

CS224N : BERT and Other Pre-trained Language Models

History of Contextual Representations

Pre-training in NLP

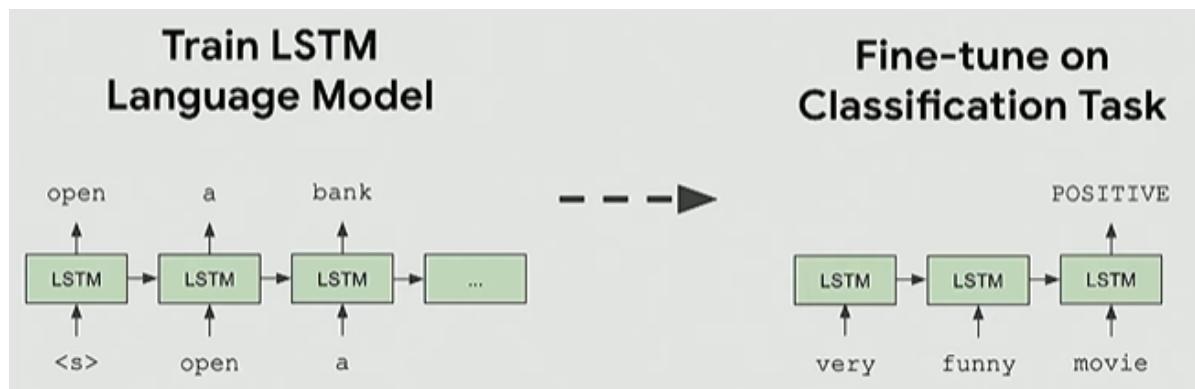
- Word embeddings are the basis of deep learning for NLP
- Word embeddings(word2vec, GloVe) are often pre-trained on text corpus from co-occurrence statistics

Contextual Representations

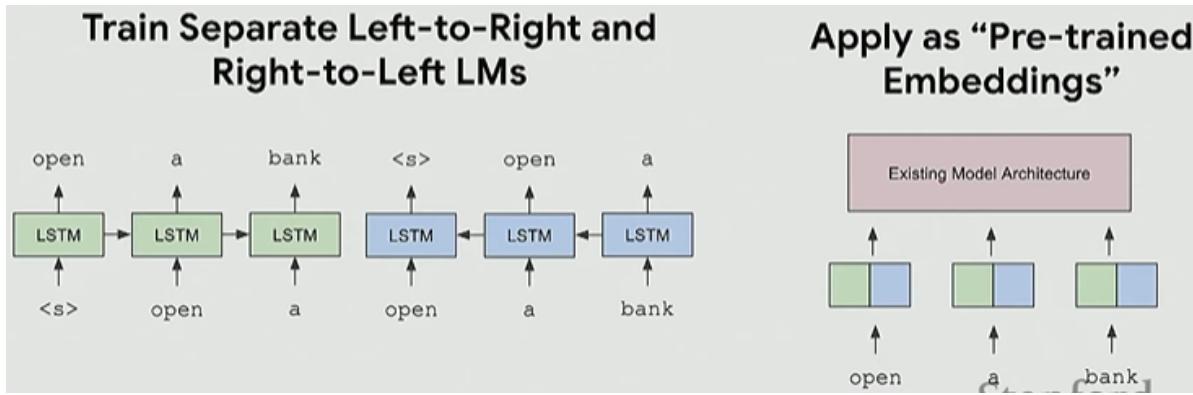
- Problem: Word embeddings are applied in a context free manner
- Solution: Train contextual representations on text corpus

