

then

Prediction Objective

BERT is trained to predict the original word based on both left and right context.

Unlike traditional language models (ex: GPT), which predict tokens only based on past context, BERT can look both ways.

Let's look into what that means

BERT vs GPT

Bo Kizildag (bk2838)

Manush Kalwari (mmk2266)

starting off...

first, we will be talking a bit about BERT, and what problem is being solved under today's topic of discussion, and why that is important.

then, we will go into the details of GPT – there will be a whole lot of discussions here about benchmarking!

finally we will discuss the open relevant problems, operational details, and the state of the art if we have more time.

BERT

“Pre-training of Deep Bidirectional Transformers for Language Understanding”



Today's key word is... *Bidirectionality*

Unlike previous models (GPT), which process text in a left-to-right or right-to-left manner, BERT learns from both past and future contexts in a sentence simultaneously.

This is achieved via a Masked Language Model (MLM), where words in a sentence are randomly masked, and the model learns to predict them based on surrounding *context*.

So what?

















Next Sentence Prediction (NSP): To improve understanding of sentence relationships, BERT allows learning to predict whether a given sentence follows another in a coherent context.

Transfer Learning at Scale: Instead of training separate models for each NLP task, BERT's pre-trained representation can be fine-tuned for various tasks (our homework 1!) with minimal task-specific adjustments.

It's really good!

It boosts GLUE scores from 72.8 (OpenAI GPT) to 80.5, a 7.7% absolute improvement – a good leap in performance.

SQuAD v1.1 question-answering test F1 score reaches 93.2, *outperforming human baselines in some cases.*

13	LG AI Research	ANNA		89.8	68.7	97.0	92.7/90.1	93.0/92.8	75.3/90.5	91.8	91.6	96.0	91.8	95.1
14	Zihang Dai	Funnel-Transformer (Ensemble B10-10-10H1024)		89.7	70.5	97.5	93.4/91.2	92.6/92.3	75.4/90.7	91.4	91.1	95.8	90.0	94.1
15	ELECTRA Team	ELECTRA-Large + Standard Tricks		89.4	71.7	97.1	93.1/90.7	92.9/92.5	75.6/90.8	91.3	90.8	95.8	89.8	91.1
16	David Kim	2digit LAnet		89.3	71.8	97.3	92.4/89.6	93.0/92.7	75.5/90.5	91.8	91.6	96.4	91.1	88.4
17	倪仕文	DropAttack-RoBERTa-large		88.8	70.3	96.7	92.6/90.1	92.1/91.8	75.1/90.5	91.1	90.9	95.3	89.9	89.1
18	Microsoft D365 AI & UMD	FreeLB-RoBERTa (ensemble)		88.4	68.0	96.8	93.1/90.8	92.3/92.1	74.8/90.3	91.1	90.7	95.6	88.7	89.0
19	Junjie Yang	HIRE-RoBERTa		88.3	68.6	97.1	93.0/90.7	92.4/92.0	74.3/90.2	90.7	90.4	95.5	87.9	89.0
20	Shiwen Ni	ELECTRA-large-M (bert4keras)		88.3	69.3	95.8	92.2/89.6	91.2/91.1	75.1/90.5	91.1	90.9	93.8	87.9	91.1
21	Facebook AI	RoBERTa		88.1	67.8	96.7	92.3/89.8	92.2/91.9	74.3/90.2	90.8	90.2	95.4	88.2	89.0
22	Microsoft D365 AI & MSR AI	MT-DNN-ensemble		87.6	68.4	96.5	92.7/90.3	91.1/90.7	73.7/89.9	87.9	87.4	96.0	86.3	89.0
23	GLUE Human Baselines	GLUE Human Baselines		87.1	66.4	97.8	86.3/80.8	92.7/92.6	59.5/80.4	92.0	92.8	91.2	93.6	95.1
24	kk xx	ELECTRA-Large-NewSCL(single)		85.6	73.3	97.2	92.7/90.2	92.0/91.7	75.3/90.6	90.8	90.3	95.6	86.9	60.1
25	Adrian de Wynter	Bort (Alexa AI)		83.6	63.9	96.2	94.1/92.3	89.2/88.3	66.0/85.9	88.1	87.8	92.3	82.7	71.1
26	Lab LV	ConvBERT base		83.2	67.8	95.7	91.4/88.3	90.4/89.7	73.0/90.0	88.3	87.4	93.2	77.9	65.1
27	Stanford Hazy Research	Snorkel MeTaL		83.2	63.8	96.2	91.5/88.5	90.1/89.7	73.1/89.9	87.6	87.2	93.9	80.9	65.1
28	XLM Systems	XLM (English only)		83.1	62.9	95.6	90.7/87.1	88.8/88.2	73.2/89.8	89.1	88.5	94.0	76.0	71.1
29	WATCH ME	ConvBERT-base-paddle-v1.1		83.1	66.3	95.4	91.6/88.6	90.0/89.2	73.9/90.0	88.2	87.7	93.3	78.2	65.1
30	Zhuosheng Zhang	SemBERT		82.9	62.3	94.6	91.2/88.3	87.8/86.7	72.8/89.8	87.6	86.3	94.6	84.5	65.1
31	Jun Yu	mpnet-base-paddle		82.9	60.5	95.9	91.6/88.9	90.8/90.3	72.5/89.7	87.6	86.6	93.3	82.4	65.1
32	Danqi Chen	SpanBERT (single-task training)		82.8	64.3	94.8	90.9/87.9	89.9/89.1	71.9/89.5	88.1	87.7	94.3	79.0	65.1
33	GAL team	distilRoBERTa+GAL (6-layer transformer single model)		82.6	60.0	95.3	91.9/89.2	90.0/89.6	73.3/90.0	87.4	86.5	92.7	81.8	65.1

Click on a submission to see more information

More on *MLM*

Fun fact: MLM is inspired by the **Cloze Task**, where certain words in a sentence are hidden, and the model is trained to predict them.

Put the right word in each sentence

dogs cats ~~pencils~~ car house pizzas

👉 There are three pencils on my table

1 My _____ is pretty fast

2 My _____ barks a lot but welcomes thieves
equipped with sausages

3 I love _____ because they say
"meeeeooow"

4 I'm able to eat 4 _____ at any time

5 My _____ has 3 swimming pools and a
bowling lane, are you envious ?

Submit

How it works

Randomly Masking Words

In each training, 15% of the words in the input are *randomly* selected for prediction.

Of these selected words:

- 80% are replaced with the [MASK] token ("The cat sat on the [MASK]")

- 10% are replaced with a random word (robustness to noise)

- 10% remain unchanged (for normal contextual representation).

example:

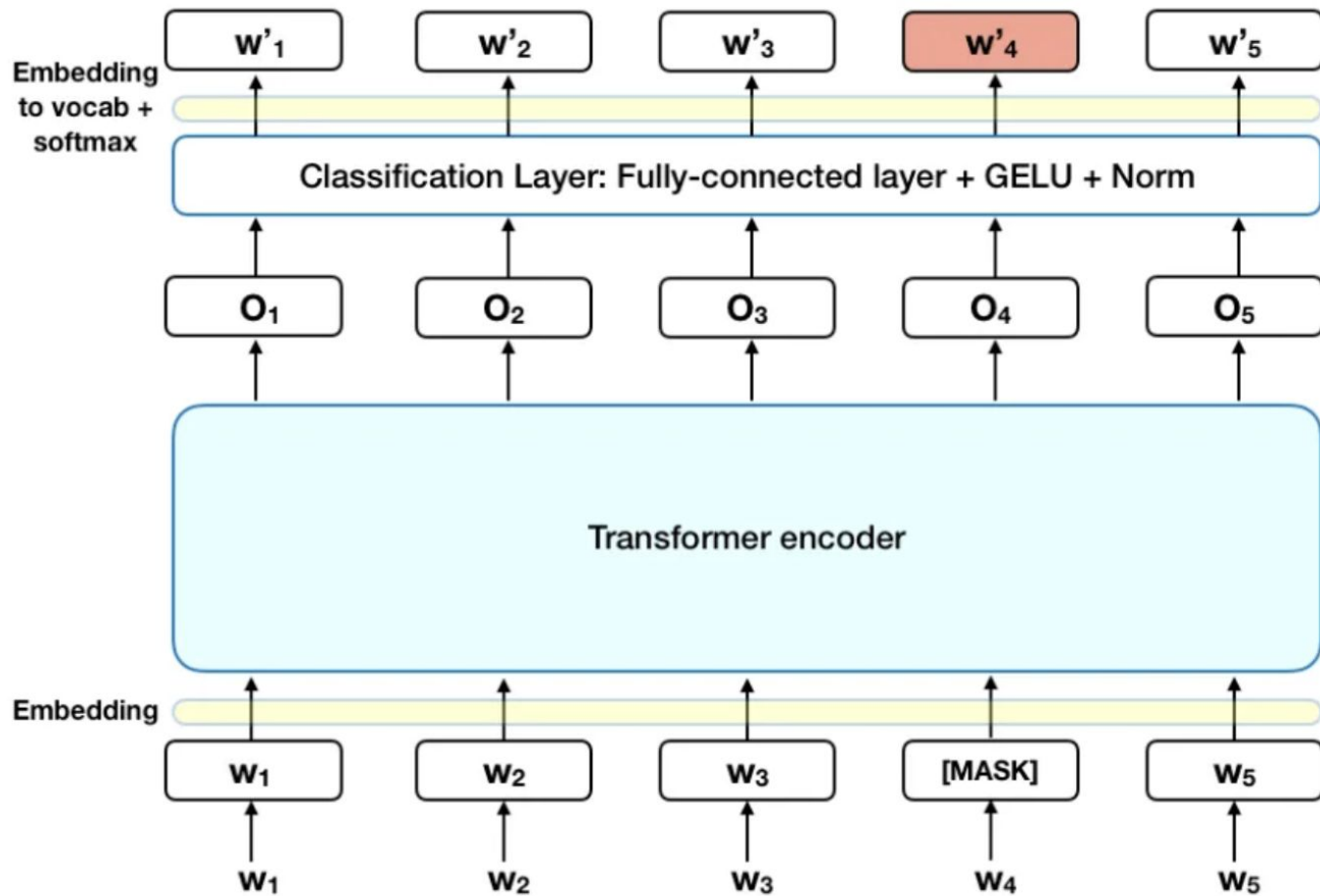
Input Sentence:

