Pretraining and BERT

Spring 2023

Slides adapted from Mohit Iyyer, Rui Zhang, Claire Cardie

Outline

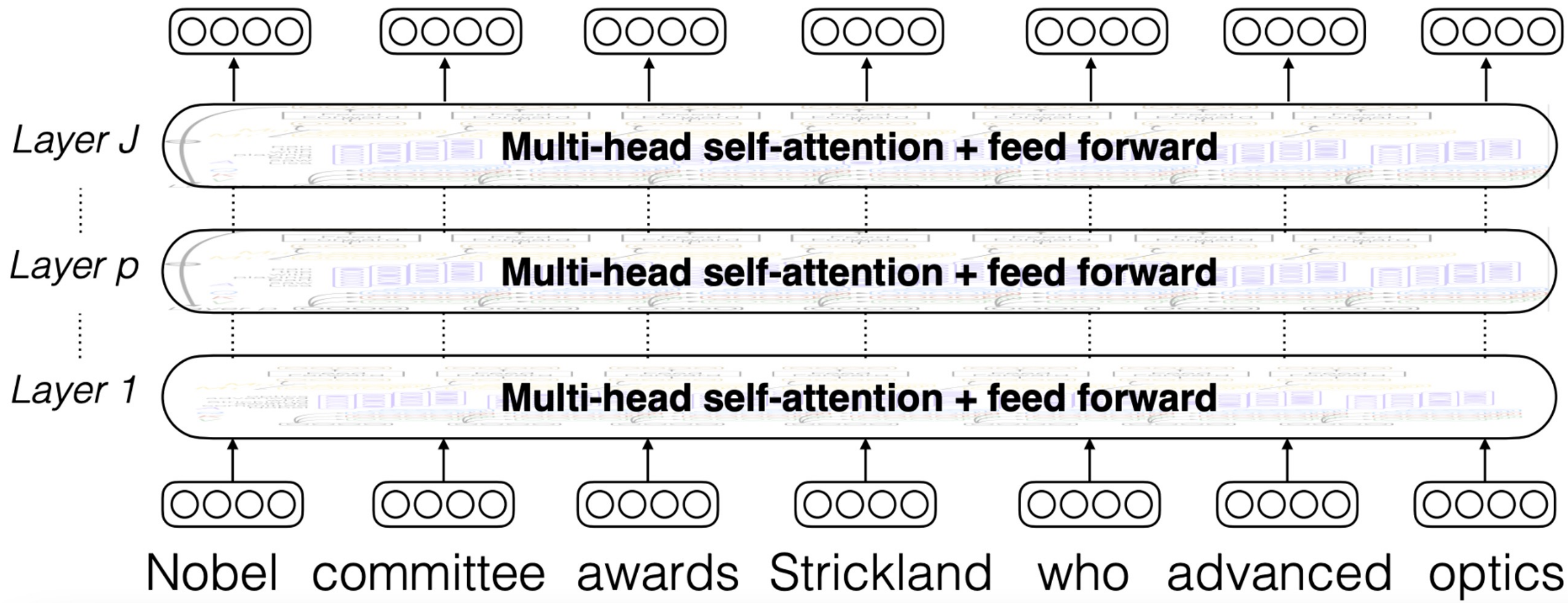
NLP

- Pretrained Language Models
- BERT and its variants
- Model Analysis

ML

- Pretraining
- Finetuning

Transformer



How is Transformer used

We can use Transformer separately for each task.
Transformer is initialized randomly, and trained for each dataset using supervised learning.

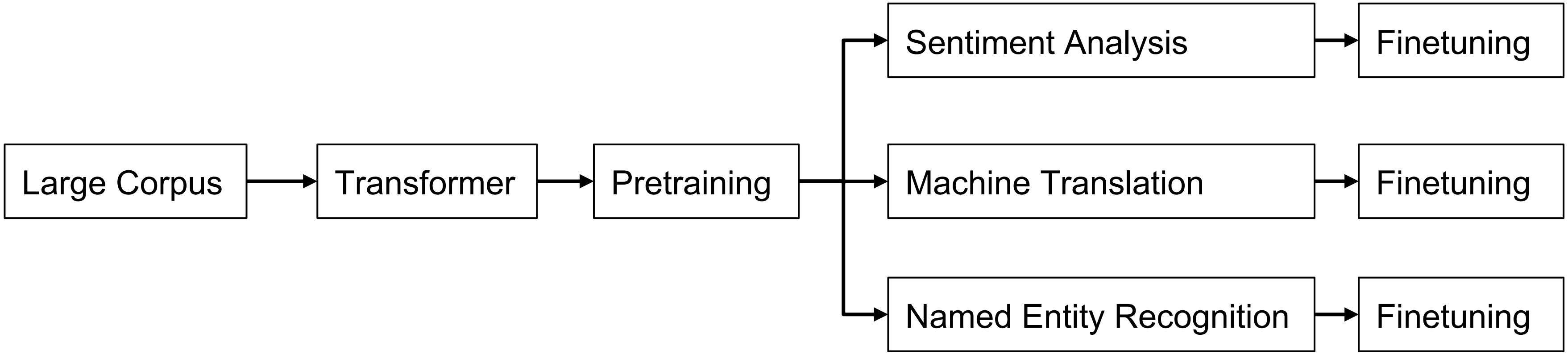
Sentiment Analysis → SA Data → Vanilla Transformer

Machine Translation → MT Data → Vanilla Transformer

Named Entity Recognition → NER Data → Vanilla Transformer

Pretraining

- First train Transformer using a lot of general text using *unsupervised* learning. This is called **pretraining**.
- Then train the pretrained Transformer for a specific task using *supervised* learning.This is called **finetuning.**
- The whole process can be called **transfer learning**.



Unsupervised pre-training

The cabs ___ the same rates as those ___ by horse-drawn cabs and were ___ quite popular, ___ the Prince of Wales (the ___ King Edward VII) travelled in ___. The cabs quickly ___ known as "hummingbirds" for ___ noise made by their motors and their distinctive black and ___ livery. Passengers ___ ___ the interior fittings were ___ when compared to ___ cabs but there ___ some complaints ___ the ___ lighting made them too ___ to those outside ___.

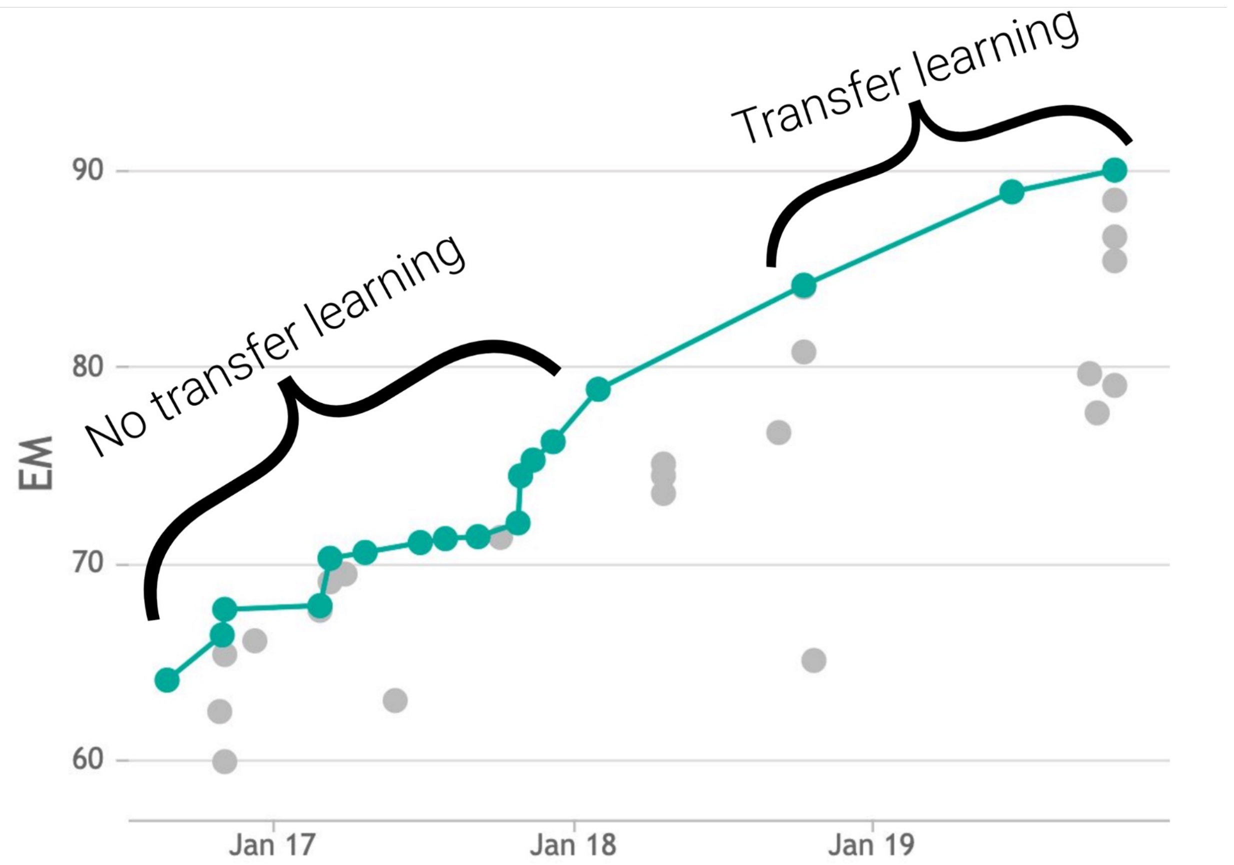
charged, used, initially, even, future, became, the, yellow, reported, that, luxurious, horse-drawn, were that, internal, conspicuous, cab

Supervised fine-tuning

This movie is terrible! The acting is bad and I was bored the entire time. There was no plot and nothing interesting happened. I was really surprised since I had very high expectations. I want 103 minutes of my life back!

negative

Slides Credit: Collin Raffel



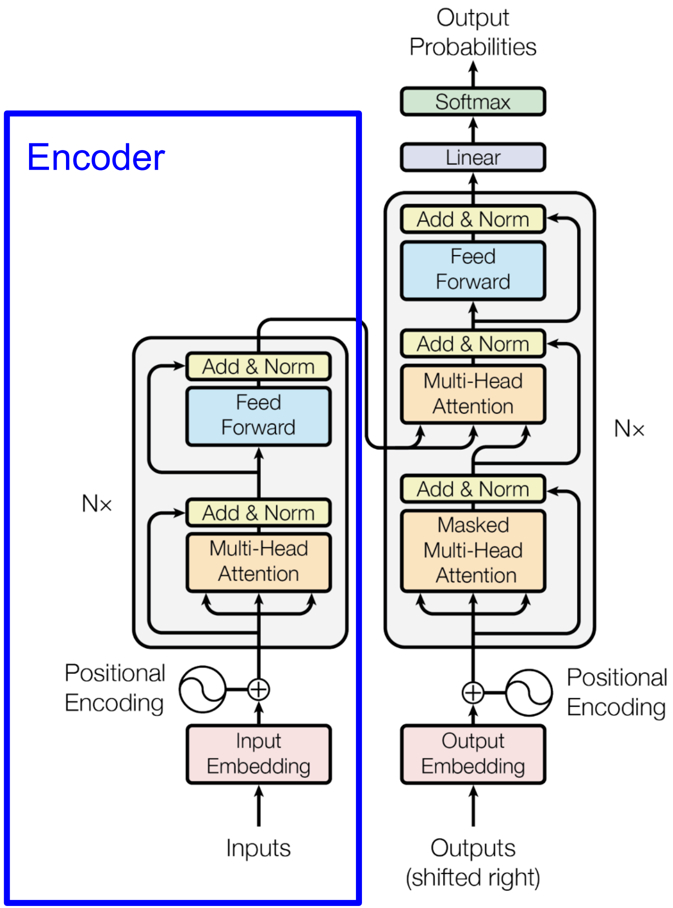
Source: https://paperswithcode.com/sota/question-answering-on-squad11-dev

