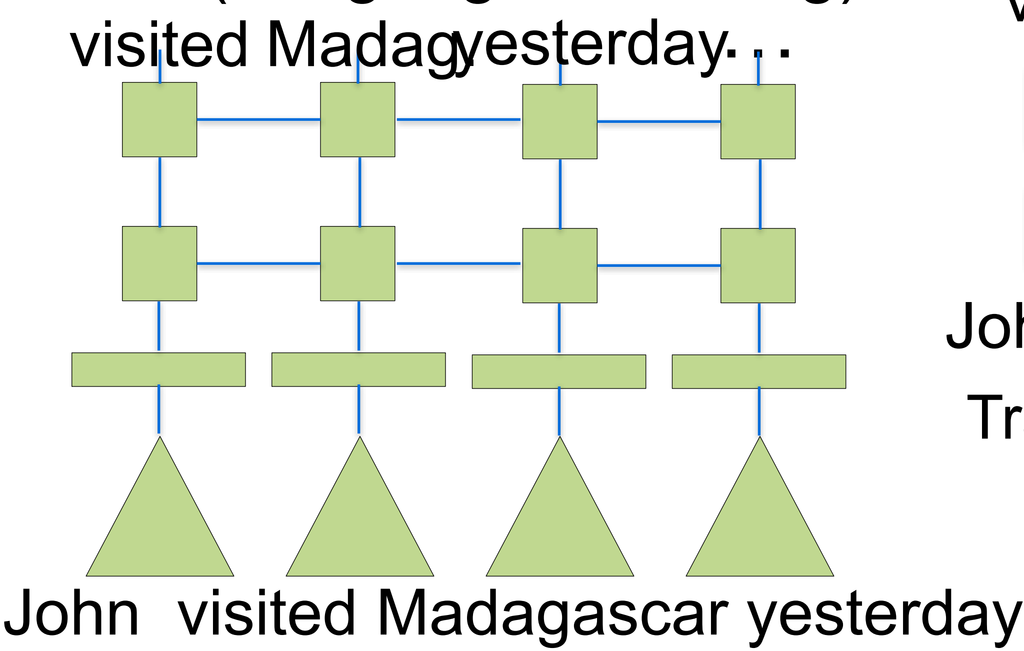


Devlin et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 2019

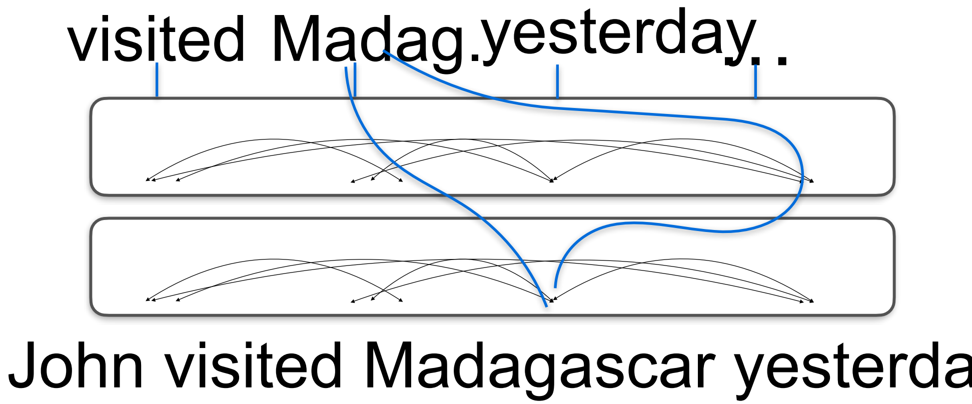
Bidirectional Context

- How to learn a "deeply bidirectional" model? What happens if we just replace an LSTM with a transformer?

ELMo (Language Modeling)



BERT

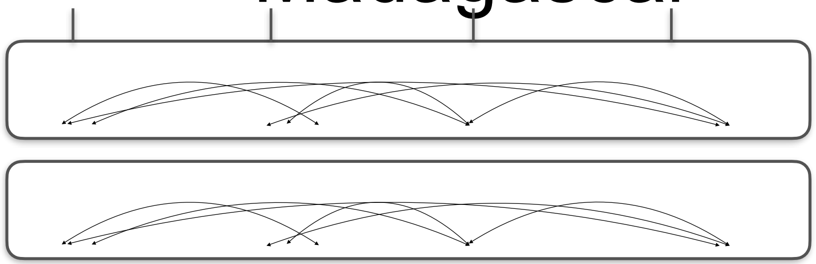


Transformer LMs have to be "one-sided" (only attend to previous tokens), not what we want

Masked Language Modeling

- How to prevent cheating? Next word prediction fundamentally doesn't work for bidirectional models, instead do masked language modeling
- BERT formula: take a chunk of text, predict 15% of the tokens
  - For 80% (of the 15%), replace the input token with [MASK]
  - For 10%, replace w/ random
  - For 10%, keep same (why?)