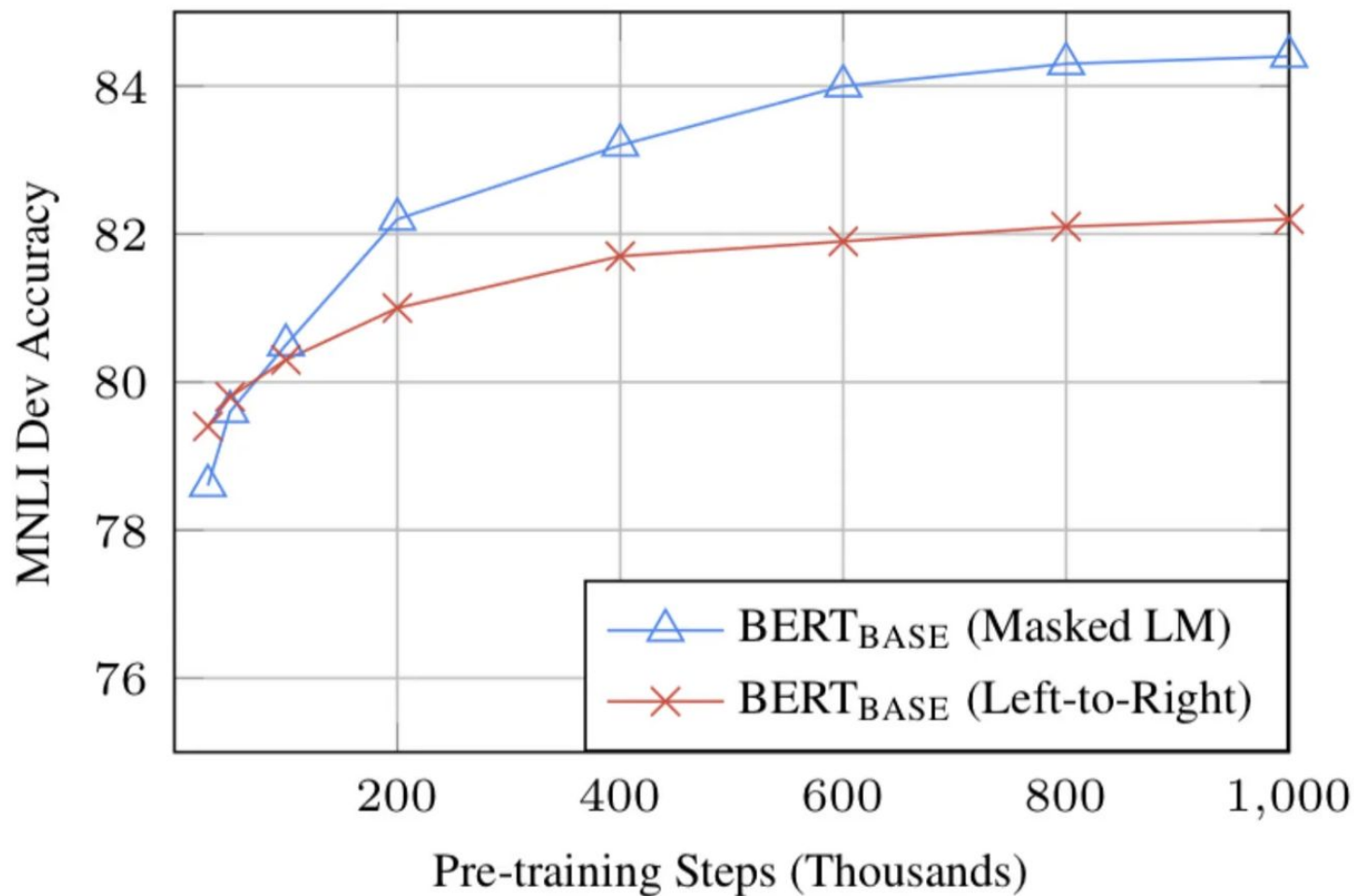
"The [MASK] barked at the mailman."

A left-to-right model can only rely on "The" to guess the missing word.

A right-to-left model can only use "barked at the mailman."

BERT (MLM) can use the entire sentence, leading to a better understanding that "dog" is a likely candidate.





Can't forget *NSP*

MLM helps BERT understand word-level context, and NSP enables it to *learn relationships between sentences*

But how?

Sentence Pair Formation

During pre-training, each training example consists of two sentences:

=> 50% of the time, the second sentence logically follows the first (labeled as "IsNext").

=> 50% of the time, the second sentence is randomly selected from the corpus (labeled as "NotNext").

Prediction Objective: To predict whether sentence B actually follows sentence A.

example:

Sentence A: *"The cat sat on the mat."*

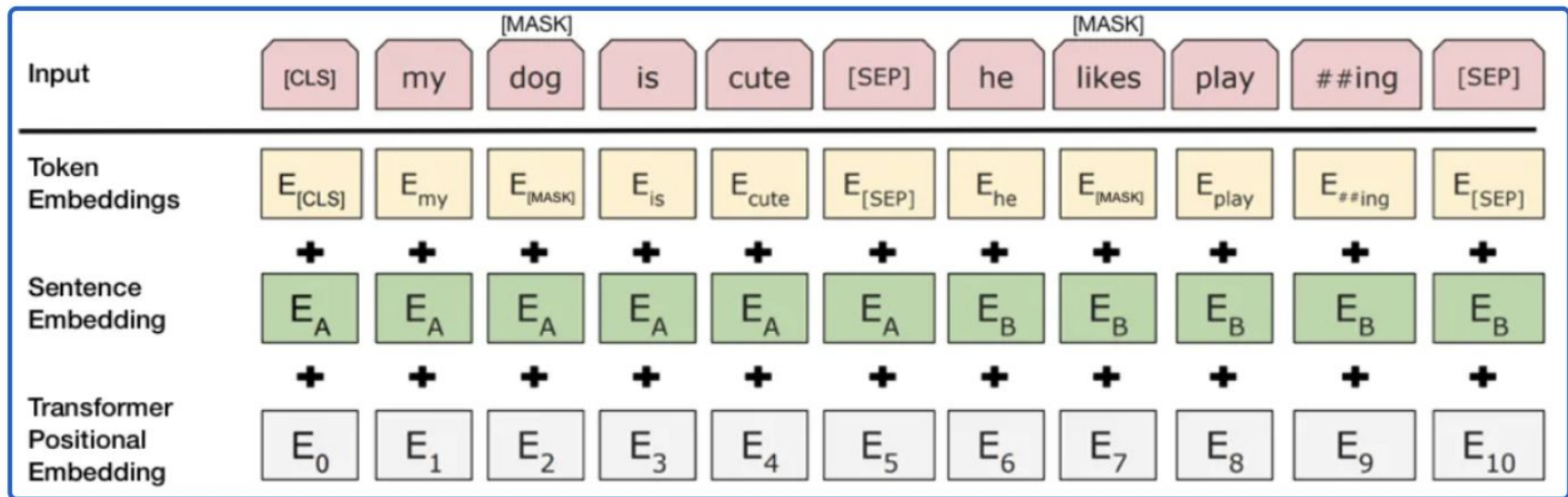
Sentence B: *"It purred softly as it curled up to sleep."*

Label: **IsNext** (Sentence B logically follows Sentence A)

Why?

Sentence B provides a natural continuation of Sentence A.

The reference to "**it** purred softly" logically relates to "**The cat** sat on the mat."



Source: BERT [Devlin et al., 2018], with modifications

They are essentially complementary

MLM teaches BERT to deeply *understand word-level context*.