Slides Credit: Collin Raffel

BERT (Devlin et al. 2018)

Model: Only use Transformer Encoder (no decoder part)

Encoder Only Model



BERT (Devlin et al. 2018)

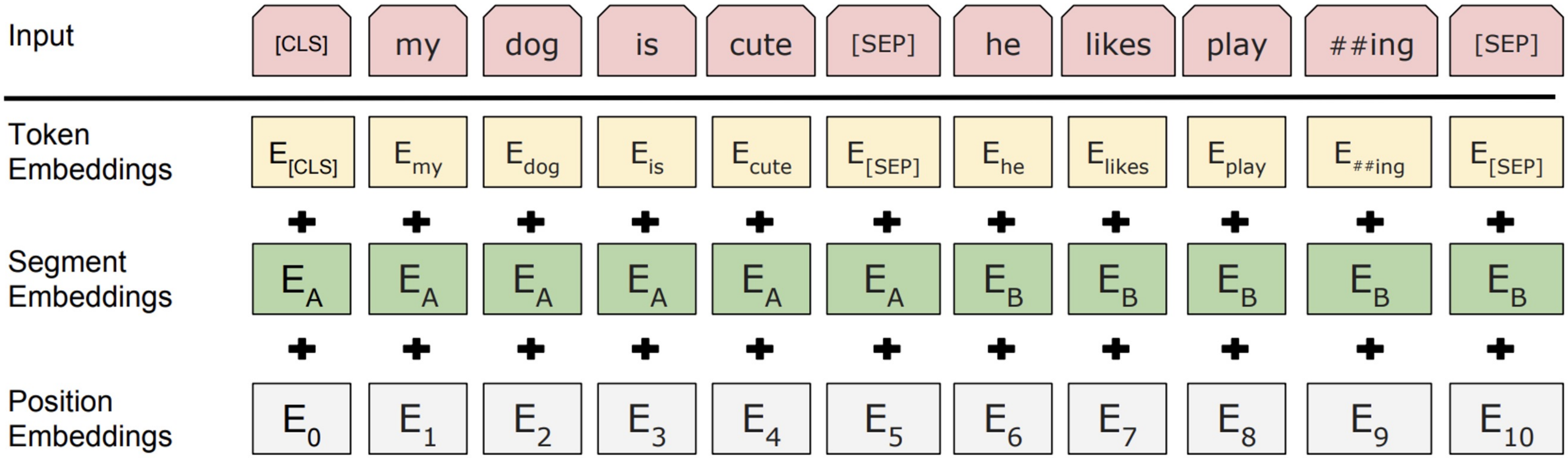
Model: Only use Transformer Encoder (no decoder part)

Data: BooksCorpus (800 million words) + English Wikipedia (2,500 million words)

Training Objective

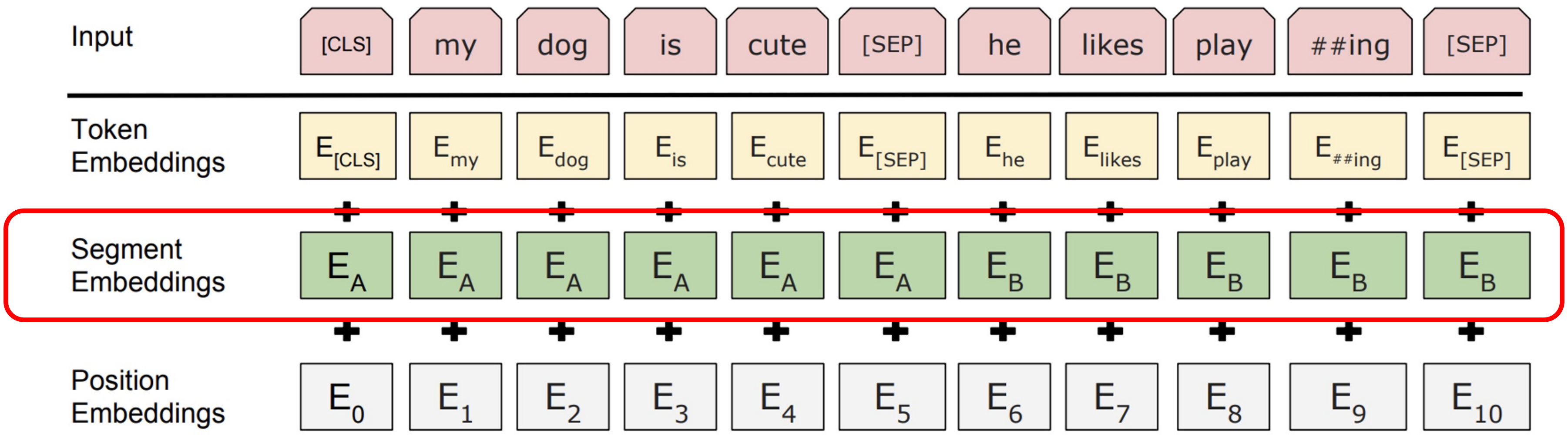
- Masked Language Modeling: predict word given bidirectional context.
- Next-sentence Prediction: predict the next sentence given the current sentence.

BERT Input



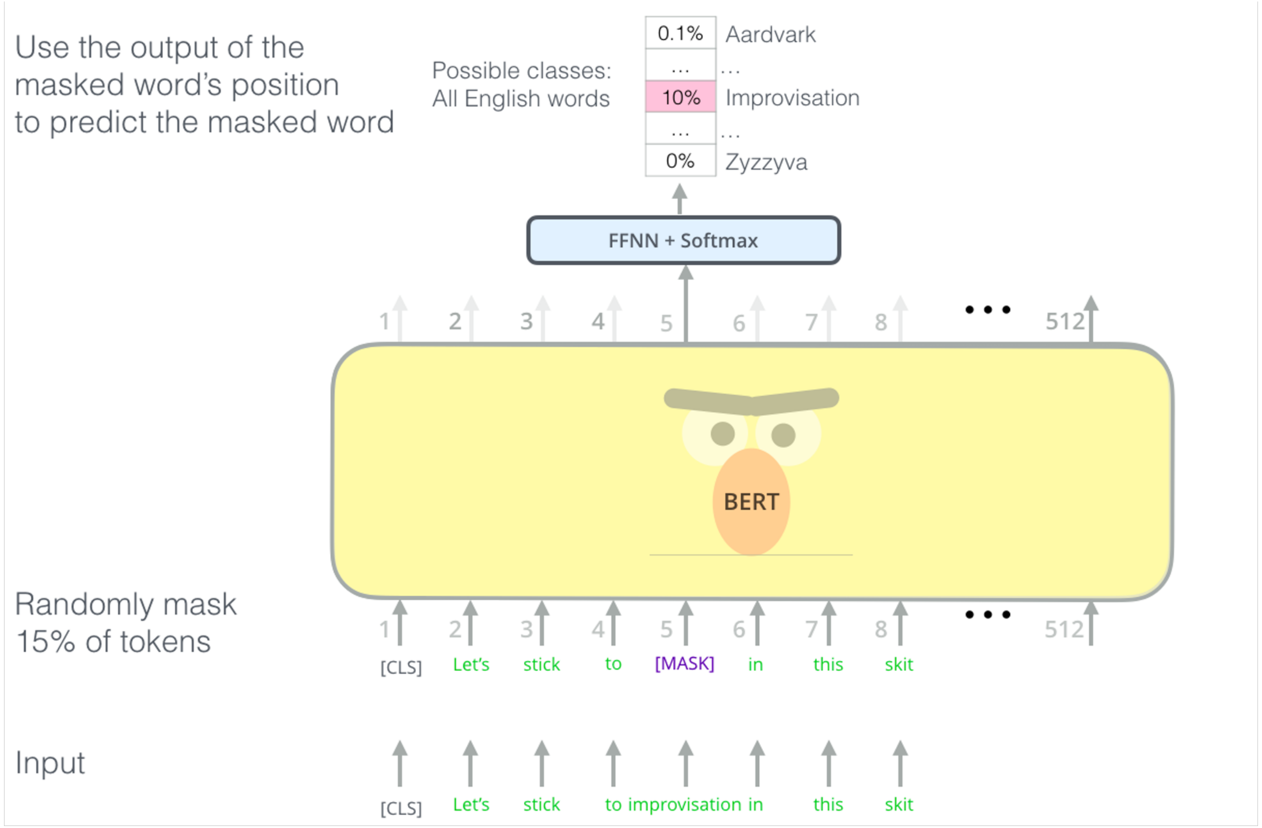
Input
[CLS]
my
dog
is
cute
[SEP]
he
likes
play
##ing
[SEP]
Token Embeddings
$E_{[CLS]}$ E_{my} E_{dog} E_{is} E_{cute} $E_{[SEP]}$ E_{he} E_{likes} E_{play} $E_{\#\#ing}$ $E_{[SEP]}$
Segment Embeddings
E_A E_A E_A E_A E_A E_A E_B E_B E_B E_B E_B
Position Embeddings
E_0 E_1 E_2 E_3 E_4 E_5 E_6 E_7 E_8 E_9 E_{10}

BERT Input



Input
[CLS]
my
dog
is
cute
[SEP]
he
likes
play
##ing
[SEP]
Token Embeddings
$E_{[CLS]}$ E_{my} E_{dog} E_{is} E_{cute} $E_{[SEP]}$ E_{he} E_{likes} E_{play} $E_{\#\#ing}$ $E_{[SEP]}$
Segment Embeddings
E_A E_A E_A E_A E_A E_A E_B E_B E_B E_B E_B
Position Embeddings
E_0 E_1 E_2 E_3 E_4 E_5 E_6 E_7 E_8 E_9 E_{10}

Training Objective 1: Masked Language Modeling

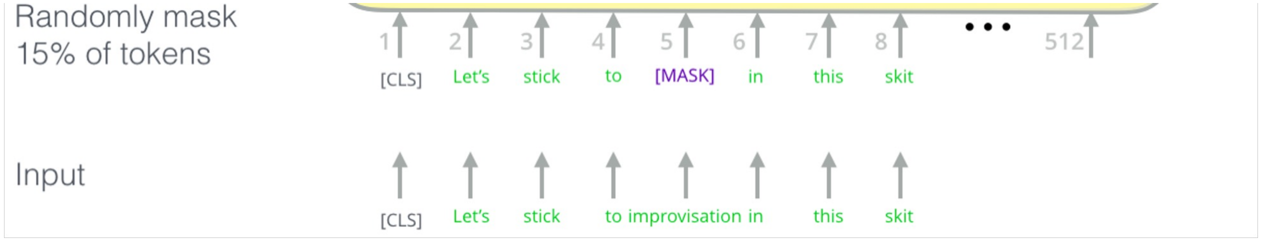


https://jalammar.github.io/illustrated-bert/

Training Objective 1: Masked Language Modeling

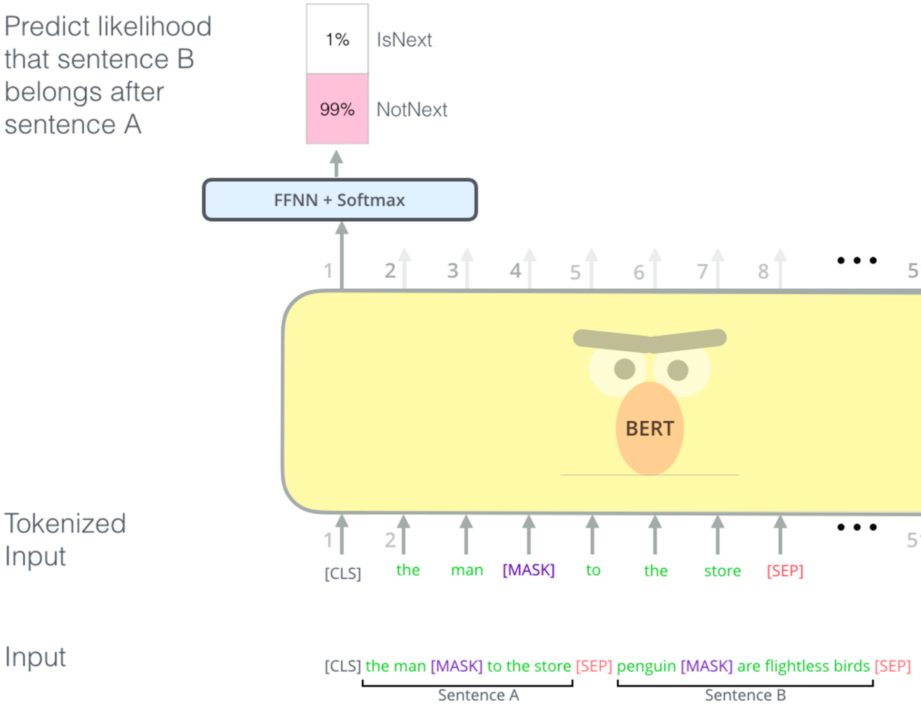
Predict a random 15% of (sub)word tokens, and of these 15%:

- 80%: Replace input word with [MASK]
- 10%: Replace input word with a random token
- 10%: Leave input word unchanged 10% (but still predict it!)