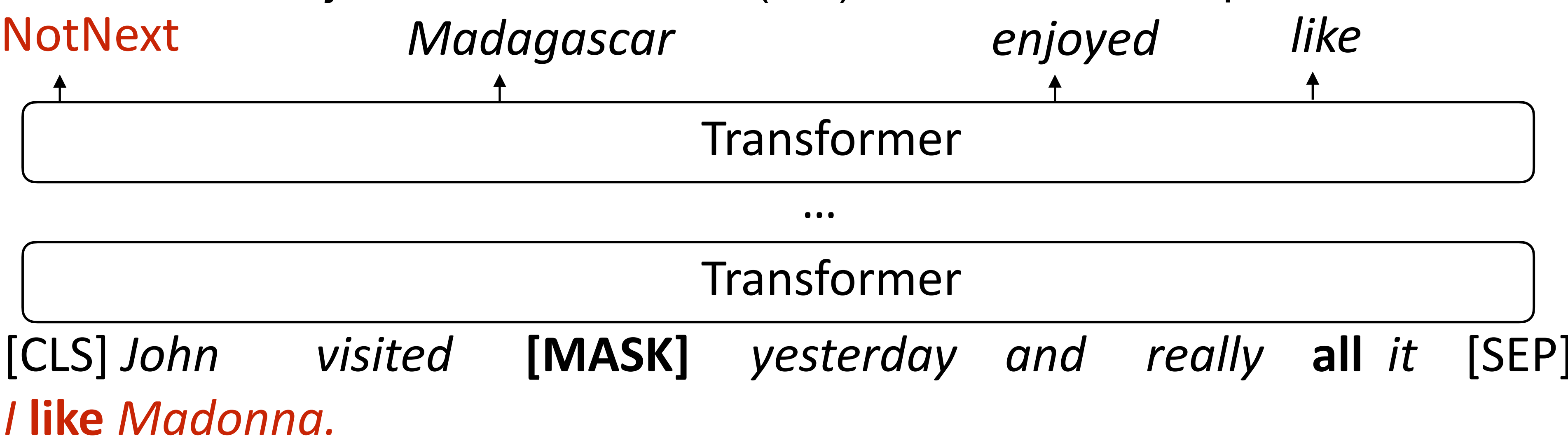
Madagascar



John visited [MASK] yesterday
John visited of yesterday
John visited Madagascar yesterday

Next "Sentence" Prediction

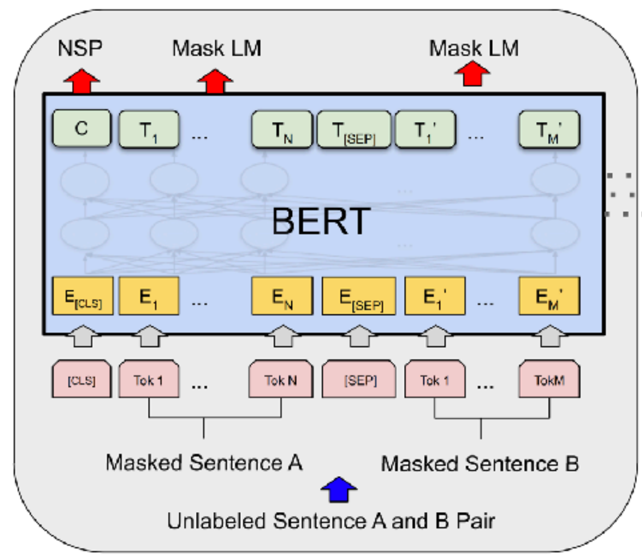
- Input: [CLS] Text chunk 1 [SEP] Text chunk 2
- 50% of the time, take the true next chunk of text, 50% of the time take a random other chunk. Predict whether the next chunk is the "true" next
- BERT objective: masked LM (CE) + next sentence prediction

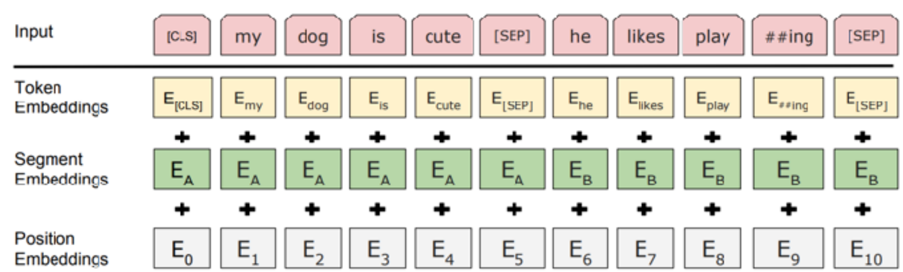


Devlin et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 2019

BERT Architecture

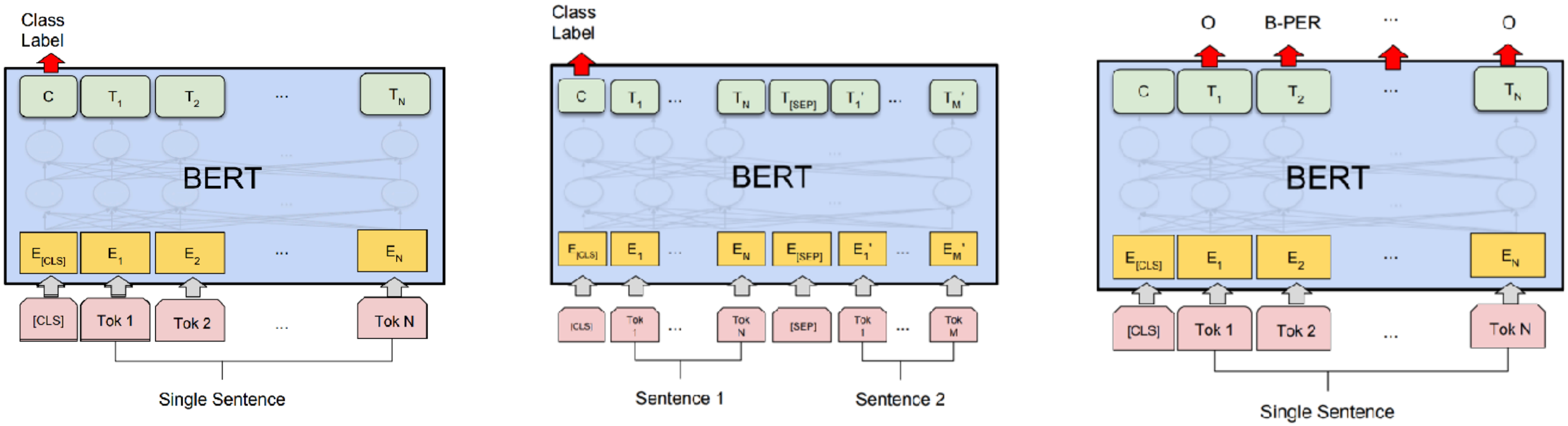
- BERT Base: 12 Transformer encoder layers, 768-dim, 12 heads. Total params = 110M
- BERT Large: 24 layers, 1024-dim, 16 heads. Total params = 340M
- Vocabulary: 30k wordpiece
- Positional embeddings and segment embeddings
- Data: Wikipedia (2.5B words + BookCorpus (800M words)