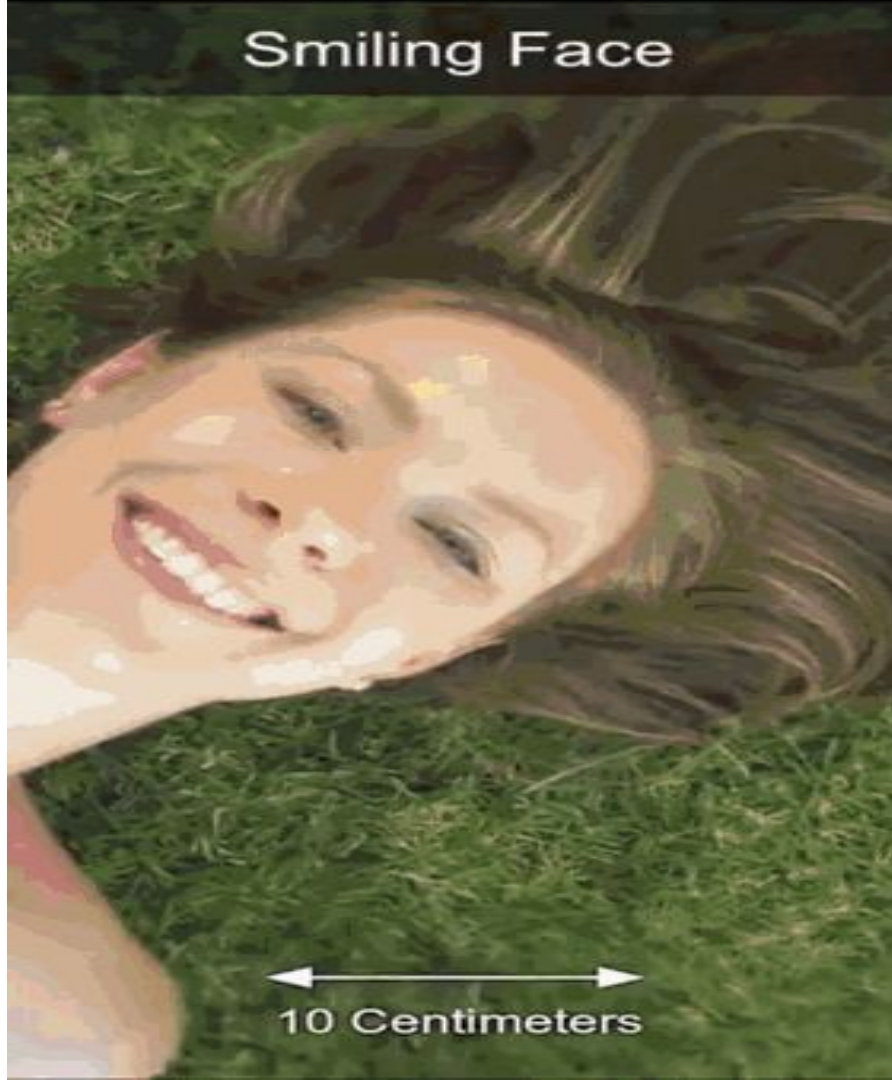
NSP teaches BERT to recognize sentence relationships. Enabling to *capture long-range dependencies between text pairs* (important for tasks like entailment and question answering).

Together, they allow BERT to perform well across a variety of NLP tasks *without task-specific architecture changes*.



■ However, later research (RoBERTa) found that NSP was not as critical as MLM, and removing it could even improve performance on some tasks. (Instead, RoBERTa used longer training times and more data to achieve better results).

Zooming out



Architectural takeaways

Model size matters, even at huge scale. BERT_large, with 345 million parameters, was the largest model of its kind then. It was demonstrably superior on small-scale tasks to BERT_base, which uses the same architecture with “only” 110 million parameters.

With enough training data, more training steps == higher accuracy

BERT's MLM approach converges slower than left-to-right approaches (as only 15% of words are predicted in each batch) but as mentioned bidirectional training still outperforms left-to-right training after a small number of pre-training steps.

Issues?

cost!

A. Pre-training is Extremely Expensive

Training BERTLARGE required 64 TPU chips for 4 days! This makes it infeasible for most researchers and companies to train BERT from scratch. Energy consumption is massive too.

B. Inference is Computationally Expensive

The model size is huge. Performing inference requires significant memory and compute resources.

Latency issues arise when deploying BERT for real-time applications (chatbots, search engines).

Implication:

Smaller, more efficient alternatives are needed (DistilBERT, ALBERT, TinyBERT)

tuning issues...

During fine-tuning the [MASK] token is never used (since real-world tasks don't have masked words).

=> This creates a discrepancy between training and real-world usage.

Solution in the Paper is Not Ideal: BERT tries to mitigate this by:

Using [MASK] only 80% of the time, replacing it with a random word 10% of the time, and leaving the original word unchanged 10% of the time.

However, this does not completely eliminate the mismatch.

So: This pre-training/fine-tuning gap reduces efficiency in real-world applications.

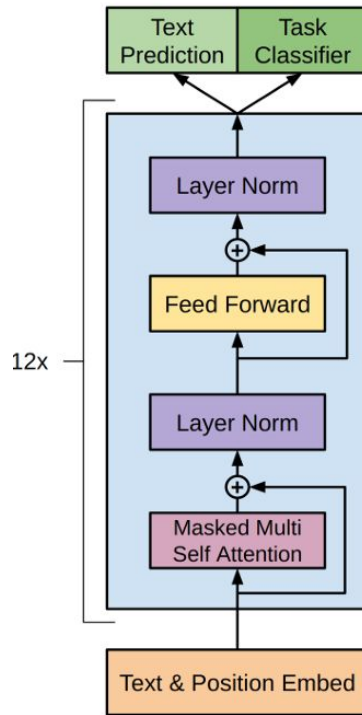
(Later models like RoBERTa remove the NSP task and use continuous token masking to address this).

(BO END)

GPT - 1,2,3

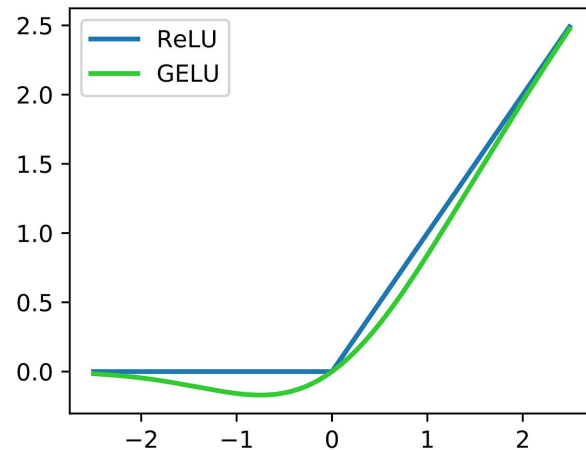
GPT-1

- General-purpose LLM.
- Can be fine-tuned for specific tasks with minimal task-specific architecture changes.
- Unsupervised pre-training followed by supervised fine-tuning.
- Transformer specification-
 - Decoder-only
 - multi-headed
 - residual connections
 - No cross-attention
 - Layer Norm after the residual
 - GELU instead of RELU



GELU (Gaussian Error Linear Unit)

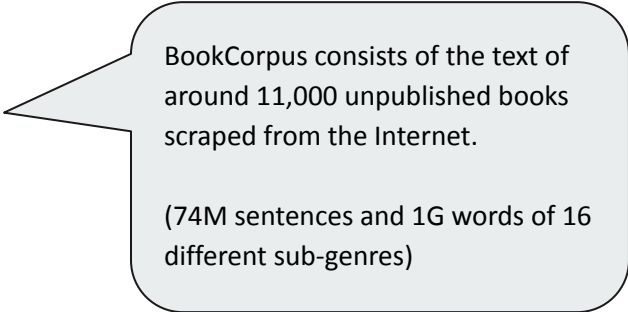
- ReLU has hard cutoff at 0, causing optimization issues in very deep networks. GELU is smoother and differentiable for all values.
- Avoids dying ReLU issue.
- Empirically has better convergence speed and final accuracy, despite the slightly higher computational cost.
- Probabilistic interpretation- Acts as gate, where the input x is either passed through or suppressed based on a probability derived from the Gaussian distribution.
 - large positive x , the gate \approx ON and behaves like ReLU
 - large negative x , the gate \approx OFF and behaves like ReLU
 - $x=0 \Rightarrow$ output=0 (but smooth)



$$\text{GELU}(x) = xP(X \leq x) = x\Phi(x) = x \cdot \frac{1}{2} \left[1 + \text{erf}(x/\sqrt{2}) \right]$$

GPT-1 Experiments

- Pre-trained on the Books Corpus dataset.
- Fine-tuned on a variety of NLP benchmarks, including:
 - Natural Language Inference tasks
 - Question Answering
 - Text Classification
- Outperforms task-specific architectures on some benchmarks.



BookCorpus consists of the text of around 11,000 unpublished books scraped from the Internet.

(74M sentences and 1G words of 16 different sub-genres)

GPT-1 Benchmark

Natural Language inference

Model determines relationship between 2 sentences-

- Entailment: The second sentence logically follows from the first.
- Contradiction: The second sentence contradicts the first.
- Neutral: The second sentence neither follows nor contradicts the first.

Method	MNLI-m	MNLI-mm	SNLI	SciTail	QNLI	RTE
ESIM + ELMo [44] (5x)	-	-	<u>89.3</u>	-	-	-
CAFE [58] (5x)	80.2	79.0	<u>89.3</u>	-	-	-
Stochastic Answer Network [35] (3x)	<u>80.6</u>	<u>80.1</u>	-	-	-	-
CAFE [58]	78.7	77.9	88.5	<u>83.3</u>		
GenSen [64]	71.4	71.3	-	-	<u>82.3</u>	59.2
Multi-task BiLSTM + Attn [64]	72.2	72.1	-	-	82.1	61.7
Finetuned Transformer LM (ours)	82.1	81.4	89.9	88.3	88.1	56.0

GPT-1 Benchmark

Question Answering (QA) & Common Sense Reasoning (CSR)

- QA - Model answers questions based on a given passage. The answer can be a span directly from the context (Extractive QA), may not directly appear (Abstractive QA), or can be MCQ based.
- CSR - Model reasons about situations that humans intuit but the facts are not explicitly stated.