Training Objective 2: Next-sentence Prediction

Give two sentences as input, classify if the second sentence really follows the first one.



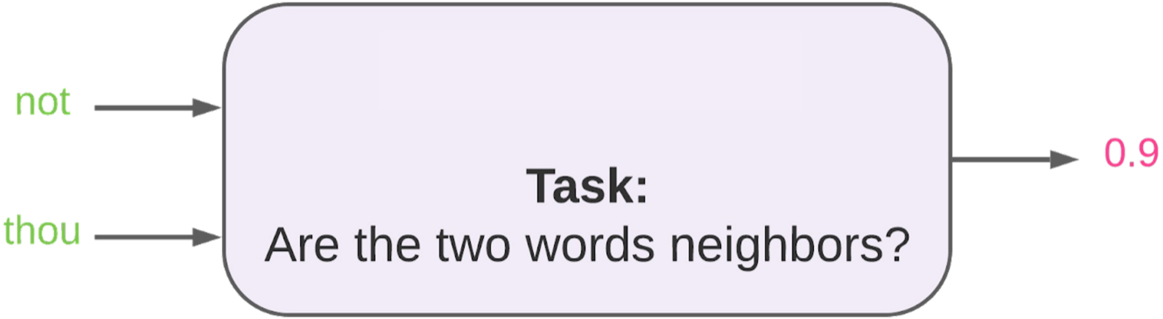
https://jalammar.github.io/illustrated-bert/

Revisit Word Representations

- After we have BERT ...
- Word Embeddings v.s. Contextualized Word Embeddings.

Word2Vec (Skip-Gram)

- Represent words as dense vectors (25 - 300 dimensions)
- Train a logistic regression classifier on whether words are neighbors
- Regression weights become embeddings for the words



What are issues with a single vector per word?

- Does not take into account multiple senses (polysemy, homonym)
- Same vector in every context (yet meaning is contextual)

Meaning is *contextual*

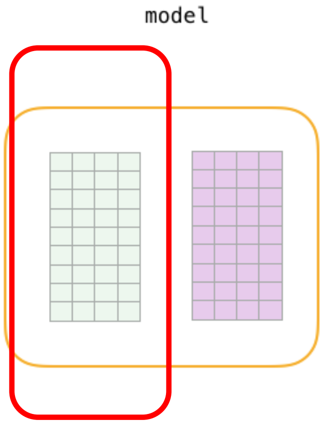
restrain:

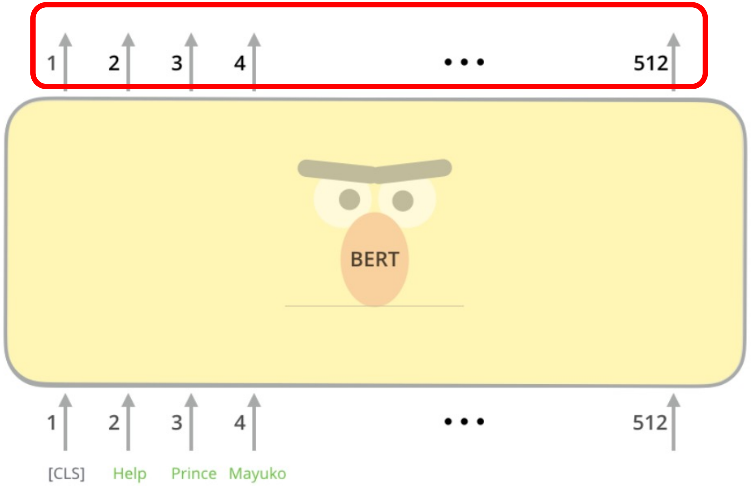
- **To hold back physically:** “His classmates had to **restrain** him from eating the last cupcake.”
- **To control emotions:** “I wasn’t able to **restrain** my excitement upon winning the tournament - I threw my ping-pong paddle into the crowd and hit my poor brother on the forehead, knocking him out.”
- **To limit:** “The embargoes and tariffs were designed to **retrain** trade.”

Contextualized word representations

dataset

input word	output word	target
not	thou	1
not	aaron	0
not	taco	0
not	shalt	1
not	mango	0
not	finglonger	0
not	make	1
not	plumbus	0
...	...	...





Paper presentations

- CS6301_paper_presentation_slides Upload slides before the class to this folder.
- We recommend that you use your own laptop
- One example (ELMo) on elearning, critiques for the paper are also welcome (e.g. limitations).
- For days where other groups are presenting: non-presenting students are expected to prepare for class by at least skimming the abstract and intro of the paper(s) to be presented
- Upload the final version to gradescope by the end of the presentation day (11:59pm)

Paper presentations

- Next week

Week 6	Feb 24	-				
		Pretrained Language Models (PLMs)				(Blog post) Generalized Language Models 2019, Pre-trained Models for Natural Language Processing: A Survey (Liu et al 2019), Percy Liang's introduction to LLMs
Week 7	Mar 3	Paper Presentation * 2 (PLMs)		**Encoder-only models**: BERT, ELECTRA,	Group 17	
				Encoder-decoder models: T5, mT5, (Optional) Flan, T0, Scaling Instruction-Finetuned Language Model (FLAN)	Group 20	
		DL for NLP applications (QA, NLG)				Huggingface Datasets, Paper with Code

quiz 1

- on eLearning
- close book
- 20 minutes

Contextualized word 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
```

Contextualized representations

- Derives contextualized representation of words
- Each token is assigned a representation that is a function of the entire input sequence

```
 [0.9, -0.2, 1.6, …]                          [-1.9, -0.4, 0.1, …]
        ↑                                              ↑
open a bank account                      on the river bank
```

Contextualized representations

- Rather than having a dictionary ‘look-up’ of words, BERT/ ELMo *creates vectors on-the-fly by passing text through a recurrent model*
- **Idea**: Pretrain BERT/ELMo as a language model then use **context vectors** for each word as pre-trained word vectors