Devlin et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 2019

Unified model across NLP Tasks



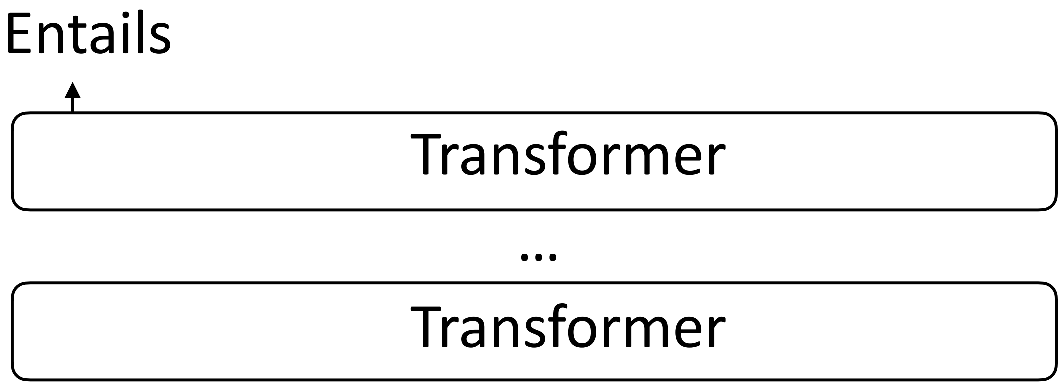
(b) Single Sentence Classification Tasks: SST-2, CoLA

(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

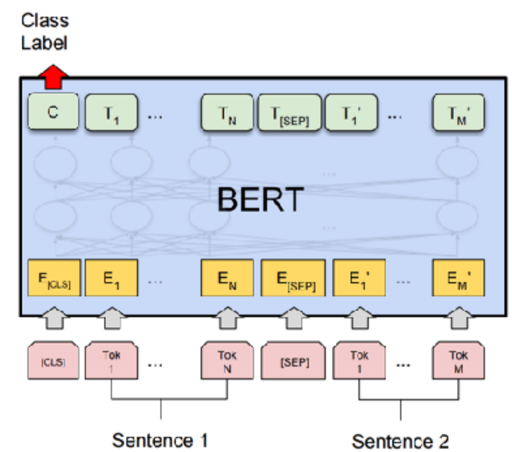
(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

- CLS token is used to provide classification decisions
- Sentence pair tasks (entailment): feed both sentences into BERT
- BERT can also do tagging by predicting tags at each word piece

What can BERT do?



[CLS] A boy plays in the snow [SEP] A boy is outside



(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

- How does BERT model this sentence pair stuff?
- Transformers can capture interactions between the two sentences, even though the NSP objective doesn't really cause this to happen

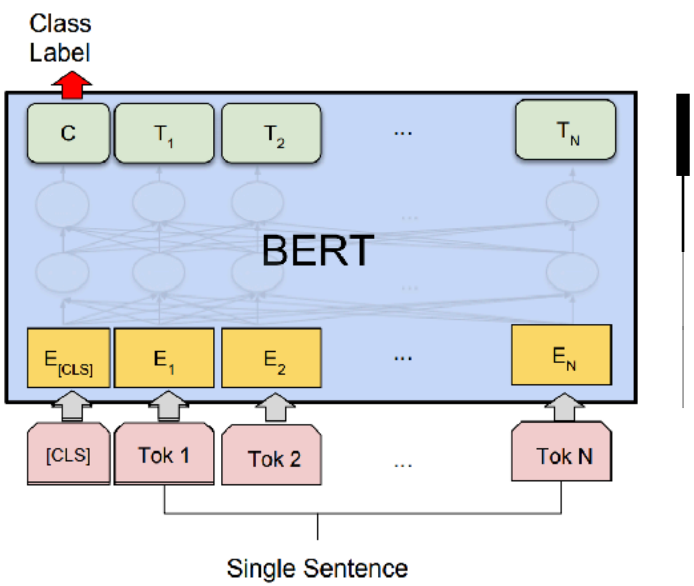
Devlin et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 2019

What can BERT NOT do?

- Does not give sentence probability
- BERT cannot generate text (at least not in an obvious way)
  - Not an autoregressive model, can do weird things like stick a [MASK] at the end of a string, fill in the mask, and repeat
- Masked language models are intended to be used primarily for "understanding/ analysis" tasks (NLU)

Fine-tuning BERT

- Fine-tune for 1-3 epochs, batch size 2-32, learning rate 2e-5 - 5e-5



(b) Single Sentence Classification Tasks: SST-2, CoLA

- Large changes to weights up here (particularly in last layer to route the right information to [CLS])
- Smaller changes to weights lower down in the transformer
- Small LR and short fine-tuning schedule mean weights don't change much
- More complex "triangular learning rate" schemes exist

Fine-tuning BERT

Pretraining	Adaptation	NER	SA	Nat. lang. inference		Semantic textual similarity		
		CoNLL 2003	SST-2	MNLI	SICK-E	SICK-R	MRPC	STS-B
Skip-thoughts	❄️	-	81.8	62.9	-	86.6	75.8	71.8
ELMo	❄️	91.7	**91.8**	**79.6**	**86.3**	**86.1**	**76.0**	**75.9**
	🔥	**91.9**	91.2	76.4	83.3	83.3	74.7	75.5
	Δ=🔥-❄️	0.2	-0.6	-3.2	-3.3	-2.8	-1.3	-0.4
BERT-base	❄️	92.2	93.0	**84.6**	84.8	86.4	78.1	82.9
	🔥	**92.4**	**93.5**	**84.6**	**85.8**	**88.7**	**84.8**	**87.1**
	Δ=🔥-❄️	0.2	0.5	0.0	1.0	2.3	6.7	4.2