Method	Story Cloze	RACE-m	RACE-h	RACE
val-LS-skip [55]	76.5	-	-	-
Hidden Coherence Model [7]	<u>77.6</u>	-	-	-
Dynamic Fusion Net [67] (9x)	-	55.6	49.4	51.2
BiAttention MRU [59] (9x)	-	<u>60.2</u>	<u>50.3</u>	<u>53.3</u>
Finetuned Transformer LM (ours)	86.5	62.9	57.4	59.0

GPT-2

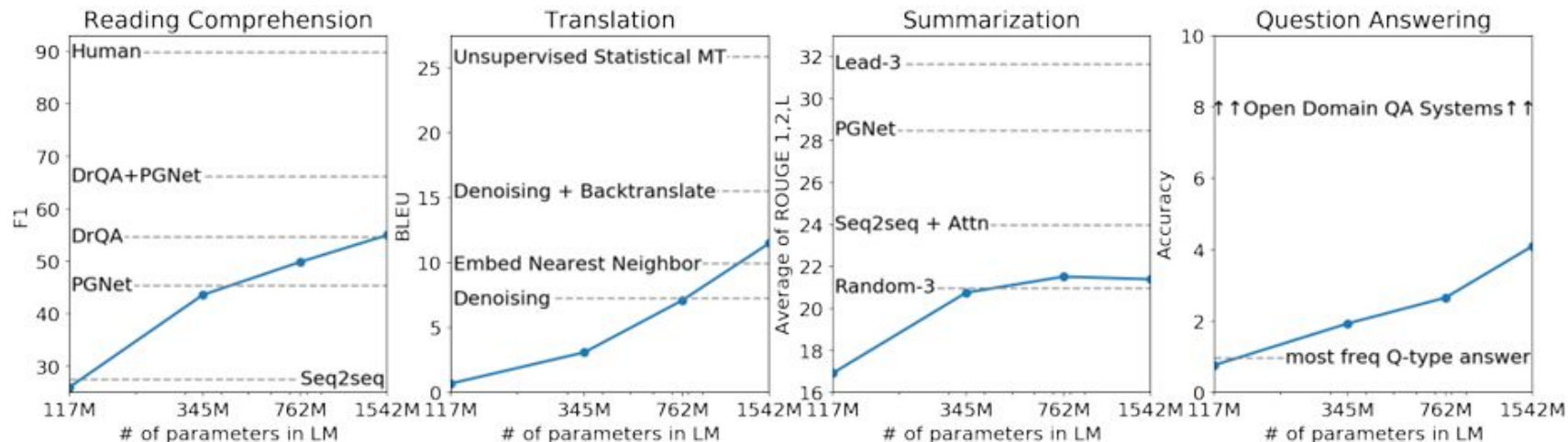
- **SCALE UP 10x !!** (117M -> 1.5B)
- Pre-trained on WebText.
- Task Conditioning - model is given extra information (prompts, or input modifications) to guide its response.
- No task-specific fine-tuning.
- Evaluated on a wide range of tasks (without fine-tuning)
- Approaches (does not beat) SOTA in zero-shot settings on several tasks.

Parameters	Layers	dmodel
117M	12	768
345M	24	1024
762M	36	1280
1542M	48	1600

WebText is an internal OpenAI corpus created by scraping web pages. Authors scraped all outbound links from Reddit which received at least 3 karma (heuristic indicator for whether other users found the link interesting, educational, or funny)

Over 8 million documents, total 40 GB of text. (excluded Wikipedia Links)

GPT-2 Experiments



Zero-shot task performance of WebText LMs as a function of model size on many NLP tasks. Reading Comprehension results are on CoQA (Reddy et al., 2018), translation on WMT-14 Fr-En (Artetxe et al., 2017), summarization on CNN and Daily Mail (See et al., 2017), and Question Answering on Natural Questions (Kwiatkowski et al., 2019)

GPT-2 Benchmark

Zero Shot Performance

- model is directly tested on a task without fine-tuning on any labelled data

	LAMBADA (PPL)	LAMBADA (ACC)	CBT-CN (ACC)	CBT-NE (ACC)	WikiText2 (PPL)	PTB (PPL)	enwik8 (BPB)	text8 (BPC)	WikiText103 (PPL)	1BW (PPL)
SOTA	99.8	59.23	85.7	82.3	39.14	46.54	0.99	1.08	18.3	21.8
117M	35.13	45.99	87.65	83.4	29.41	65.85	1.16	1.17	37.50	75.20
345M	15.60	55.48	92.35	87.1	22.76	47.33	1.01	1.06	26.37	55.72
762M	10.87	60.12	93.45	88.0	19.93	40.31	0.97	1.02	22.05	44.575
1542M	8.63	63.24	93.30	89.05	18.34	35.76	0.93	0.98	17.48	42.16

GPT-2 Benchmark

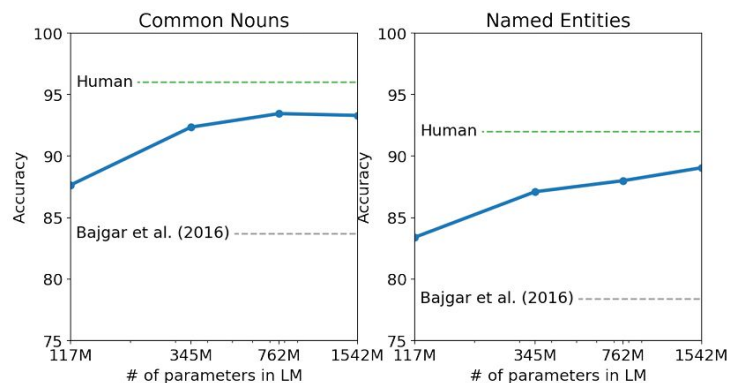
Summarization Performance

- Mode inputs long form text & generates a shorter coherent text.
- Dataset - CNN & Daily Mail news articles paired with human-written summaries.
- Calculates ROUGE Score
- Slightly lower than fine-tuned models (~40+), but notable because GPT-2 wasn't explicitly trained.

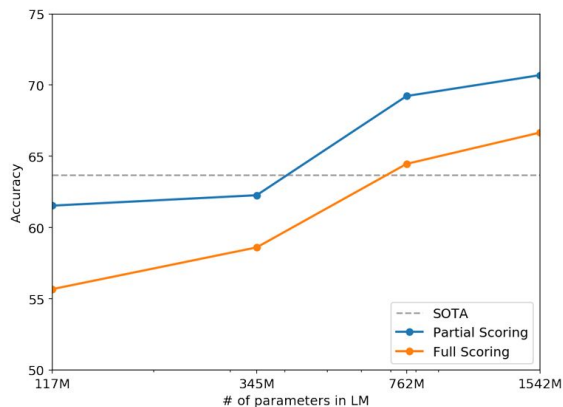
	R-1	R-2	R-L	R-AVG
Bottom-Up Sum	41.22	18.68	38.34	32.75
Lede-3	40.38	17.66	36.62	31.55
Seq2Seq + Attn	31.33	11.81	28.83	23.99
GPT-2 _{TL; DR:}	29.34	8.27	26.58	21.40
Random-3	28.78	8.63	25.52	20.98
GPT-2 no hint	21.58	4.03	19.47	15.03

GPT-2 Benchmark

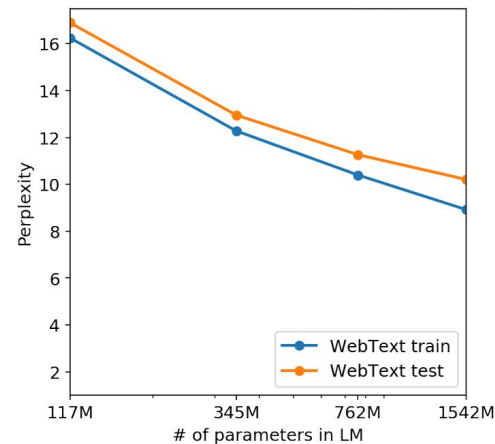
Effect of Model Size



Performance on the Children's Book test as a function of model capacity. Human performance are from Bajgar et al.(2016)



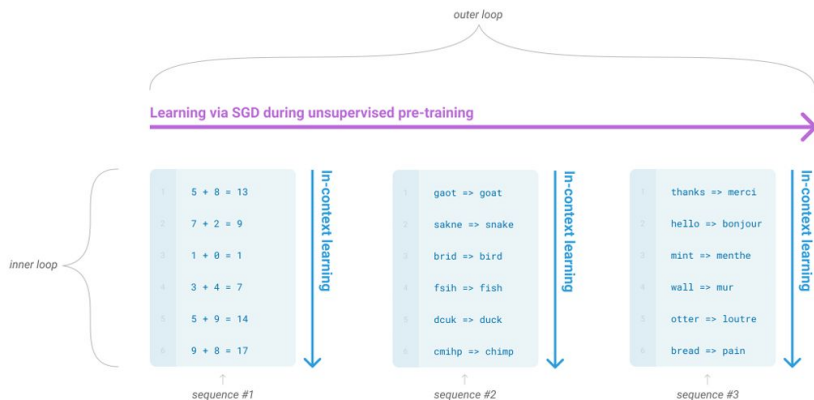
Performance on the Winograd Schema Challenge as a function of model capacity.



Performance on WebText as a function of model size.