```
  [0.9, -0.2, 1.6, …]                  [-1.9, -0.4, 0.1, …]
          ↑                                     ↑
open a bank account              on the river bank
```

ELMo (contextual) vs. GloVe (static)

	Source	Nearest Neighbors
GloVe	play	playing, game, games, played, players, plays, player, Play, football, multiplayer
biLM	Chico Ruiz made a spectacular <u>play</u> on Alusik 's grounder {. . . }	Kieffer , the only junior in the group , was commended for his ability to hit in the clutch , as well as his all-round excellent <u>play</u> .
	Olivia De Havilland signed to do a Broadway <u>play</u> for Garson {. . . }	{. . . } they were actors who had been handed fat roles in a successful <u>play</u> , and had talent enough to fill the roles competently , with nice understatement .

Table 4: Nearest neighbors to "play" using GloVe and the context embeddings from a biLM.

Come back to BERT

- BERT for different tasks

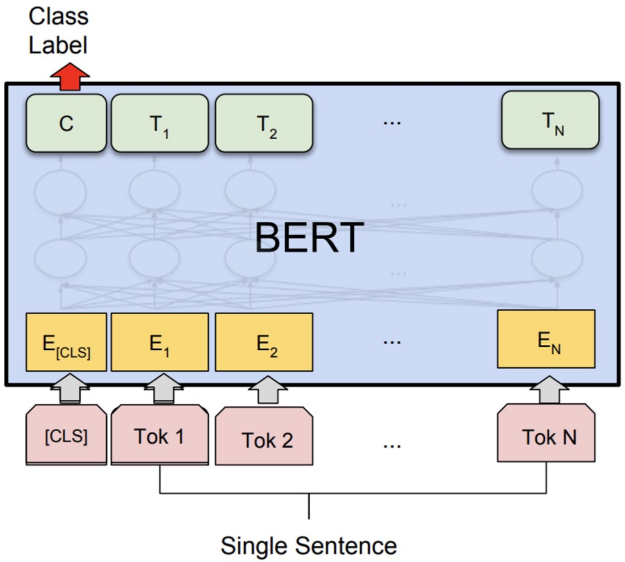
BERT on Different Tasks

We can use BERT for different tasks by changing the inputs and adding classification layers on top of output embeddings.

> *"We show that pre-trained representations reduce the need for many heavily-engineered task specific architectures. BERT is the first finetuning based representation model that achieves state-of-the-art performance on a large suite of sentence-level and token-level tasks, outperforming many task-specific architectures."*

BERT on Different Tasks

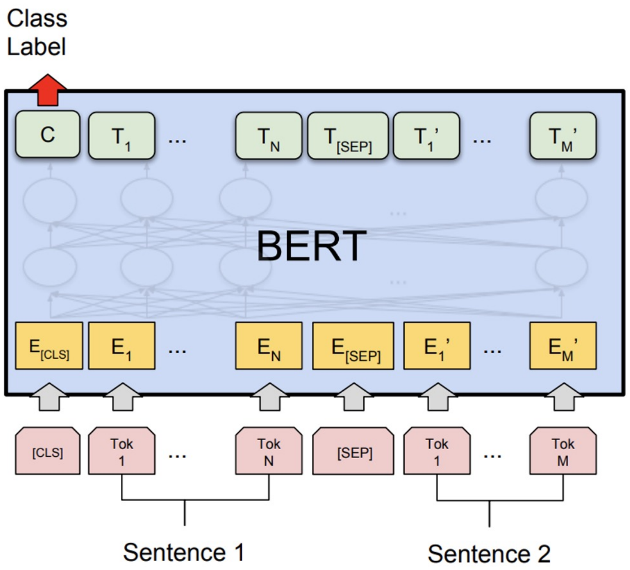
Sentence Classification



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

BERT on Different Tasks

Sentence Pair Classification



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

How powerful is BERT?

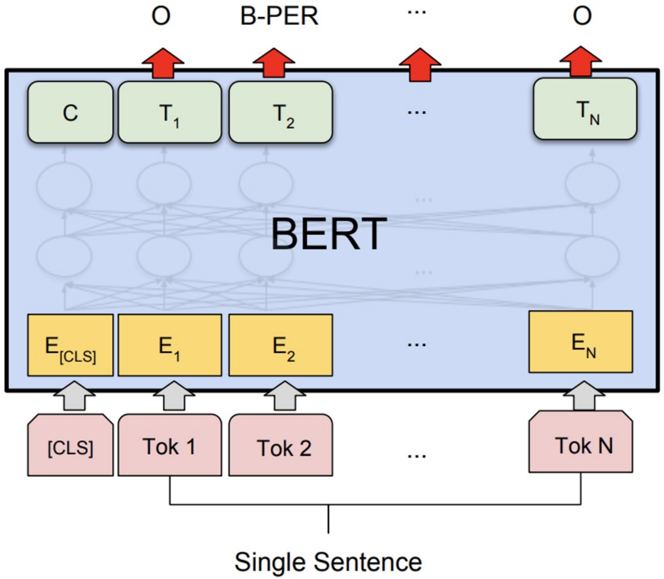
System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	**Average**
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

Table 1: GLUE Test results, scored by the evaluation server (`https://gluebenchmark.com/leaderboard`). The number below each task denotes the number of training examples. The "Average" column is slightly different than the official GLUE score, since we exclude the problematic WNLI set.[8] BERT and OpenAI GPT are single-model, single task. F1 scores are reported for QQP and MRPC, Spearman correlations are reported for STS-B, and accuracy scores are reported for the other tasks. We exclude entries that use BERT as one of their components.

Diverse set of important NLP benchmark tasks

BERT on Different Tasks

Sequence Labeling



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

BERT on Different Tasks

Question Answering (MRC type)

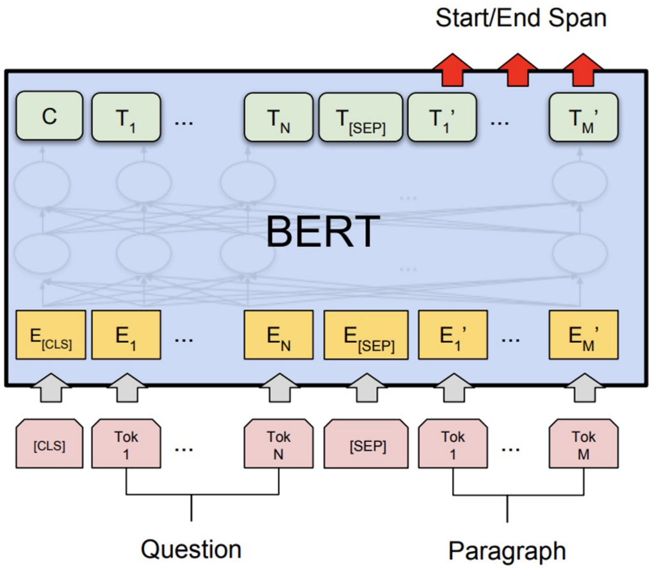
In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under **gravity**. The main forms of precipitation include drizzle, rain, sleet, snow, **graupel** and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or 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

Figure 1: Question-answer pairs for a sample passage in the SQuAD dataset. Each of the answers is a segment of text from the passage.



(c) Question Answering Tasks: SQuAD v1.1

BERT for QA

System	Dev EM	Dev F1	Test EM	Test F1
Top Leaderboard Systems (Dec 10th, 2018)				
Human	-	-	82.3	91.2
#1 Ensemble - nlnet	-	-	86.0	91.7
#2 Ensemble - QANet	-	-	84.5	90.5
Published				
BiDAF+ELMo (Single)	-	85.6	-	85.8
R.M. Reader (Ensemble)	81.2	87.9	82.3	88.5
Ours				
$\text{BERT}_{\text{BASE}}$ (Single)	80.8	88.5	-	-
$\text{BERT}_{\text{LARGE}}$ (Single)	84.1	90.9	-	-
$\text{BERT}_{\text{LARGE}}$ (Ensemble)	85.8	91.8	-	-
$\text{BERT}_{\text{LARGE}}$ (Sgl.+TriviaQA)	**84.2**	**91.1**	**85.1**	**91.8**
$\text{BERT}_{\text{LARGE}}$ (Ens.+TriviaQA)	**86.2**	**92.2**	**87.4**	**93.2**

Table 2: SQuAD 1.1 results. The BERT ensemble is 7x systems which use different pre-training checkpoints and fine-tuning seeds.

BERT

Initially two BERT models are trained and released with the paper

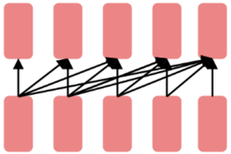
- **BERT-base**: 12 layers, 768-dim hidden states, 12 attention heads, 110 million params.
- **BERT-large**: 24 layers, 1024-dim hidden states, 16 attention heads, 340 million params.

Pretraining is expensive

- 64 TPU chips for a total of 4 days

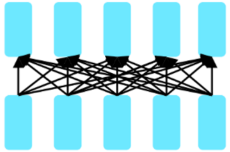
Pretraining for three types of architectures

The neural architecture influences the type of pretraining, and natural use cases.



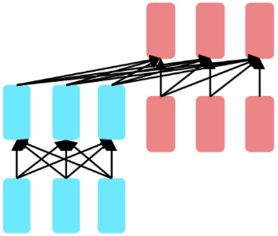
Decoders

- Language models! What we've seen so far.
- Nice to generate from; can't condition on future words



Encoders

- Gets bidirectional context – can condition on future!