- BERT is typically better if the whole network is fine-tuned, unlike ELMo

Peters, Ruder, Smith. To Tune or Not to Tune? Adapting Pretrained Representations to Diverse Tasks (2019)

Evaluation: GLUE

Corpus	|Train|	|Test|	Task	Metrics	Domain
			Single-Sentence Tasks		
CoLA	8.5k	**1k**	acceptability	Matthews corr.	misc.
SST-2	67k	1.8k	sentiment	acc.	movie reviews
			Similarity and Paraphrase Tasks		
MRPC	3.7k	1.7k	paraphrase	acc./F1	news
STS-B	7k	1.4k	sentence similarity	Pearson/Spearman corr.	misc.
QQP	364k	**391k**	paraphrase	acc./F1	social QA questions
			Inference Tasks		
MNLI	393k	**20k**	NLI	matched acc./mismatched acc.	misc.
QNLI	105k	5.4k	QA/NLI	acc.	Wikipedia
RTE	2.5k	3k	NLI	acc.	news, Wikipedia
WNLI	634	**146**	coreference/NLI	acc.	fiction books

Wang et al. GLUE. 2019

Results

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	**Average**
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.9	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	88.1	91.3	45.4	80.0	82.3	56.0	75.2
$BERT_{BASE}$	84.6/83.4	71.2	90.1	93.5	52.1	85.8	88.9	66.4	79.6
$BERT_{LARGE}$	**86.7/85.9**	**72.1**	**91.1**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**81.9**

- Huge improvements over prior work (even compared to ELMo)
- Effective at “sentence pair” tasks: textual entailment (does sentence A imply sentence B), paraphrase detection

Devlin et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 2019

Improving BERT

- Dynamic masking: standard BERT uses the same MASK scheme for every epoch, RoBERTa recomputes them

epoch 2

epoch 1

… John visited Madagascar yesterday …

- Whole word masking: don't mask out parts of words

… _John _visited _Mada gas car yesterday …

Liu et al. (2019)

RoBERTa

- "Robustly optimized BERT" incorporating some of these tricks
- 160GB of data instead of 16 GB

Model	data	bsz	steps	SQuAD (v1.1/2.0)	MNLI-m	SST-2
RoBERTa						
with BOOKS + WIKI	16GB	8K	100K	93.6/87.3	89.0	95.3
+ additional data (§3.2)	160GB	8K	100K	94.0/87.7	89.3	95.6
+ pretrain longer	160GB	8K	300K	94.4/88.7	90.0	96.1
+ pretrain even longer	160GB	8K	500K	**94.6/89.4**	**90.2**	**96.4**
$BERT_{LARGE}$						
with BOOKS + WIKI	13GB	256	1M	90.9/81.8	86.6	93.7

- New training + more data = better performance

Liu et al. (2019)

ALBERT

- Factorized embedding matrix to save parameters, model context-independent words with fewer parameters

  Ordinarily |V| x H — |V| is 30k-90k, H is >1000

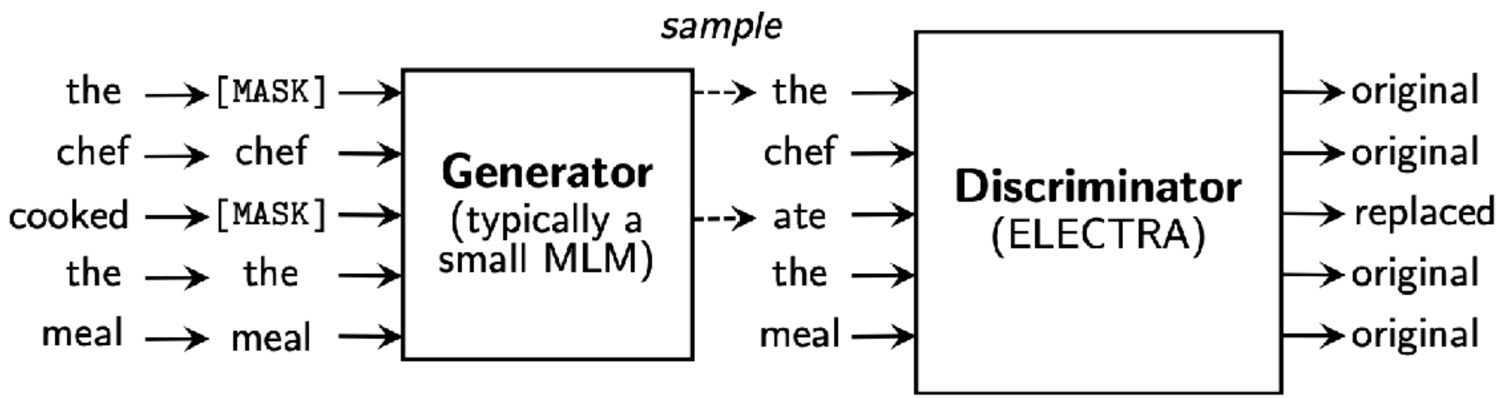
  Factor into two matrices with a low-rank approximation

  Now: |V| x E and E x H — E is 128 in their implementation

- Additional cross-layer parameter sharing

Lan et al. (2020)

ELECTRA



- No need to necessarily have a generative model (predicting words)
- This objective is more computationally efficient (trains faster) than the standard BERT objective

Clark et al. ELECTRA. 2020

BERT/MLMs

- There are lots of ways to train these models!
- Key factors:
  - Big enough model
  - Big enough data
  - Well-designed “self-supervised” objective (something like language modeling). Needs to be a hard enough problem!