GPT-3

- **SCALE UP 100x !!!** (1.5B -> 175B)
- Pre-trained on a diverse dataset, including Common Crawl, WebText, books, and Wikipedia.
- Highlight- In context learning
- Evaluated on a wide range of tasks in zero-shot, one-shot, and few-shot settings.
- Achieved state-of-the-art results in few-shot and zero-shot settings.



Model Name	n_{params}	m_{layers}	d_{model}	m_{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}

We use the term “in-context learning” to describe the inner loop of this process, which occurs within the forward-pass upon each sequence.

GPT-3

In-Context Learning

- Model develops general skills, pattern recognition used readily at inference to adapt quickly to the task.
- Only Forward pass during inference.
- Emergent abilities (basic arithmetic, code generation) that weren't explicitly trained.

The three settings we explore for in-context learning

Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
1 Translate English to French: ← task description
2 cheese => ..... ← prompt
```

One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

```
1 Translate English to French: ← task description
2 sea otter => loutre de mer ← example
3 cheese => ..... ← prompt
```

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

```
1 Translate English to French: ← task description
2 sea otter => loutre de mer ← examples
3 peppermint => menthe poivrée ←
4 plush giraffe => girafe peluche ←
5 cheese => ..... ← prompt
```

Traditional fine-tuning (not used for GPT-3)

Fine-tuning

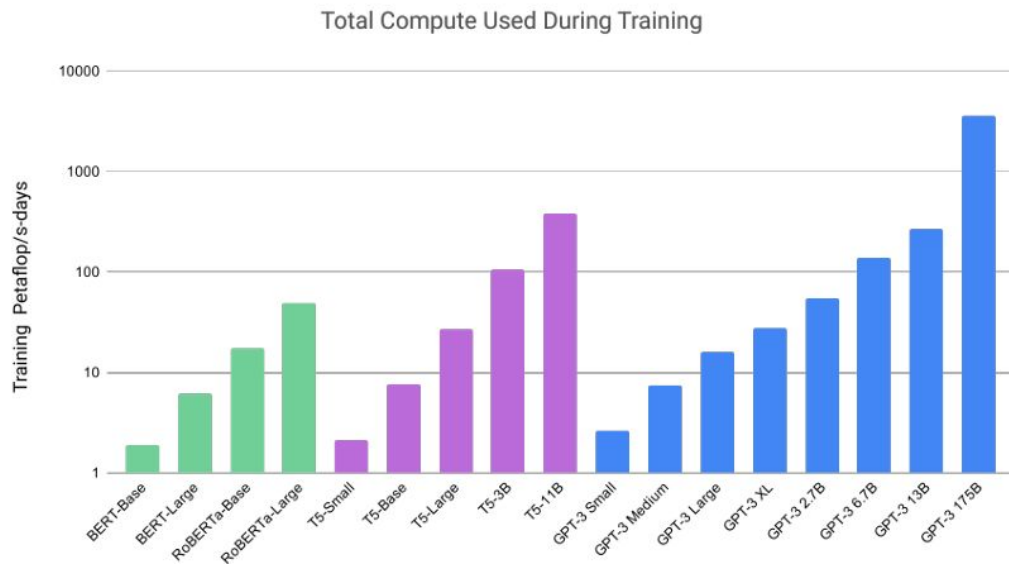
The model is trained via repeated gradient updates using a large corpus of example tasks.