- Wait, how do we pretrain them?



Encoder-Decoders

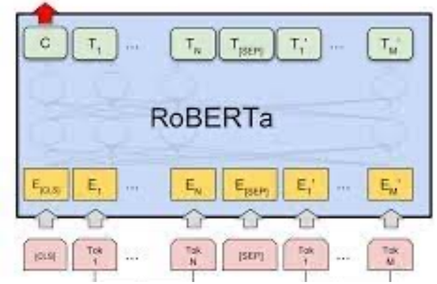
- Good parts of decoders and encoders?
- What's the best way to pretrain them?

RoBERTa (Liu et al. 2019)

RoBERTa: A Robustly Optimized BERT Pretraining Approach

Model: same as BERT
Data: same as BERT
Training Objective

- MLM same as BERT, but train longer
- Remove next-sentence prediction.



Takeaway: more compute and more data can help; *next-sentence prediction not necessary.*

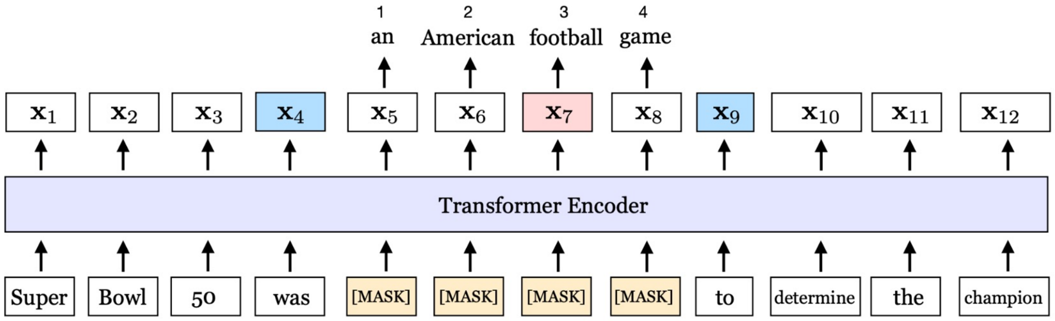
SpanBERT (Joshi et al., 2019)

SpanBERT: Improving Pre-training by Representing and Predicting Spans

Model: same as BERT
Data: same as bert
Training Objective

- Masking contiguous random spans, rather than random tokens
- Training the span boundary representations to predict the entire content of the masked span, without relying on the individual token representations within it.

Takeaway: predicting entire spans is better than random tokens



GPT (Radford et al., 2018)

GPT

- Generative Pretrained Transformer
- Generative PreTraining

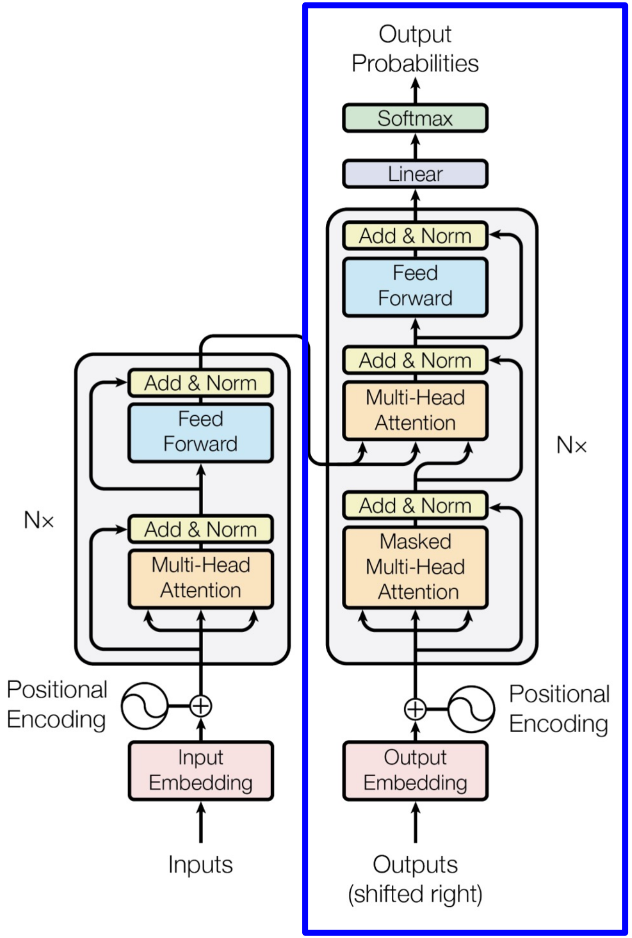
Model: only Transformer decoder.

Data: BooksCorpus: over 7000 unique books.

Training Objective: Language Modeling

Followed by GPT-2 and GPT-3

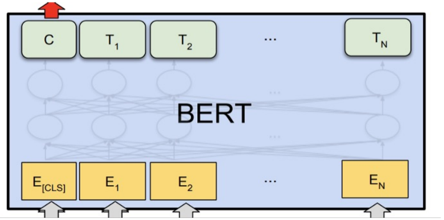
- GPT (Jun 2018): 117 million parameters
- GPT-2 (Feb 2019): 1.5 billion parameters
- GPT-3 (July 2020): 175 billion parameters



ELECTRA (Clark et al. 2020)

ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators

Model: Same as BERT
Data: Same as BERT
Training Objective:



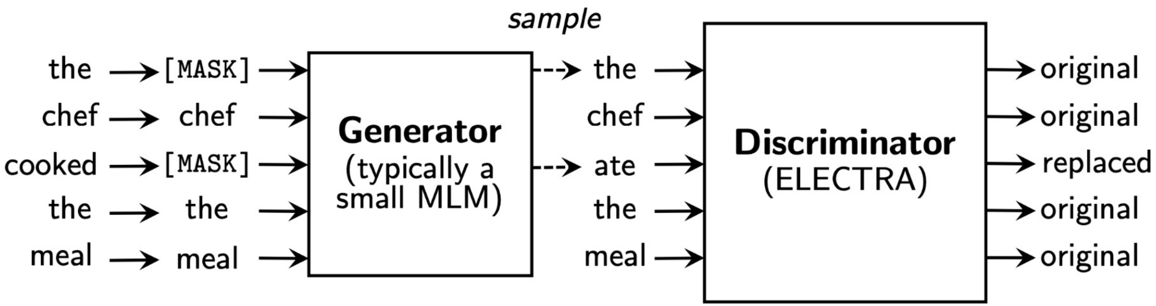


Figure 2: An overview of replaced token detection. The generator can be any model that produces an output distribution over tokens, but we usually use a small masked language model that is trained jointly with the discriminator. Although the models are structured like in a GAN, we train the generator with maximum likelihood rather than adversarially due to the difficulty of applying GANs to text. After pre-training, we throw out the generator and only fine-tune the discriminator (the ELECTRA model) on downstream tasks.

ELECTRA (Clark et al. 2020)

the task is defined over all input tokens rather than just the small subset that was masked out. As a result, the contextual representations learned by our approach substantially outperform the ones learned by BERT given the same model size, data, and compute. This makes training more efficient.

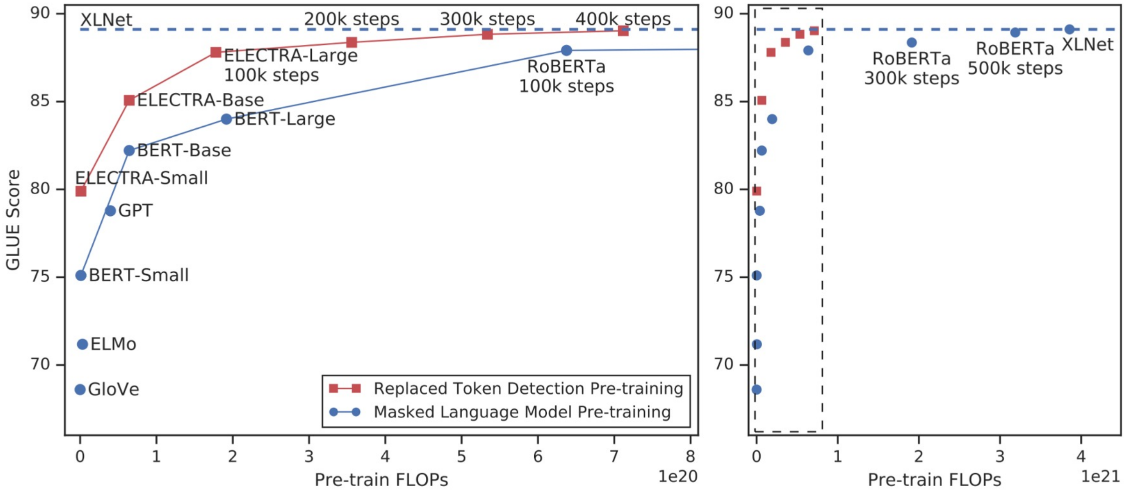
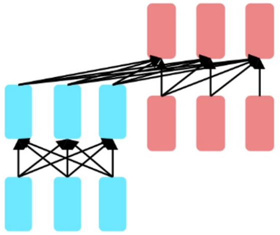


Figure 1: Replaced token detection pre-training consistently outperforms masked language model pre-training given the same compute budget. The left figure is a zoomed-in view of the dashed box.

T5 (Raffel et al., 2019)



T5: “Text-to-Text Transfer Transformer”

Treat every text processing problem as a “text-to-text” problem, i.e. taking text as input and producing new text as output.

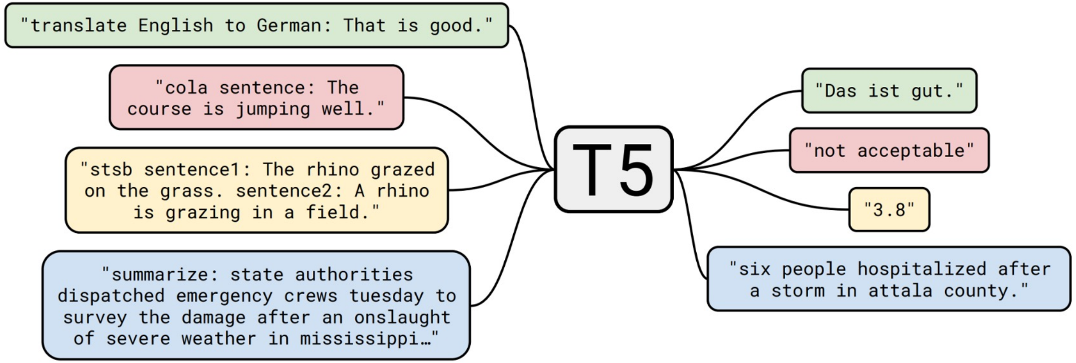


Figure 1: A diagram of our text-to-text framework. Every task we consider—including translation, question answering, and classification—is cast as feeding our model text as input and training it to generate some target text. This allows us to use the same model, loss function, hyperparameters, etc. across our diverse set of tasks. It also provides a standard testbed for the methods included in our empirical survey. “T5” refers to our model, which we dub the “**T**ext-**t**o-**T**ext **T**ransfer **T**ransformer”.

T5 (Raffel et al., 2019)

Model: Transformer Encoder-Decoder

Data: C4 (Colossal Clean Crawled Corpus), a data set consisting of hundreds of gigabytes of clean English text scraped from the web

Training Objective

- ?

Objective	Inputs	Targets
Prefix language modeling	Thank you for inviting	me to your party last week .
BERT-style Devlin et al. (2018)	Thank you <M> <M> me to your party apple week .	*(original text)*
Deshuffling	party me for your to . last fun you inviting week Thank	*(original text)*
MASS-style Song et al. (2019)	Thank you <M> <M> me to your party <M> week .	*(original text)*
I.i.d. noise, replace spans	Thank you <X> me to your party <Y> week .	<X> for inviting <Y> last <Z>
I.i.d. noise, drop tokens	Thank you me to your party week .	for inviting last
Random spans	Thank you <X> to <Y> week .	<X> for inviting me <Y> your party last <Z>

T5 (Raffel et al., 2019)

Model: Transformer Encoder-Decoder

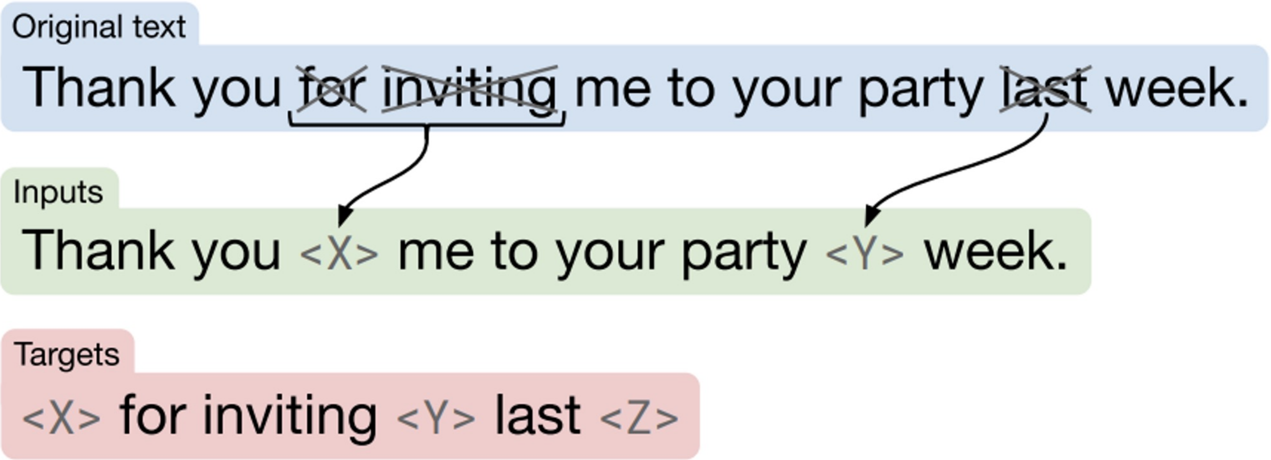
Data: C4 (Colossal Clean Crawled Corpus), a data set consisting of hundreds of gigabytes of clean English text scraped from the web