Analysis/Visualization of BERT

BERTology

(1) How can we probe syntactic + semantic knowledge of BERT? What does BERT "know" in its representations?

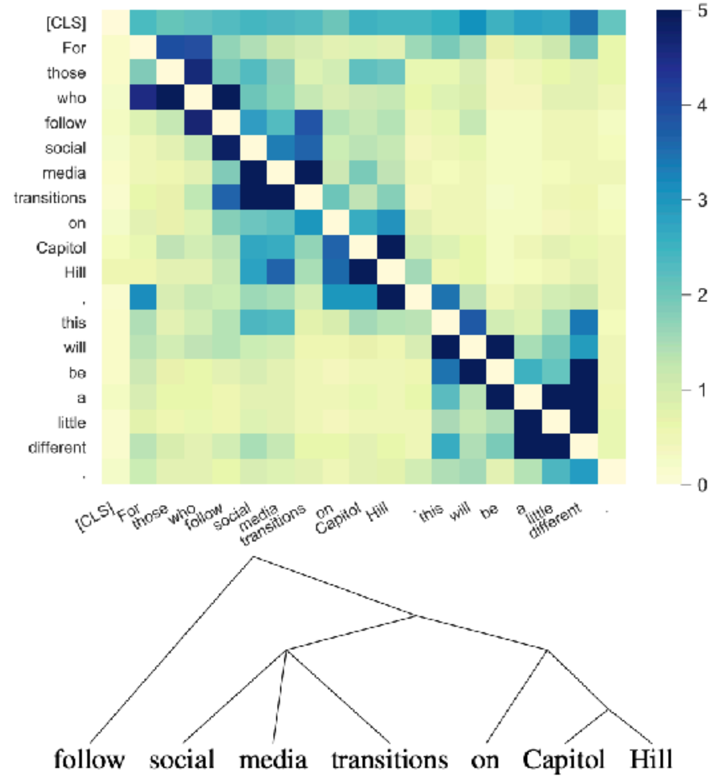
(2) What can we learn from looking at attention heads?

(3) What can we learn about training BERT (more efficiently, etc.)?

Rogers et al. (2020)

BERTology: Probing

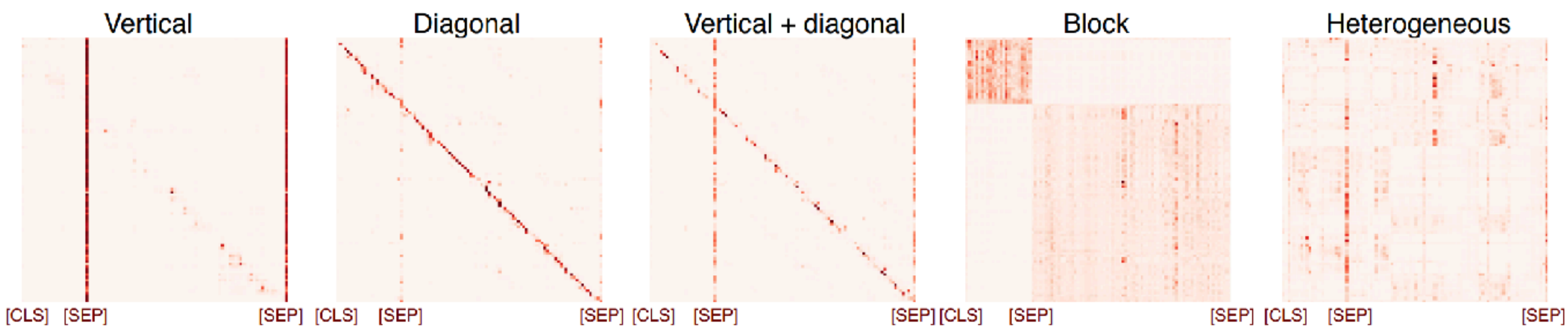
(1) In general: set up some "probing" task to try to determine syntactic features from BERT's hidden states
E.g.: Words with syntactic relations have a higher impact on one another during MLM prediction