```
1 sea otter => loutre de mer ← example #1
↓
gradient update
↓
1 peppermint => menthe poivrée ← example #2
↓
gradient update
↓
...
↓
1 plush giraffe => girafe peluche ← example #N
↓
gradient update
↓
1 cheese => ..... ← prompt
```

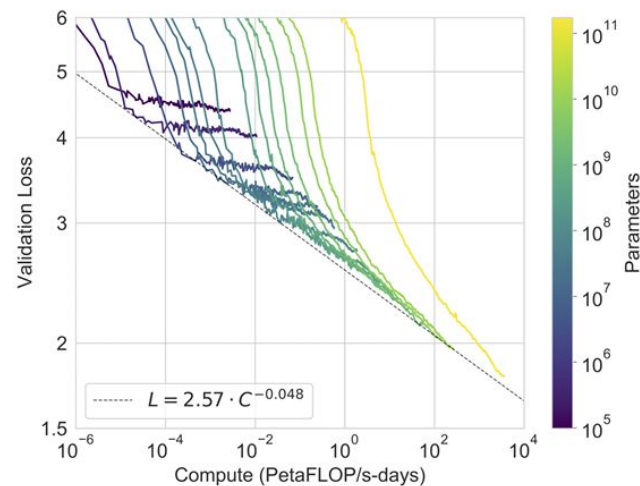
Few-shot require the model to perform the task with only forward passes at test time.

GPT-3

Scaling Laws

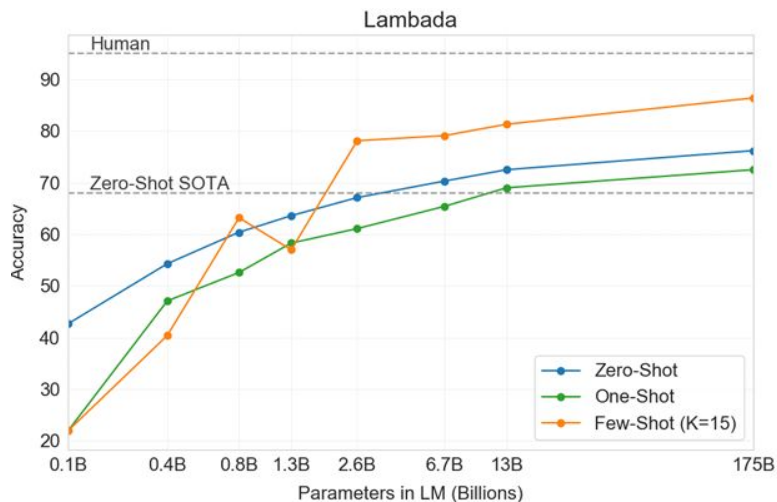


Scaling Laws For Neural Language Models suggest that we train much larger models on many fewer tokens than is typical. As a consequence, although GPT-3 3B is almost 10x larger than RoBERTa-Large (355M params), both models took roughly 50 petaflop/s-days of compute during pre-training.

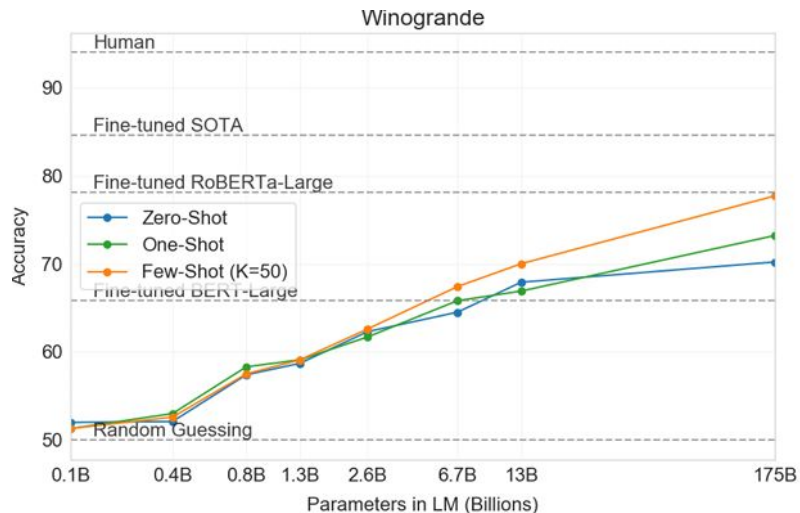


Performance (measured in terms of cross-entropy validation loss) follows a power-law trend with the amount of compute used for training.

GPT-3 Benchmarks

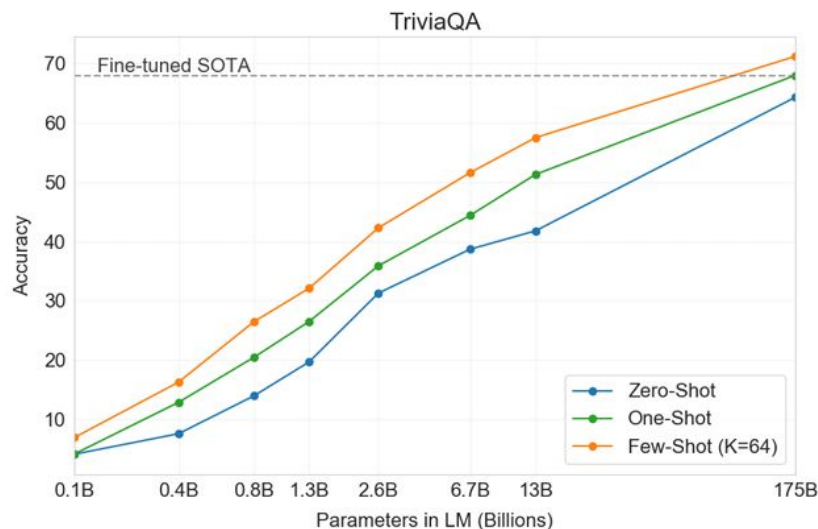


GPT-3 2.7B outperforms the SOTA 17B parameter Turing-NLG in this setting, and GPT-3 175B advances the state of the art by 18%



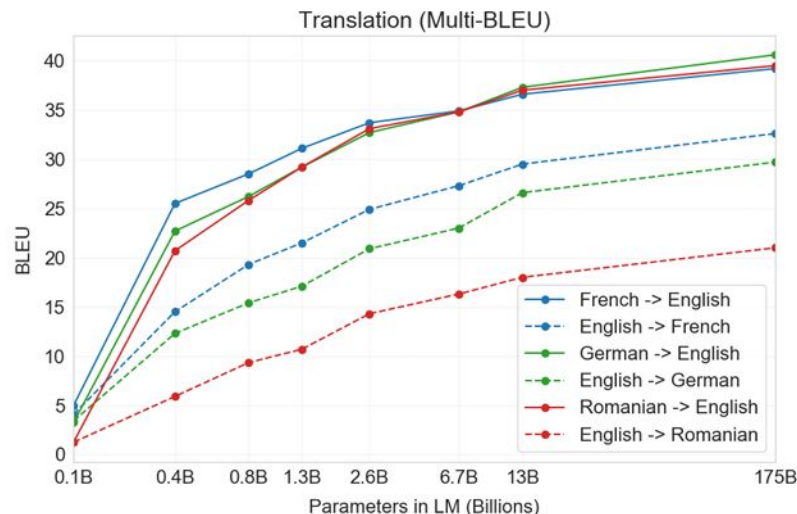
Scaling is relatively smooth with the gains to few-shot learning increasing with model size. Few-shot GPT-3 175B is competitive with a fine-tuned RoBERTa-large.

GPT-3 Benchmarks



(TriviaQA = 650k pairs)

One-shot and few-shot performance make significant gains over zero-shot behavior, matching and exceeding the performance of the SOTA

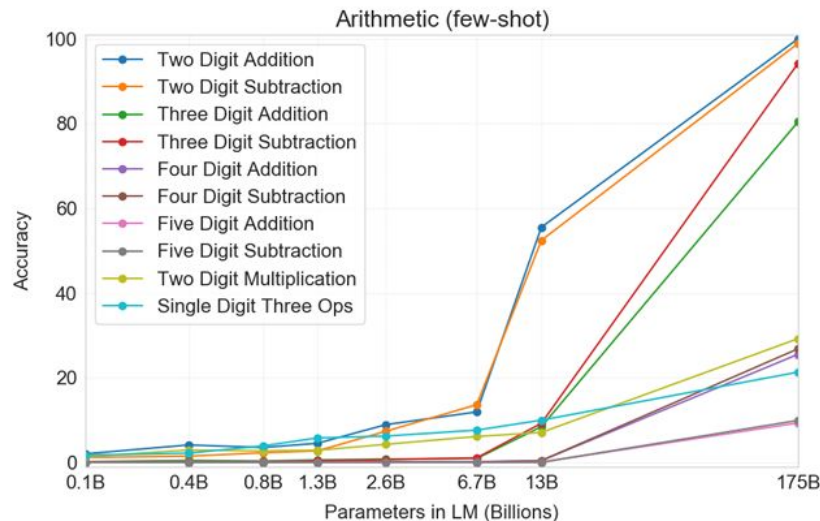


Consistent trend of improvement across all datasets as model scales. Tendency for translation into English to be stronger than translation from English.

GPT-3 Benchmarks

Setting	2D+	2D-	3D+	3D-	4D+	4D-	5D+	5D-	2Dx	1DC
GPT-3 Zero-shot	76.9	58.0	34.2	48.3	4.0	7.5	0.7	0.8	19.8	9.8
GPT-3 One-shot	99.6	86.4	65.5	78.7	14.0	14.0	3.5	3.8	27.4	14.3
GPT-3 Few-shot	100.0	98.9	80.4	94.2	25.5	26.8	9.3	9.9	29.2	21.3

Basic arithmetic tasks for GPT-3 175B. Results become progressively stronger moving from the zero-shot to one-shot to few-shot setting.



All small models do poorly on all of these tasks— even the 13 billion parameter model (the second largest) can solve 2 digit addition and subtraction only half the time, and all other operations less than 10% of the time. There is a significant jump from the second largest model (GPT-3 13B) to the largest model (GPT-3 175B)

GPT Architecture Comparison

	GPT-1	GPT-2	GPT-3	GPT 3.5 (est.)	GPT 4 (est.)
Parameters	117 M	1.5 B	175 B	175 B	1 T+
Decoder Layers	12	48	96	96	120+
Attention heads per layer	12	25	96	96	128+
Hidden Units Dimension	768	1600	12,288	12,288	16,384+
Context Window (tokens)	512	1024	2048	4096	8192
Activation	GELU	GELU	GELU	GELU	GELU/SiLU

FUTURE WORK

- Does few-shot learning actually learn new tasks “from scratch” at inference time, or simply recognizes and identifies tasks learned during training. Ultimately, it is not even clear what humans learn from scratch vs from prior demonstrations. Understanding precisely how few-shot learning works is an important unexplored direction.
- Expensive and inconvenient to perform inference on, which may present a challenge for practical applicability of models of this scale. One possible future direction to address this is distillation of large models down to a manageable size for specific tasks.
- Retains the biases of the data it has been trained on, may lead the model to generate stereotyped or prejudiced content. But not over-correction!!

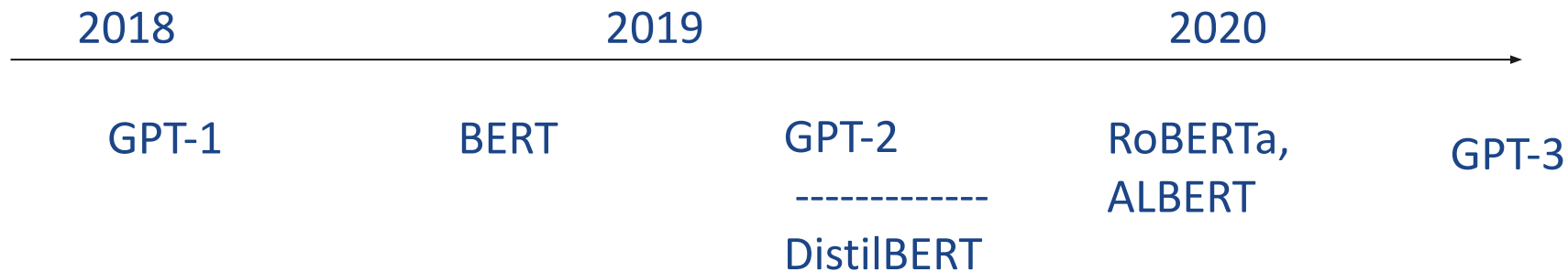
FUTURE WORK

- Still sees much more text during pre-training than a human sees in their lifetime. Improving pre-training sample efficiency is an important direction.
- Not grounded in other domains of experience, such as video or real-world physical interaction, and thus lack a large amount of context about the world. Promising future directions include learning the objective function from humans, fine-tuning with reinforcement learning, or adding additional modalities such as images to provide grounding and a better model of the world.

BERT vs GPT

Jack Bosco
jab2516

Timeline



Pretraining GPT

- GPTs use token prediction (autoregressive) for pre training
- This method predates transformers
 - Used to pre train LSTM classifiers in 2015
- Works only for transformer decoders (why?)

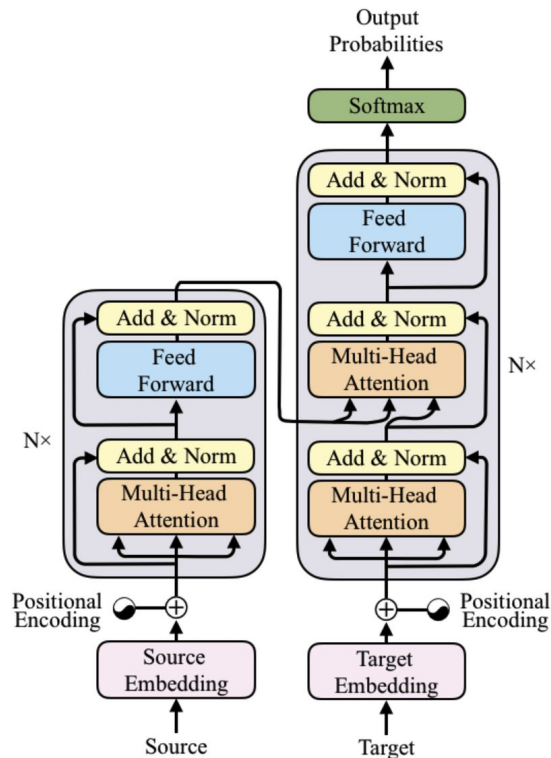
Semi-supervised Sequence Learning

Andrew M. Dai
Google Inc.
adai@google.com

Quoc V. Le
Google Inc.
qvl@google.com

Abstract