Training Objective

- **Span Corruption (denoising)**

Objective	Inputs	Targets
Prefix language modeling	Thank you for inviting	me to your party last week .
BERT-style Devlin et al. (2018)	Thank you <M> <M> me to your party apple week .	*(original text)*
Deshuffling	party me for your to . last fun you inviting week Thank	*(original text)*
MASS-style Song et al. (2019)	Thank you <M> <M> me to your party <M> week .	*(original text)*
I.i.d. noise, replace spans	Thank you <X> me to your party <Y> week .	<X> for inviting <Y> last <Z>
I.i.d. noise, drop tokens	Thank you me to your party week .	for inviting last
Random spans	Thank you <X> to <Y> week .	<X> for inviting me <Y> your party last <Z>

Span Corruption (denoising)



T5 (Raffel et al., 2019)

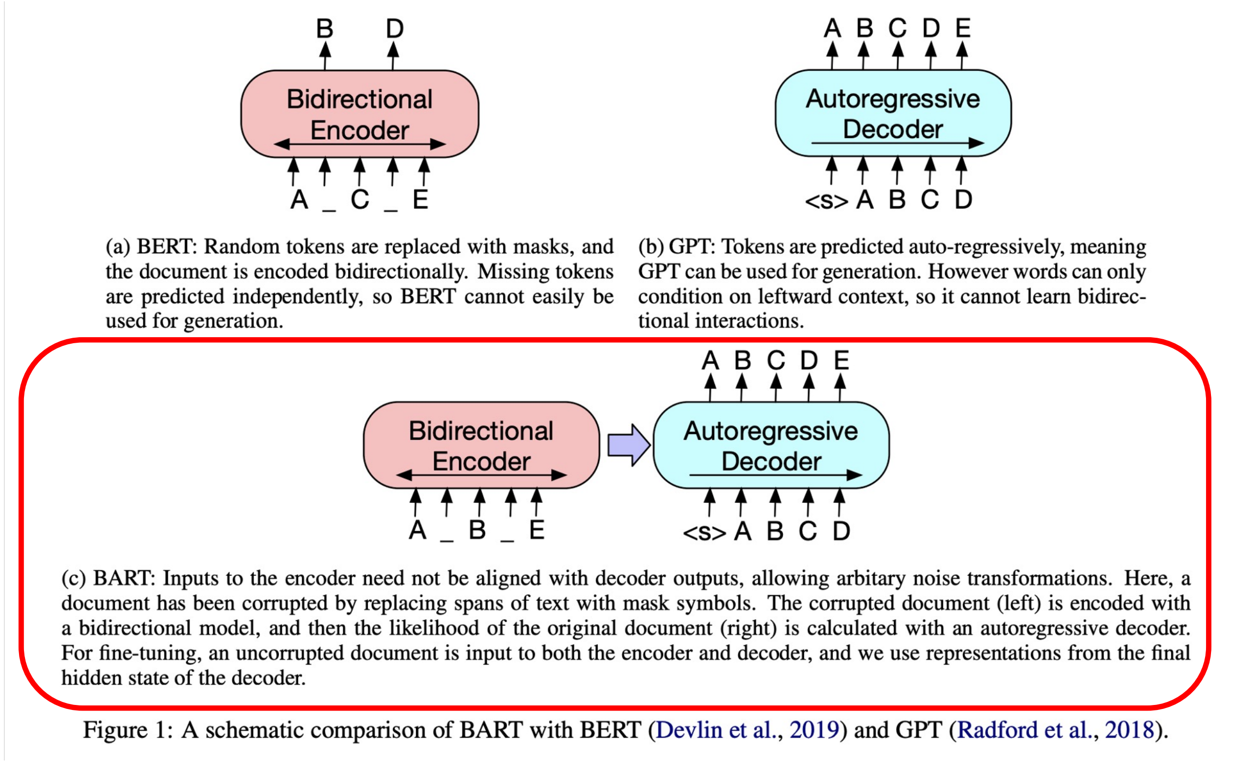
can be used for both classification (GLUE) and generation tasks such as Summarization (CNNDM), Machine Translation (EnDe, EnFr, EnRo).

Objective	GLUE	CNNDM	SQuAD	SGLUE	EnDe	EnFr	EnRo
BERT-style (Devlin et al., 2018)	82.96	19.17	**80.65**	69.85	26.78	**40.03**	27.41
MASS-style (Song et al., 2019)	82.32	19.16	80.10	69.28	26.79	**39.89**	27.55
★ Replace corrupted spans	83.28	**19.24**	**80.88**	**71.36**	**26.98**	39.82	**27.65**
Drop corrupted tokens	**84.44**	**19.31**	**80.52**	68.67	**27.07**	39.76	**27.82**

Table 5: Comparison of variants of the BERT-style pre-training objective. In the first two variants, the model is trained to reconstruct the original uncorrupted text segment. In the latter two, the model only predicts the sequence of corrupted tokens.

BART ([Lewis et al., 2019](#))

BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension.



(a) BERT: Random tokens are replaced with masks, and the document is encoded bidirectionally. Missing tokens are predicted independently, so BERT cannot easily be used for generation.

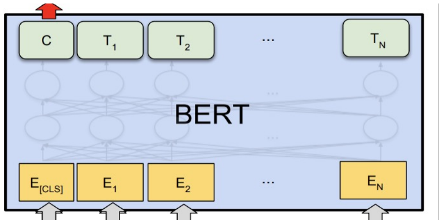
(b) GPT: Tokens are predicted auto-regressively, meaning GPT can be used for generation. However words can only condition on leftward context, so it cannot learn bidirectional interactions.

(c) BART: Inputs to the encoder need not be aligned with decoder outputs, allowing arbitary noise transformations. Here, a document has been corrupted by replacing spans of text with mask symbols. The corrupted document (left) is encoded with a bidirectional model, and then the likelihood of the original document (right) is calculated with an autoregressive decoder. For fine-tuning, an uncorrupted document is input to both the encoder and decoder, and we use representations from the final hidden state of the decoder.

Figure 1: A schematic comparison of BART with BERT (Devlin et al., 2019) and GPT (Radford et al., 2018).

Domain Specific

SciBERT (Beltagy et al., 2020)



SciBERT: A Pretrained Language Model for Scientific Text.

Keep pretraining BERT on in-domain data improves the end task performance.

Field	Task	Dataset	SOTA	BERT-Base		SciBERT	
				Frozen	Finetune	Frozen	Finetune
Bio	NER	BC5CDR (Li et al., 2016)	88.85[7]	85.08	86.72	88.73	**90.01**
		JNLPBA (Collier and Kim, 2004)	**78.58**	74.05	76.09	75.77	77.28
		NCBI-disease (Dogan et al., 2014)	**89.36**	84.06	86.88	86.39	88.57
	PICO	EBM-NLP (Nye et al., 2018)	66.30	61.44	71.53	68.30	**72.28**
	DEP	GENIA (Kim et al., 2003) - LAS	**91.92**	90.22	90.33	90.36	90.43
		GENIA (Kim et al., 2003) - UAS	**92.84**	91.84	91.89	92.00	91.99
	REL	ChemProt (Kringelum et al., 2016)	76.68	68.21	79.14	75.03	**83.64**
CS	NER	SciERC (Luan et al., 2018)	64.20	63.58	65.24	65.77	**67.57**
	REL	SciERC (Luan et al., 2018)	n/a	72.74	78.71	75.25	**79.97**
	CLS	ACL-ARC (Jurgens et al., 2018)	67.9	62.04	63.91	60.74	**70.98**
Multi	CLS	Paper Field	n/a	63.64	65.37	64.38	**65.71**
		SciCite (Cohan et al., 2019)	84.0	84.31	84.85	**85.42**	**85.49**
Average				73.58	77.16	76.01	79.27

Table 1: Test performances of all BERT variants on all tasks and datasets. **Bold** indicates the SOTA result (multiple

Legal-BERT

- LEGAL-BERT: The Muppets straight out of Law School
  - use the original BERT out of the box,
  - adapt BERT by additional pre-training on domain-specific corpora
  - pre-train BERT from scratch on domain-specific corpora.

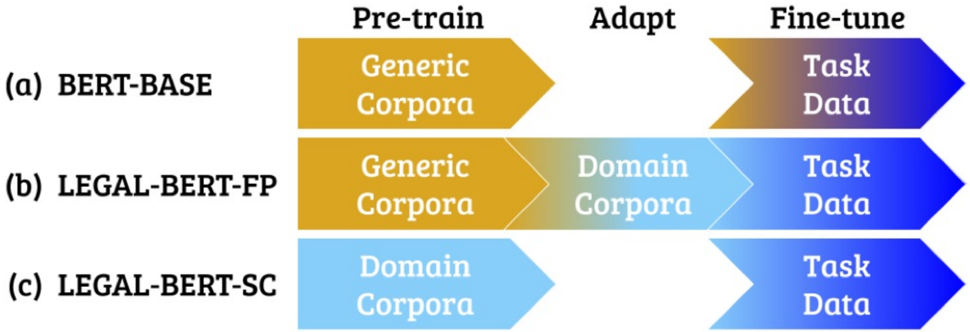


Figure 1: The three alternatives when employing BERT for NLP tasks in specialised domains: (a) use BERT out of the box, (b) further pre-train BERT (FP), and (c) pre-train BERT from scratch (SC). All strategies have a final fine-tuning step.

Legal-BERT

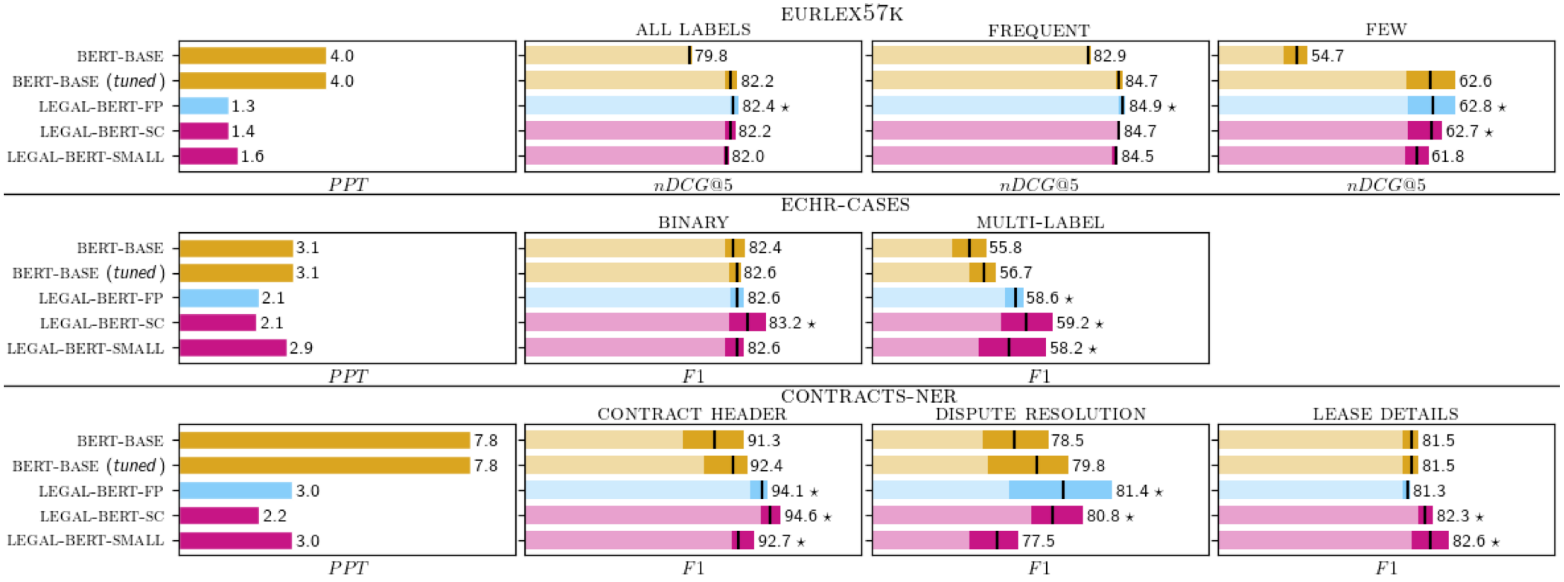
- 