Rogers et al. (2020

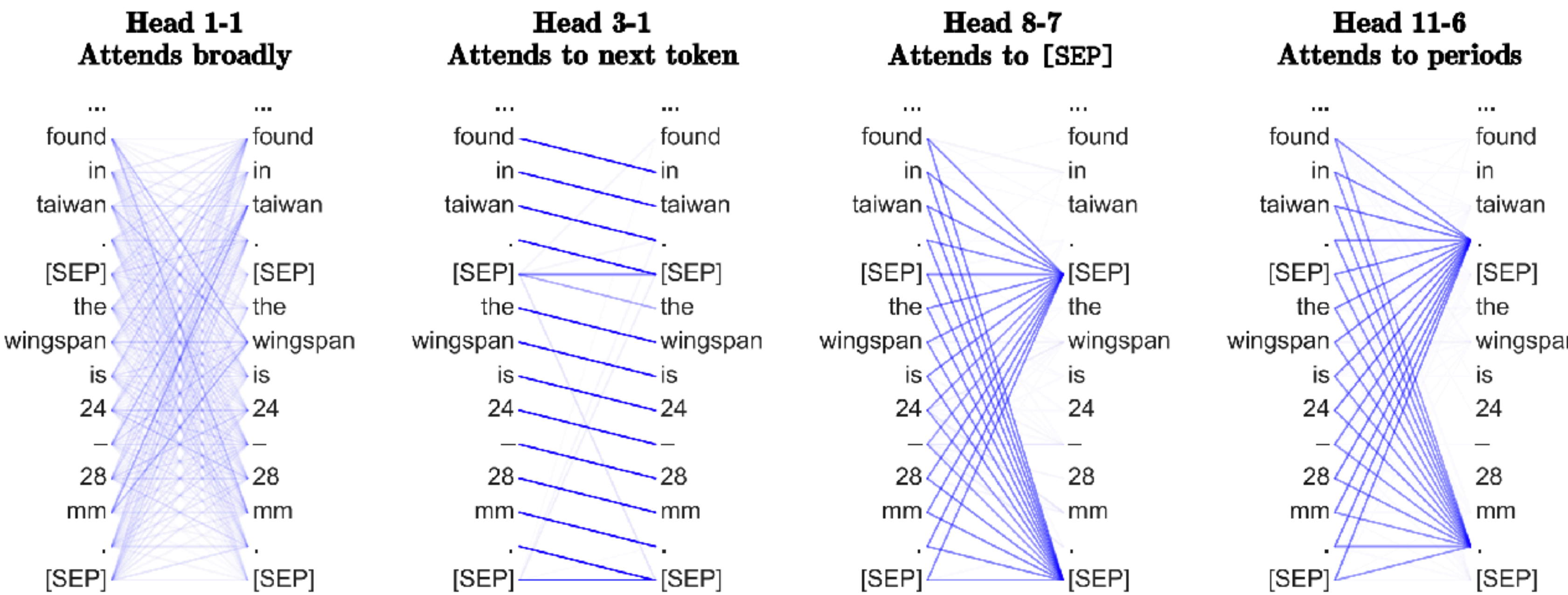
BERTology

(2) What's going inside attention heads?



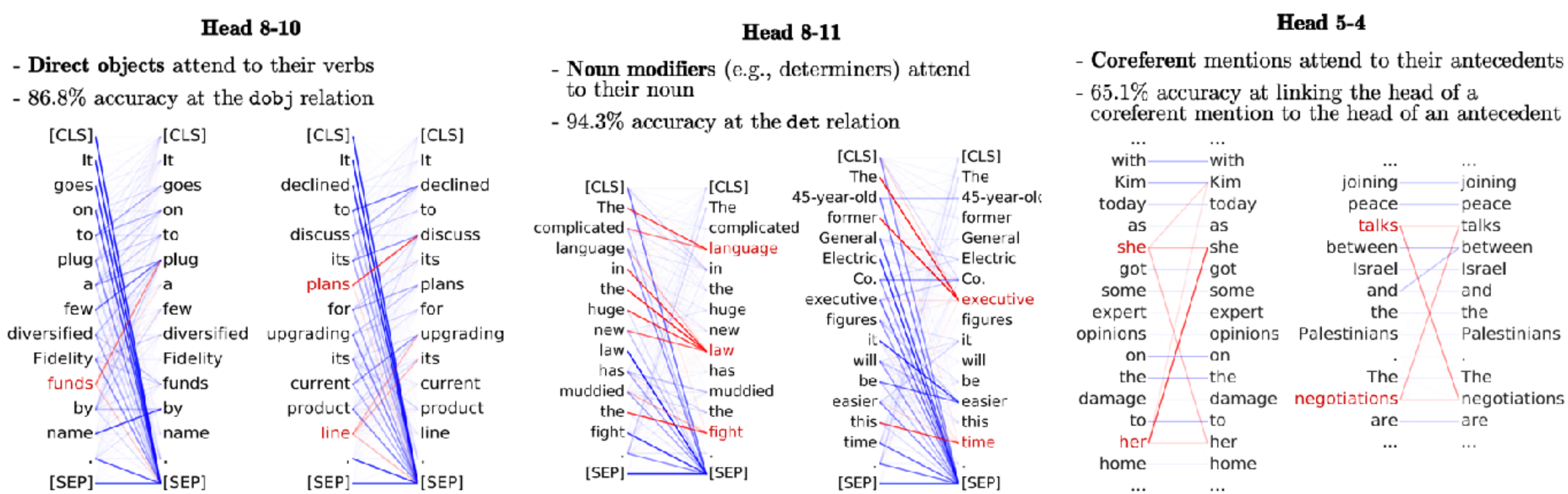
Rogers et al. (2020)

What does BERT learn?



- Heads on transformers learn interesting and diverse things: content heads (attend based on content), positional heads (based on position), etc. Clark et al. (201

What does BERT learn?

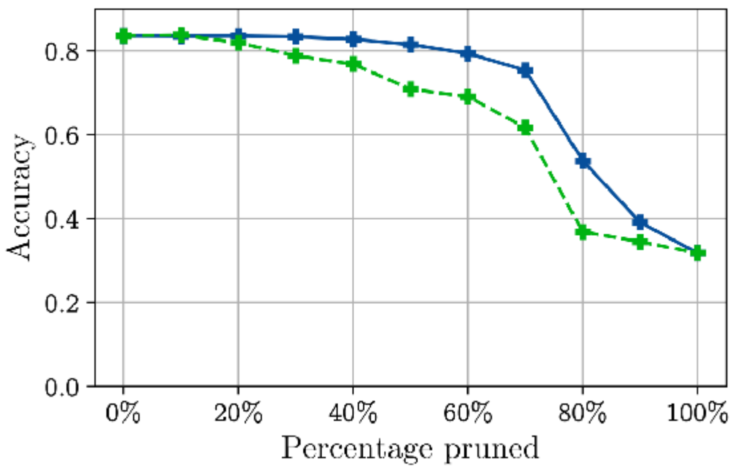


- Still way worse than what supervised systems can do, but interesting that this is learned organically

Clark et al. (20

Compressing BERT

- Remove 60+% of BERT's heads post-training with minimal drop in performance
- DistilBERT (Sanh et al., 2019): nearly as good with half the parameters of BERT (via knowledge distillation)



(b) Evolution of accuracy on the MultiNLI-matched validation set when heads are pruned from BERT according to I_h (solid blue) and accuracy difference (dashed green).