We present two approaches that use unlabeled data to improve sequence learning with recurrent networks. The first approach is to predict what comes next in a sequence, which is a conventional language model in natural language processing. The second approach is to use a sequence autoencoder, which reads the input sequence into a vector and predicts the input sequence again. These two algorithms can be used as a “pretraining” step for a later supervised sequence learning algorithm. In other words, the parameters obtained from the unsupervised step can be used as a starting point for other supervised training models. In our experiments, we find that long short term memory recurrent networks after being pretrained with the two approaches are more stable and generalize better. With pretraining, we are able to train long short term recurrent networks up to a few hundred timesteps, thereby achieving strong performance in many text classification tasks, such as IMDB, DBpedia and 20 Newsgroups.



Pretraining BERT: Problem

- MAJOR DIFFERENCE: BERT IS AN ENCODER, GPT IS A DECODER
- An encoder takes in some input and outputs the same input in a refined way
- A decoder takes in some input and makes a prediction
- Given bidirectional encoder has no masking, it is unclear what the objective function would be; we know everything to the left and right
 - For ELMo, the model is weakly bidirectional

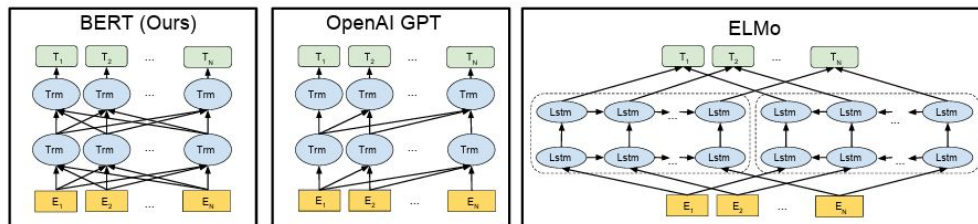
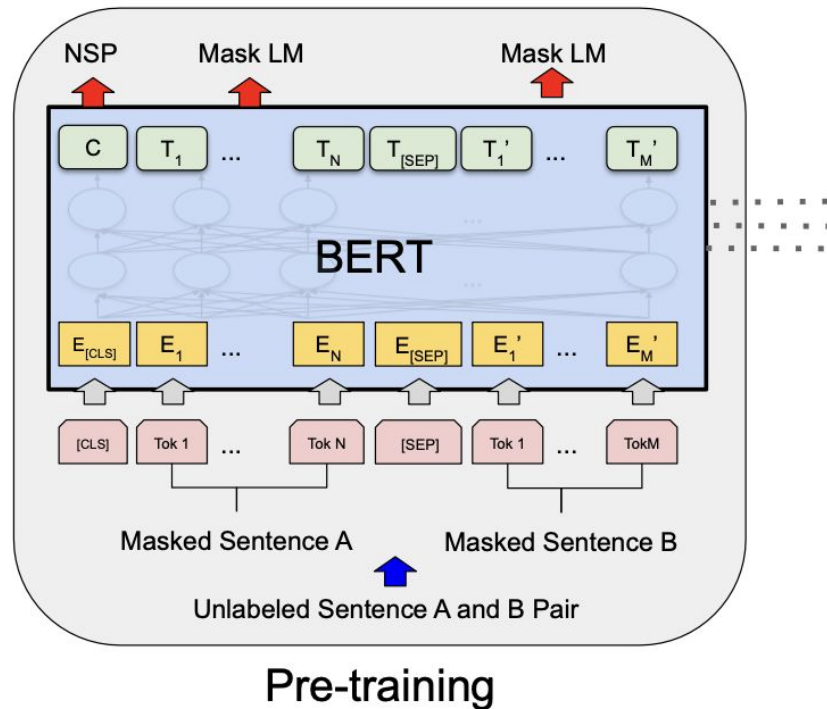


Figure 3: Differences in pre-training model architectures. BERT uses a bidirectional Transformer. OpenAI GPT uses a left-to-right Transformer. ELMo uses the concatenation of independently trained left-to-right and right-to-left LSTMs to generate features for downstream tasks. Among the three, only BERT representations are jointly conditioned on both left and right context in all layers. In addition to the architecture differences, BERT and OpenAI GPT are fine-tuning approaches, while ELMo is a feature-based approach.

Pretraining BERT

Use fine tuning objectives for pre-training

- Encoders cannot use next-token prediction for pre training
- BERT took MLM and NSP as the objective
 - MLM: Mask some words. What are the missing words?
 - Next Sentence Prediction: Pick two segments as A and B. Does B follow A?



BERT Input Representation

- Input of BERT is a pair of 2 segments (A,B)
- Tokenize A and B and insert 3 special tokens: [CLS] before A, and 2 [SEP] tokens, one between A and B and another after B
 - Positional embeddings contain information related to where a token is
 - Segment embeddings denote if a token is in the first or second sentence

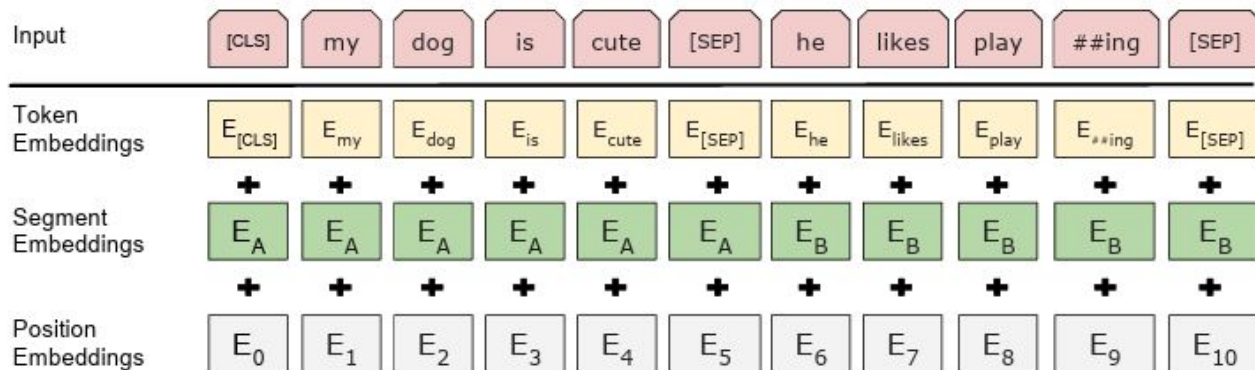


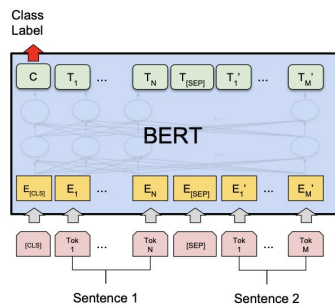
Figure 2: BERT input representation. The input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings.

Common Fine Tuning Tasks (recap)

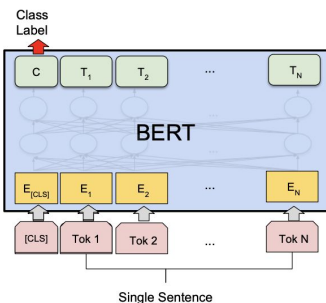
- Classification: Using the [CLS] token embedding with a classification head to predict classification from embedding
 - in practice this is the same as entailment
- Q&A: Using the Embeddings of each token to predict spans of an answer (SQUAD)
- Named-Entity Recognition: categorize person names, organizations, locations, medical codes, time expressions, quantities, etc.

Fine Tuning: Classification / Entailment

- GPT-1 restructures the prompt a little bit then just uses a FF transformation
- BERT puts an MLP on the [CLS] token



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



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

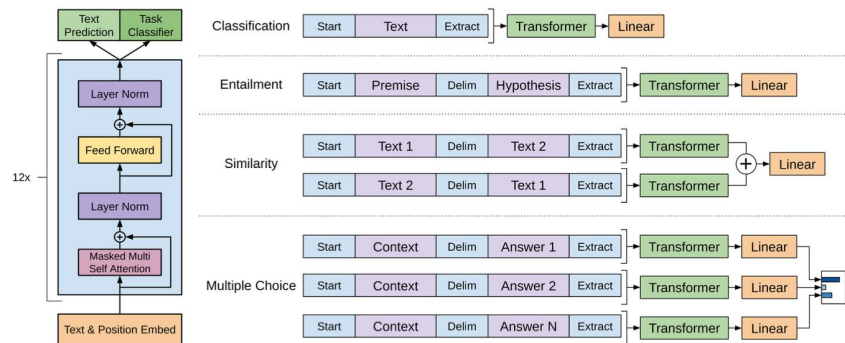
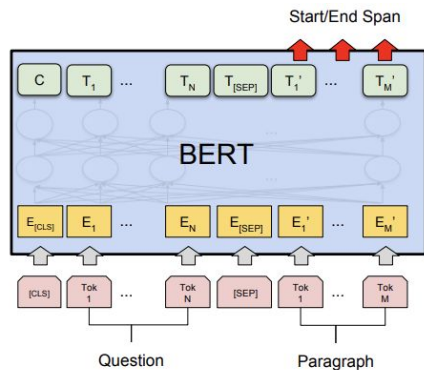


Figure 1: **(left)** Transformer architecture and training objectives used in this work. **(right)** Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer.

Fine Tuning: Q&A

- Given a <question, context> pair, find the tokens START and STOP such that the answer occurs between index START and STOP in context
- BERT introduces new tokens [START], [END]
 - i.e. Uses the dot product of [START], [END] embeddings to denote probability $P(T_i = [\text{START}]) = T_i \cdot [\text{START}]$
- GPT does poorly on Q&A, SQUAD (didn't really try)



(c) Question Answering Tasks:
SQuAD v1.1

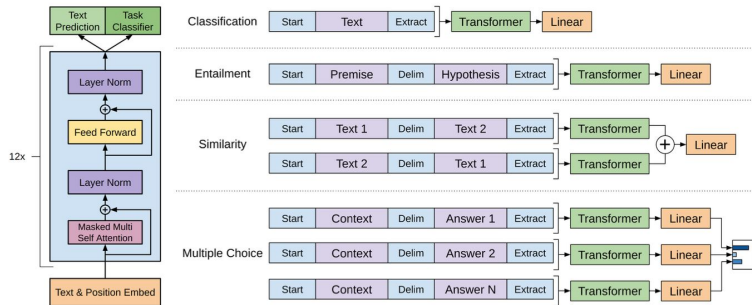
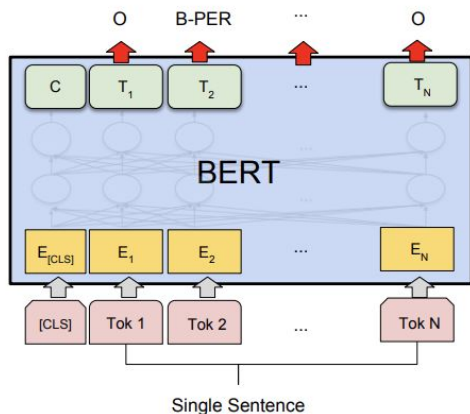


Figure 1: (left) Transformer architecture and training objectives used in this work. (right) Input transformations for fine-tuning on different tasks. **We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer.**

Fine Tuning: NER

- Neither BERT nor GPT actually fine tune for this
 - Instead both just add a MLP head on top of output



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

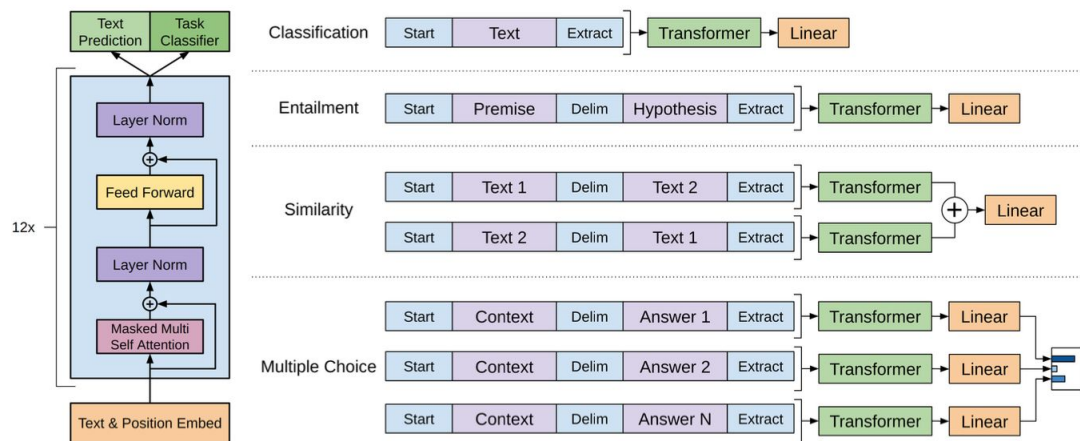


Figure 1: **(left)** Transformer architecture and training objectives used in this work. **(right)** Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer.

Zero Shot