Figure 4: Perplexities (*PPT*) and end-task results on test data across all datasets and all models considered. The reported results are averages over multiple runs also indicated by a vertical black line in each bar. The transparent and opaque parts of each bar show the minimum and maximum scores of the runs, respectively. A star indicates versions of LEGAL-BERT that perform better on average than the tuned BERT-BASE.

REALM (Guu et al., 2020)

REALM: **Retrieval-Augmented** Language Model Pre-Training

These pre-trained models, such as BERT and RoBERTa, have been shown to *memorize a surprising amount of world knowledge*, such as “the birthplace of Francesco Bartolomeo Conti”, “the developer of JDK” and “the owner of Border TV”. … these models memorize knowledge *implicitly* — i.e., world knowledge is captured in an abstract way in the model weights …

Instead, what if there was a method for pre-training that could access knowledge *explicitly*, e.g., by referencing an additional large external text corpus, in order to achieve accurate results without increasing the model size or complexity?

https://ai.googleblog.com/2020/08/realm-integrating-retrieval-into.html

REALM (Guu et al., 2020)

Open-Domain Question Answering

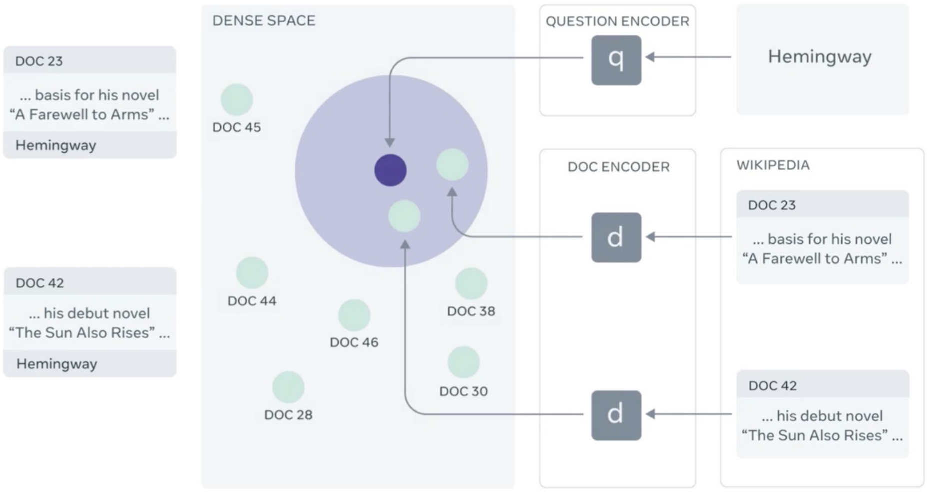
Table 1. Test results on Open-QA benchmarks. The number of train/test examples are shown in paretheses below each benchmark. Predictions are evaluated with exact match against any reference answer. Sparse retrieval denotes methods that use sparse features such as TF-IDF and BM25. Our model, REALM, outperforms all existing systems.

Name	**Architectures**	**Pre-training**	**NQ** (79k/4k)	**WQ** (3k/2k)	**CT** (1k /1k)	**# params**
BERT-Baseline (Lee et al., 2019)	Sparse Retr.+Transformer	BERT	26.5	17.7	21.3	110m
T5 (base) (Roberts et al., 2020)	Transformer Seq2Seq	T5 (Multitask)	27.0	29.1	-	223m
T5 (large) (Roberts et al., 2020)	Transformer Seq2Seq	T5 (Multitask)	29.8	32.2	-	738m
T5 (11b) (Roberts et al., 2020)	Transformer Seq2Seq	T5 (Multitask)	34.5	37.4	-	11318m
DrQA (Chen et al., 2017)	Sparse Retr.+DocReader	N/A	-	20.7	25.7	34m
HardEM (Min et al., 2019a)	Sparse Retr.+Transformer	BERT	28.1	-	-	110m
GraphRetriever (Min et al., 2019b)	GraphRetriever+Transformer	BERT	31.8	31.6	-	110m
PathRetriever (Asai et al., 2019)	PathRetriever+Transformer	MLM	32.6	-	-	110m
ORQA (Lee et al., 2019)	Dense Retr.+Transformer	ICT+BERT	33.3	36.4	30.1	330m
Ours ($\mathcal{X}$ = Wikipedia, $\mathcal{Z}$ = Wikipedia)	Dense Retr.+Transformer	REALM	39.2	40.2	**46.8**	330m
Ours ($\mathcal{X}$ = CC-News, $\mathcal{Z}$ = Wikipedia)	Dense Retr.+Transformer	REALM	**40.4**	**40.7**	42.9	330m

RAG (Lewis et al., 2020)

RAG: Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks

REALM is mostly used for short-span QA, but RAG can be used for generation-based QA.



https://ai.facebook.com/blog/retrieval-augmented-generation-streamlining-the-creation-of-intelligent-natural-language-processing-models/

RAG (Lewis et al., 2020)

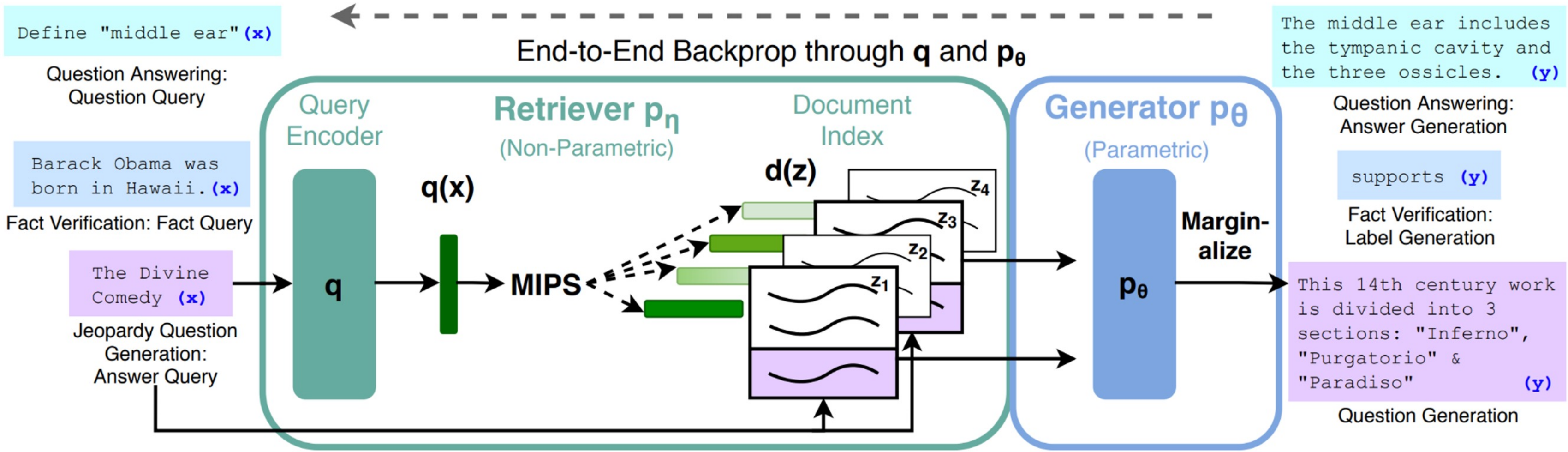


Figure 1: Overview of our approach. We combine a pre-trained retriever (*Query Encoder* + *Document Index*) with a pre-trained seq2seq model (*Generator*) and fine-tune end-to-end. For query x, we use Maximum Inner Product Search (MIPS) to find the top-K documents z_i. For final prediction y, we treat z as a latent variable and marginalize over seq2seq predictions given different documents.

ALBERT (Lan et al. 2019)

ALBERT: A Lite BERT for Self-supervised Learning of Language Representations

Can we have a smaller model but with equal performance?
Two techniques to reduce the number of parameters.

- Factorized embedding: Decompose the large vocabulary embedding matrix into two small matrices.
- Cross-layer parameter sharing.

Model		Parameters	Layers	Hidden	Embedding	Parameter-sharing
BERT	base	108M	12	768	768	False
	large	334M	24	1024	1024	False
ALBERT	base	12M	12	768	128	True
	large	18M	24	1024	128	True
	xlarge	60M	24	2048	128	True
	xxlarge	235M	12	4096	128	True

Table 1: The configurations of the main BERT and ALBERT models analyzed in this paper.

DistilBERT (Sanh et al. 2019)

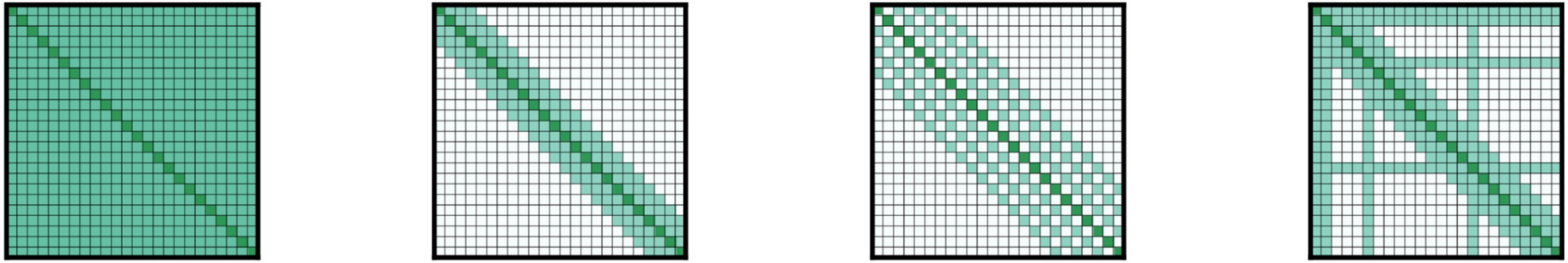
DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter

Use knowledge **distillation** during the pre-training phase.

Knowledge distillation [Bucila et al., 2006, Hinton et al., 2015] is a compression technique in which a compact model - **the student** - is trained to reproduce the behaviour of a larger model - **the teacher** - or an ensemble of models.

Longformer (Beltagy et al., 2020)

Longformer: The Long-Document Transformer

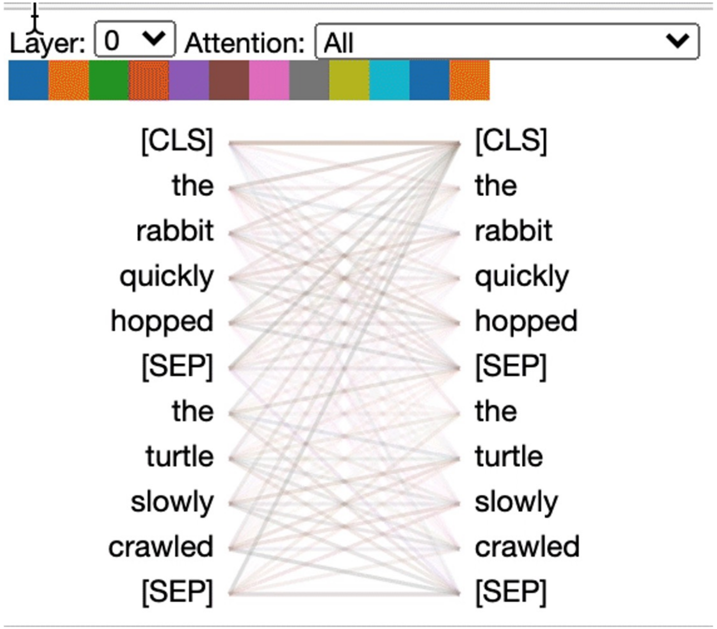