Michel et al. (20

GPT/GPT2

OpenAI GPT/GPT2

- "ELMo with transformers" (works better than ELMo)
- Train a single unidirectional transformer LM on long contexts
- GPT2: trained on 40GB of text collected from upvoted links from reddit
- 1.5B parameters — by far the largest of these models trained as of March 2019
- Because it's a language model, we can **generate** from it

Parameters	Layers	d_{model}
117M	12	768
345M	24	1024
762M	36	1280
1542M	48	1600

Radford et al. (2019)

OpenAI GPT2

SYSTEM PROMPT (HUMAN-WRITTEN)

Miley Cyrus was caught shoplifting from Abercrombie and Fitch on Hollywood Boulevard today.

MODEL COMPLETION (MACHINE-WRITTEN, SECOND TRY)

The 19-year-old singer was caught on camera being escorted out of the store by security guards.

The singer was wearing a black hoodie with the label ‘Blurred Lines’ on the front and ‘Fashion Police’ on the back.

Scroll down for video

Shoplifting: Miley Cyrus was caught shoplifting from Abercrombie and Fitch on Hollywood Boulevard today (pictured)

The singer was wearing a black hoodie with the label ‘Blurred Lines’ on the front and ‘Fashion Police’ on the back

The singer was also wearing a pair of black-rimmed glasses, a black jacket, black jeans and black sandals.

She was carrying a pair of black and white striped gloves and a small black bag.

slide credit: OpenAI

Open Questions

1) How novel is the stuff being generated? (Is it just doing nearest neighbors on a large corpus?)

2) How do we understand and distill what is learned in this model?

3) How do we harness these priors for conditional generation tasks (summarization, generate a report of a basketball game, etc.)

4) Is this technology dangerous?

GPT-3

- GPT-2 but even larger: 1.5B -> 175B parameter models

Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	6.0×10^{-4}
GPT-3 Medium	350M	24	1024	16	64	0.5M	3.0×10^{-4}
GPT-3 Large	760M	24	1536	16	96	0.5M	2.5×10^{-4}
GPT-3 XL	1.3B	24	2048	24	128	1M	2.0×10^{-4}
GPT-3 2.7B	2.7B	32	2560	32	80	1M	1.6×10^{-4}
GPT-3 6.7B	6.7B	32	4096	32	128	2M	1.2×10^{-4}
GPT-3 13B	13.0B	40	5140	40	128	2M	1.0×10^{-4}
GPT-3 175B or "GPT-3"	175.0B	96	12288	96	128	3.2M	0.6×10^{-4}

- Trained on 570GB of Common Crawl
- 175B parameter model's parameters alone take >400GB to store (4 bytes per param). Trained in parallel on a "high bandwidth cluster provided by Microsoft"

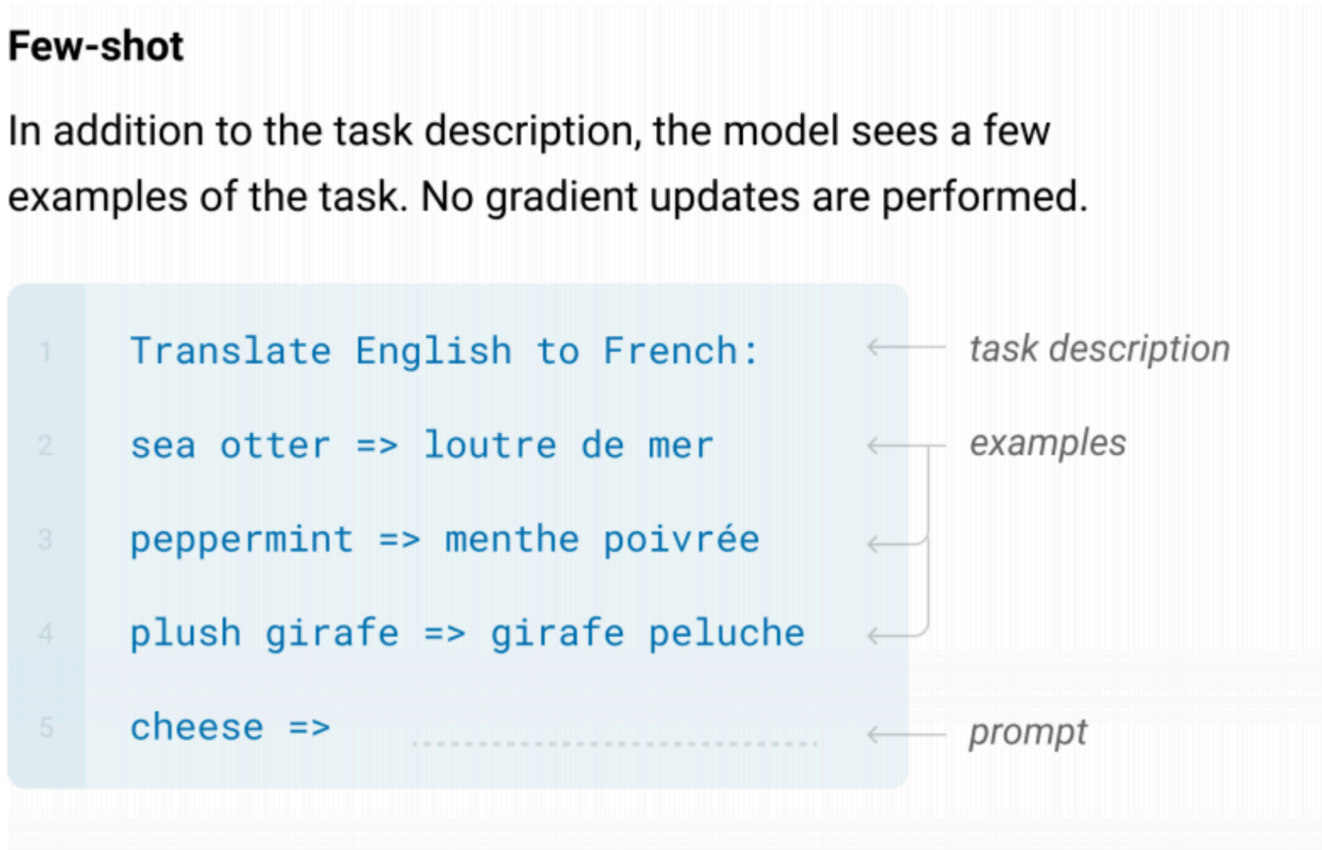
Brown et al. (2020)