Post-GPT1, OpenAI stopped posting serious fine-tuning results

- instead, they “prompt-engineered” the pretrained model
 - ex: GPTNER

In BERT, the terms “0-shot”, “few-shot” occurred **zero times**

Task Description	
I am an excelent linguist. The task is to label location entities in the given sentence. Below are some examples	
Input: Only France and Britain backed Fischler's proposal . Output: Only @@France## and @@Britain## backed Fischler's proposal .	Few-shot Demonstrations
Input: Germany imported 47,600 sheep from Britain last year , nearly half of total imports . Output: @@Germany## imported 47,600 sheep from @@Britain## last year , nearly half of total imports .	
Input: It brought in 4275 tonnes of British mutton . some 10 percent of overall imports . Output: It brought in 4275 tonnes of British mutton . some 10 percent of overall imports .	
Input: China says Taiwan spoils atmosphere for talks . Output: @@China## says @@Taiwan## spoils atmosphere for talks .	Input Sentence

Figure 1: The example of the prompt of GPT-NER. Suppose that we need to recognize location entities for the given sentence: *China says Taiwan spoils atmosphere for talks*. The prompt consists of three parts: (1) **Task Description**: It's surrounded by a red rectangle, and instructs the GPT-3 model that the current task is to recognize **Location** entities using linguistic knowledge. (2) **Few-shot Demonstrations**: It's surrounded by a yellow rectangle giving the GPT-3 model few-shot examples for reference. (3) **Input Sentence**: It's surrounded by a blue rectangle indicating the input sentence, and the output of the GPT-3 model is colored green.

PersonpLoclOrgoEventeDatedOtherz

Barack Hussein Obama II * (born August 4, 1961 *) is an American * attorney and politician who served as the 44th President of the United States * from January 20, 2009 *, to January 20, 2017 *. A member of the Democratic Party *, he was the first African American * to serve as president. He was previously a United States Senator * from Illinois * and a member of the Illinois State Senate *.

Side Note: Alternatives to Fine-Tuning

*instead of prompt engineering,
self-host a lightweight LM and
train parameter efficient
adaptors to fit the use case*

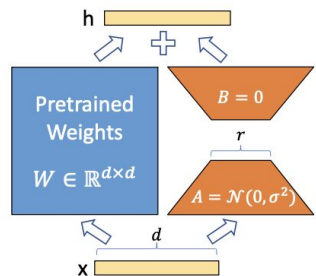


Figure 1: Our reparametrization. We only train A and B .

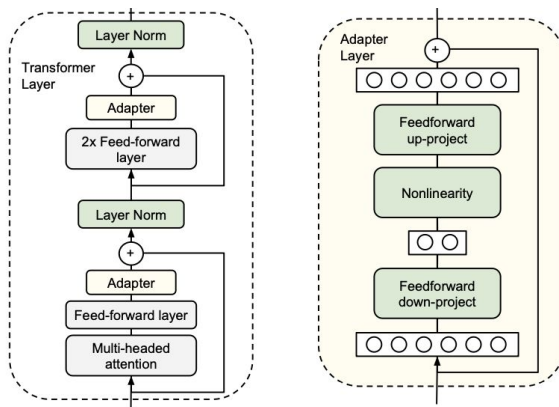
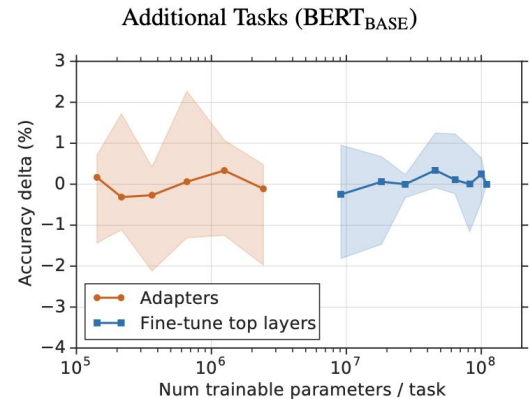
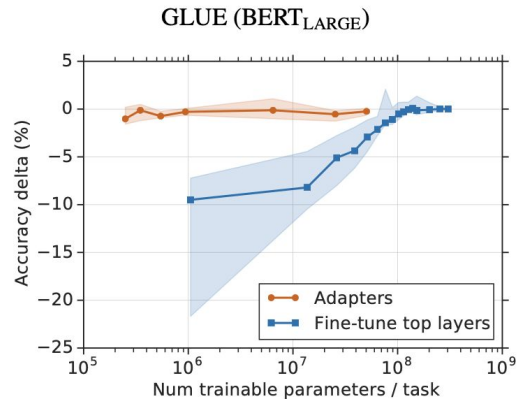


Figure 2. Architecture of the adapter module and its integration with the Transformer. **Left:** We add the adapter module twice to each Transformer layer: after the projection following multi-headed attention and after the two feed-forward layers. **Right:** The adapter consists of a bottleneck which contains few parameters relative to the attention and feedforward layers in the original model. The adapter also contains a skip-connection. During adapter tuning, the green layers are trained on the downstream data, this includes the adapter, the layer normalization parameters, and the final classification layer (not shown in the figure).

“LoRA: Low-Rank Adaptation of Large Language Models”

“Parameter-Efficient Transfer Learning for NLP”

Summary GPT vs. BERT

- Architecture
 - GPTs are unidirectional decoders (generative)
 - BERT has a bidirectional encoder, making it more cost-effective (in theory)
- Language Modeling Objectives
 - GPT uses a sentence separator ([SEP]) and classifier token ([CLS]) which are only introduced at fine-tuning time. Classic next token prediction for pre-training.
 - BERT learns [SEP], [CLS] and sentence A/B embeddings during pre-training
- Fine Tuning
 - You can't realistically fine tune any GPT past GPT-2 (without PEFT, qLoRA)
 - BERT is made to be fine-tuned for specialized representations (encodings)
- Legacy
 - Boutique BERT architectures emerge for specialized use cases
 - GPTs get bigger and bigger, applied to more general / advanced use cases

Key References and Links

- GPT-1: Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training. OpenAI. https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf
- GPT-2: Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI. https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf
- GPT-3: Brown, T., Mann, B., Ryder, N, et.al. (2020). Language models are few-shot learners. NeurIPS. <https://arxiv.org/abs/2005.14165>
- BERT: Devlin, J., Chang, M. W., et.al. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. NAACL. <https://arxiv.org/abs/1810.04805>
- GELU Activation Paper: Hendrycks, D., & Gimpel, K. (2016). Gaussian error linear units (GELUs). arXiv. <https://arxiv.org/abs/1606.08415>

transform and roll out!