(a) Full n^2 attention (b) Sliding window attention (c) Dilated sliding window (d) Global+sliding window

Figure 2: Comparing the full self-attention pattern and the configuration of attention patterns in our Longformer.

Analysis on BERT

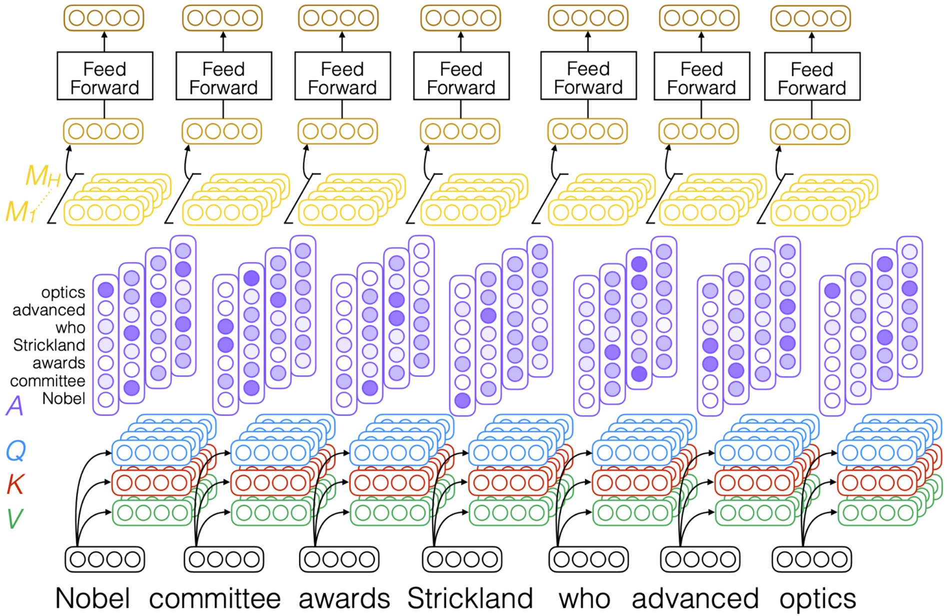
BertViz is an interactive tool for visualizing attention in Transformer language models such as BERT, GPT-2, or T5. Check out its code and demo here: https://github.com/jessevig/bertviz



Analysis on BERT

What Does BERT Look At? An Analysis of BERT's Attention (Clark et al., 2019)

Multiple Heads



Analysis on BERT

[What Does BERT Look At? An Analysis of BERT's Attention](#) (Clark et al., 2019)

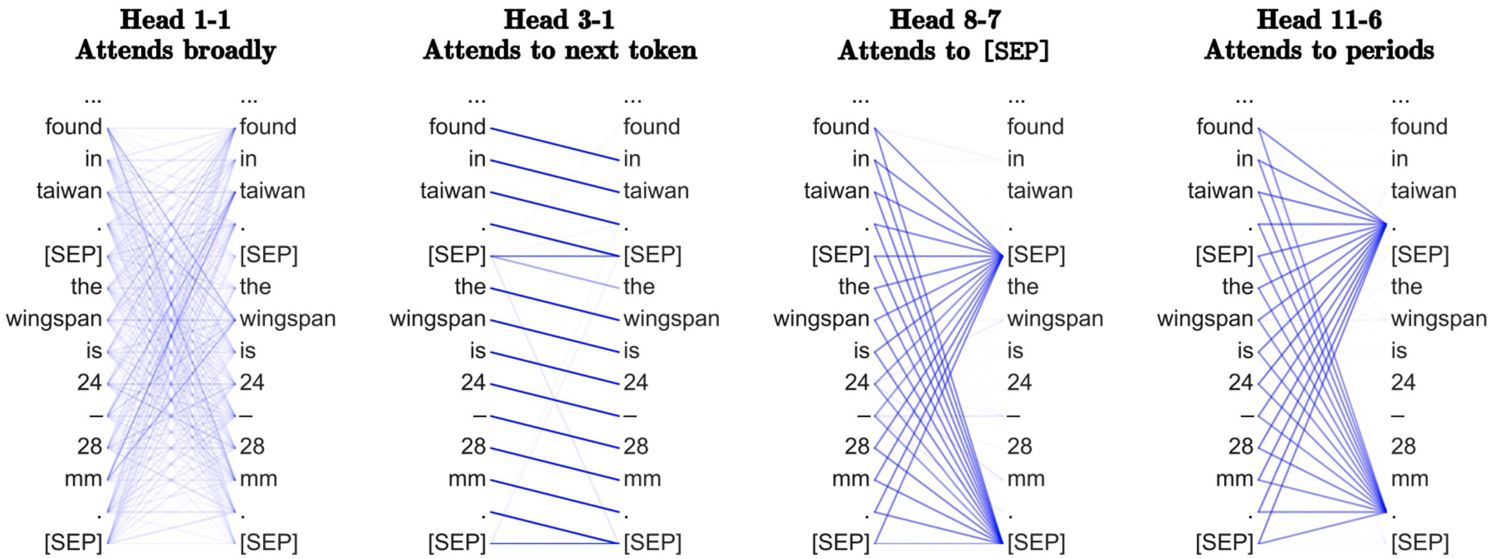


Figure 1: Examples of heads exhibiting the patterns discussed in Section 3. The darkness of a line indicates the strength of the attention weight (some attention weights are so low they are invisible).

Analysis on BERT

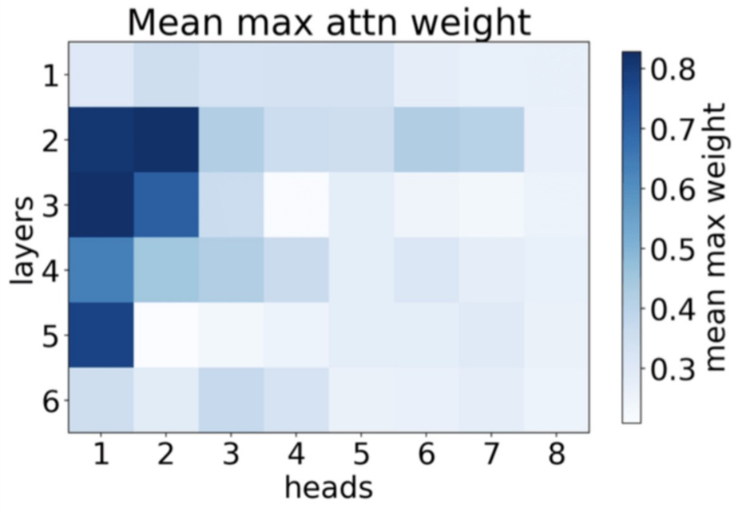
Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned (Voita et al., 2019)

- Analyze Multi-head self-attention in Neural Machine Translation
- Most important and confident heads play consistent and often linguistically-interpretable roles.
- Can **prune** less important heads.

Analysis on BERT

[Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned]() (Voita et al., 2019)

- Heads have different "confidence", measured as average of its maximum attention weight.

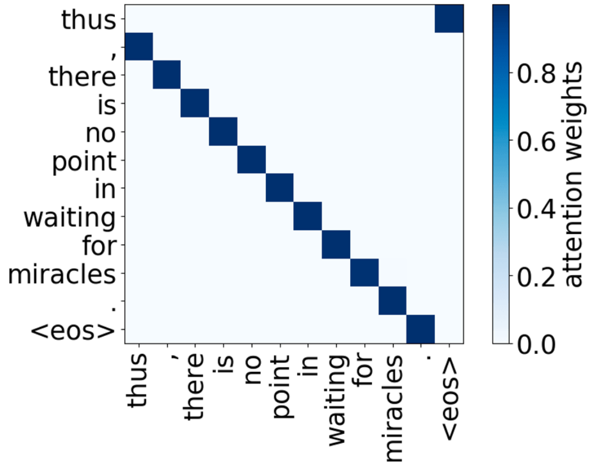


confidence

Analysis on BERT

[Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned]() (Voita et al., 2019)

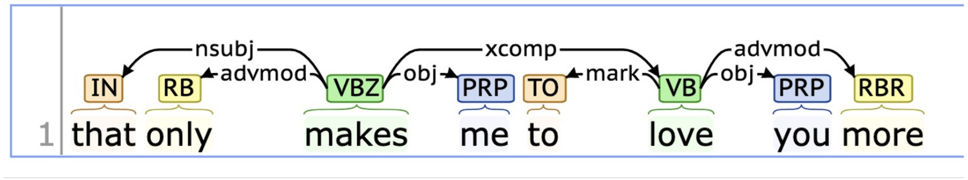
- There are different types of heads.
- **Positional heads**: We refer to a head as “positional” if at least 90% of the time its maximum attention weight is assigned to a specific relative position (in practice either -1 or +1, i.e. attention to adjacent tokens).



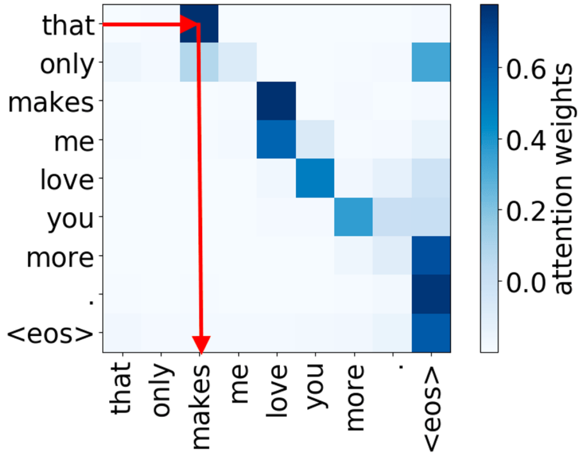
Analysis on BERT

[Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned]() (Voita et al., 2019)

- There are different types of heads.
- **Syntactic heads:** comparing its attention weights to a predicted dependency structure generated using CoreNLP.



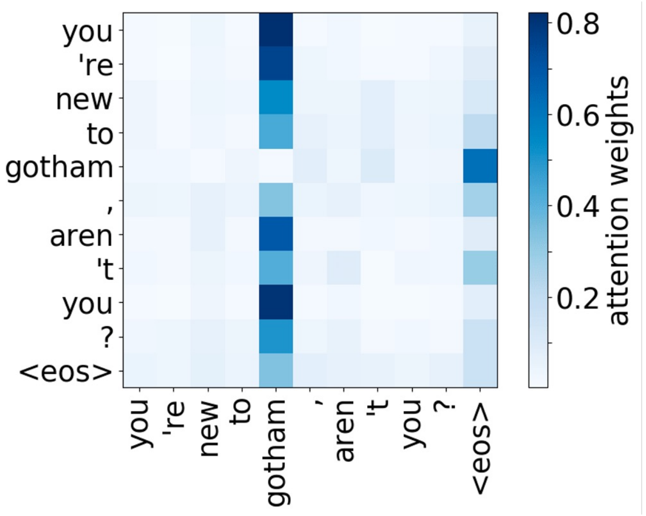
https://corenlp.run/



Analysis on BERT

[Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned]() (Voita et al., 2019)

- There are different types of heads.
- **Rare tokens**: For all models, we find a head pointing to the least frequent tokens in a sentence.



Analysis on BERT

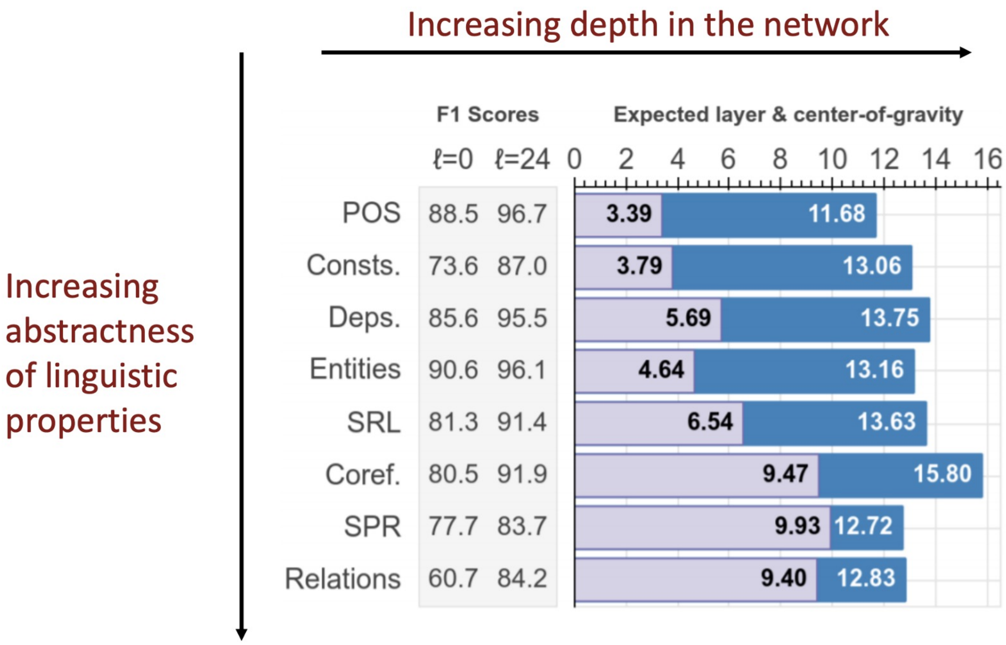
Are Sixteen Heads Really Better than One? (Michel, et al., 2019)

- Make the surprising observation that even if models have been trained using multiple heads, in practice, **a large percentage of attention heads can be removed** at test time without significantly impacting performance.
- In fact, some layers can even be reduced to a **single head.**

Analysis on BERT

BERT Rediscovers the Classical NLP Pipeline (Tenney et al., 2019)

- Quantify where linguistic information is captured within the network.
- Find that the model represents the steps of the traditional NLP pipeline in an interpretable and localizable way, and that the regions responsible for each step appear in the expected sequence: POS tagging, parsing, NER, semantic roles, then coreference



Analysis on BERT

BERT Rediscovers the Classical NLP Pipeline (Tenney et al., 2019)

- Quantify where linguistic information is captured within the network.
- Edge **Probing**. Our experiments are based on the “edge probing” approach of Tenney et al. (2019), which aims to measure how well information about linguistic structure can be extracted from a pre-trained encoder.

- Edge probing decomposes structured-prediction tasks into a common, format, where a **probing classifier receives spans s1 and s2 = [i2, j2) and must predict a label such as a relation type**.

Probing: Supervised Analysis of Neural Networks

Linguistic Knowledge and Transferability of Contextual Representations (Liu et al., 2019)

A probe, i.e. a classifier trained to predict the property from the representations.

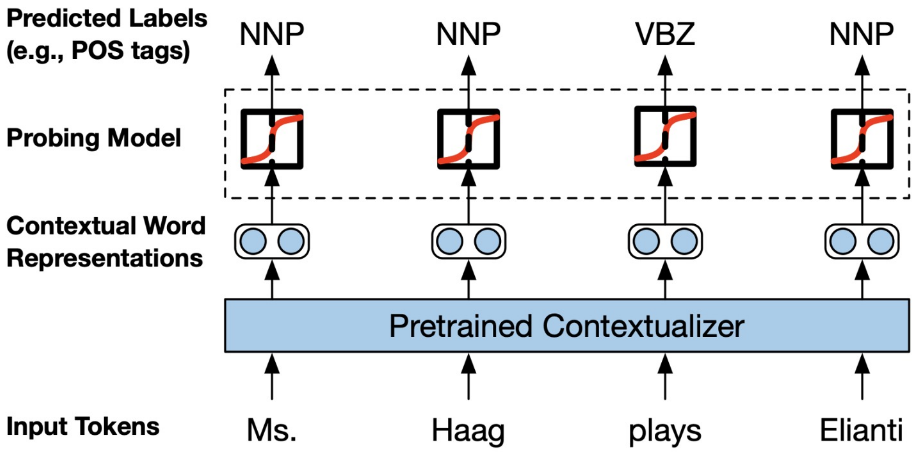


Figure 1: An illustration of the probing model setup used to study the linguistic knowledge within contextual word representations.

BERT Code Demo

Hugging Face Transformers library
Hugging Face course

Code Demo

- BERT for predicting masked tokens.
- BART for summarization
- GPT-2 for text generation.

Google's PaLM

Pathways Language Model (PaLM): Scaling to 540 Billion Parameters for

Breakthrough Performance

OPT-3: Facebook's GPT-3