GPT-3

- This is the "normal way" of doing learning in models like GPT-2

Brown et al. (2020)

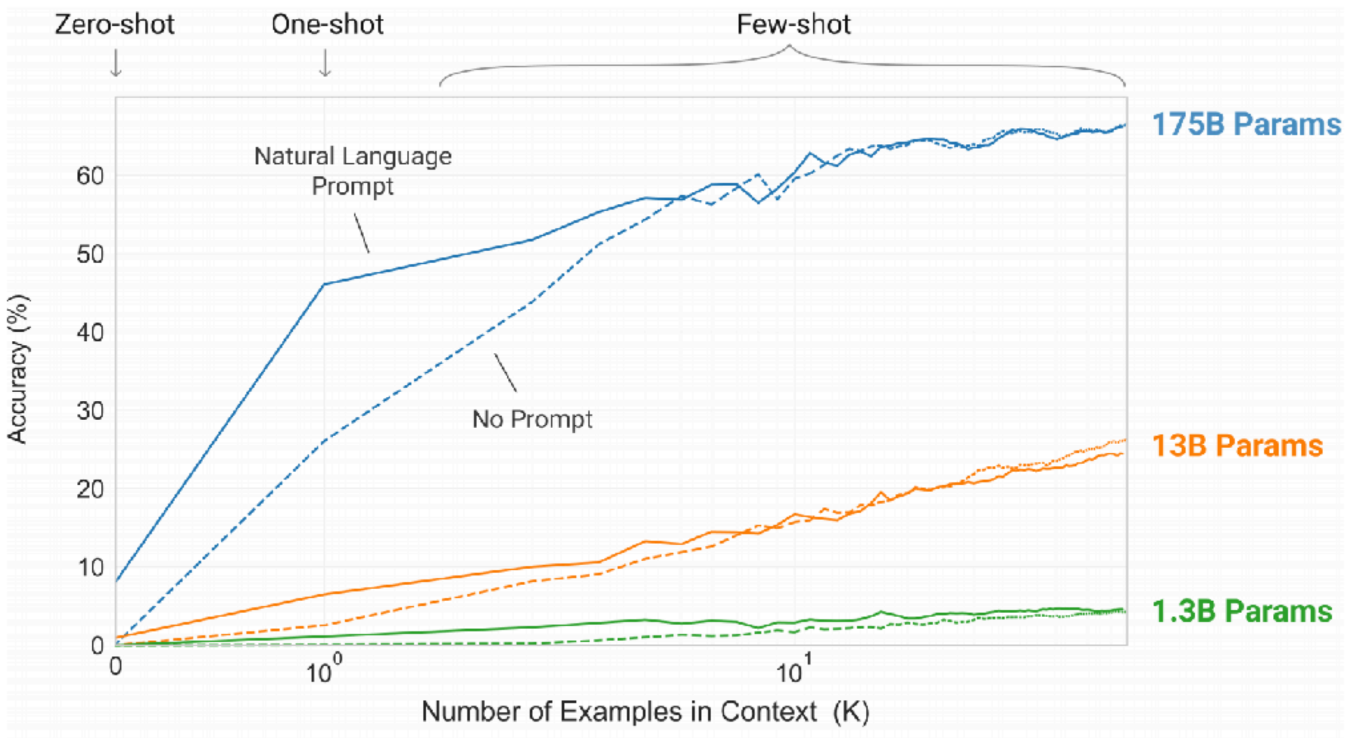
GPT-3: Few-shot Learning



Brown et a
(2020)

GPT-3

- **Key observation:** few-shot learning only works with the very largest models!



Brown et al. (2020)

GPT-3

	SuperGLUE Average	BoolQ Accuracy	CB Accuracy	CB F1	COPA Accuracy	RTE Accuracy
Fine-tuned SOTA	**89.0**	**91.0**	**96.9**	**93.9**	**94.8**	**92.5**
Fine-tuned BERT-Large	69.0	77.4	83.6	75.7	70.6	71.7
GPT-3 Few-Shot	71.8	76.4	75.6	52.0	92.0	69.0

	WiC Accuracy	WSC Accuracy	MultiRC Accuracy	MultiRC F1a	ReCoRD Accuracy	ReCoRD F1
Fine-tuned SOTA	**76.1**	**93.8**	**62.3**	**88.2**	**92.5**	**93.3**
Fine-tuned BERT-Large	69.6	64.6	24.1	70.0	71.3	72.0
GPT-3 Few-Shot	49.4	80.1	30.5	75.4	90.2	91.1

- Sometimes very impressive, (MultiRC, ReCoRD), sometimes very bad
- Results on other datasets are equally mixed — but still strong for a few-shot model!

Brown et al. (2020)

Prompt Engineering

Yelp For the Yelp Reviews Full Star dataset (Zhang et al., 2015), the task is to estimate the rating that a customer gave to a restaurant on a 1- to 5-star scale based on their review's text. We define the following patterns for an input text a:

$P_1(a) =$ It was ____. a $P_2(a) =$ Just ____! $\|$ a

$P_3(a) =$ a. All in all, it was ____.

$P_4(a) =$ a $\|$ In summary, the restaurant is ____.

We define a single verbalizer v for all patterns as

$v(1) =$ terrible $v(2) =$ bad $v(3) =$ okay
$v(4) =$ good $v(5) =$ great

"verbalizer" of labels

patterns

Fine-tune LMs on initial small dataset (note: uses smaller LMs than GPT-3)

Repeat:

- Use these models to "vote" on labels for unlabeled data
- Retrain each prompt model on this dataset

Schick and Schutze et al. (2020)

Next Up

- Graph Neural Networks