History of Contextual Representations

- Semi-Supervised Sequence Learning, Google, 2015



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



Model Architecture

Transformer encoder

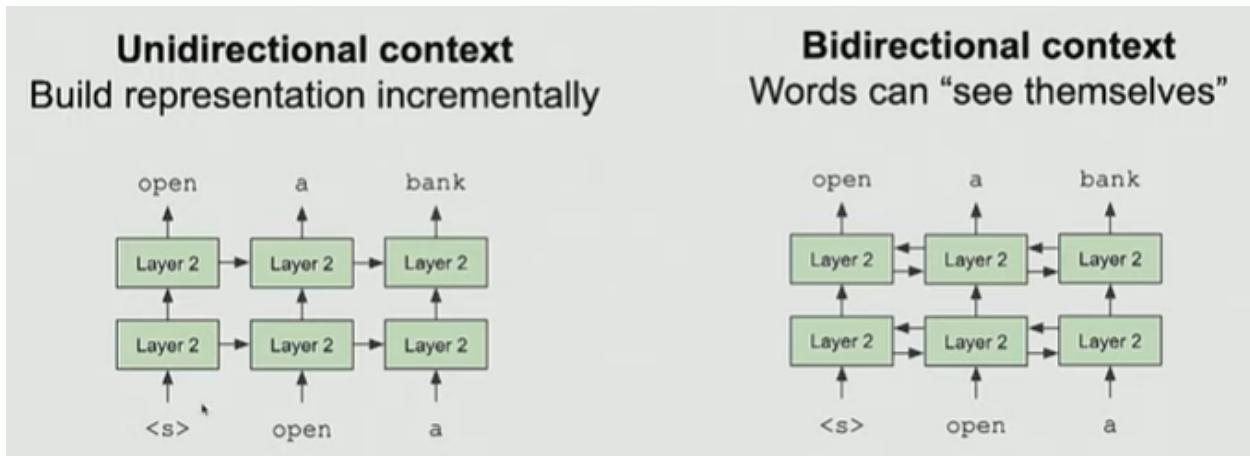
- Multi-headed self attention
 - Models context
- Feed-forward layers
 - Computes non-linear hierarchical features
- Layer norm and residuals
 - Makes training deep networks healthy
- Positional embeddings
 - Allow model to learn relative positioning
- Empirical advantages of Transformer vs. LSTM:
 1. Self-attention == no locality bias
 - Long-distance context has “equal opportunity”
 2. Single multiplication per layer == efficiency of TPU
 - Effective batch size is number of words, not sequences

BERT

Problem with Previous Methods

- Problem: Language models only use left context or right context, but language understanding is bidirectional

Unidirectional vs. Bidirectional Models



Masked LM

- Solution: Mask out $k\%$ of the input words, and then predict the masked words
- Too little masking: Too expensive to train
- Too much masking: Not enough context
- Problem: Mask token never seen at fine-tuning
- Solution: 15% of the words to predict, but don't replace with [MASK] 100% of the time.

Instead:

- 80% of the time, replace with [MASK]
- 10% of the time, replace random word
- 10% of the time, keep same

Next Sentence Prediction

- To learn relationships between sentences, predict whether Sentence B is actual sentence that proceeds Sentence A, or a random sentence

Input Representation

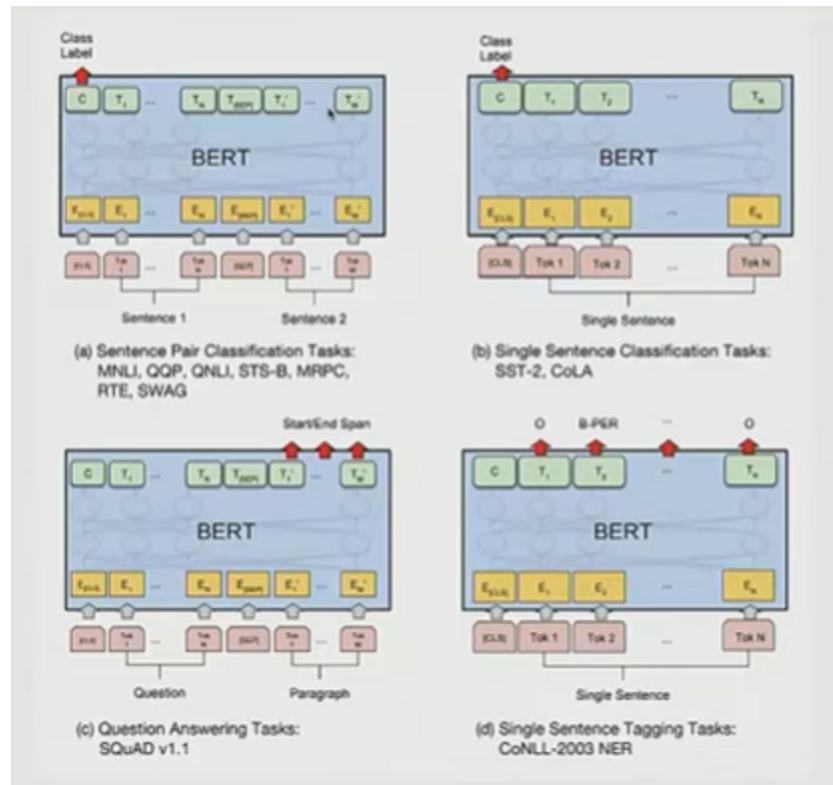
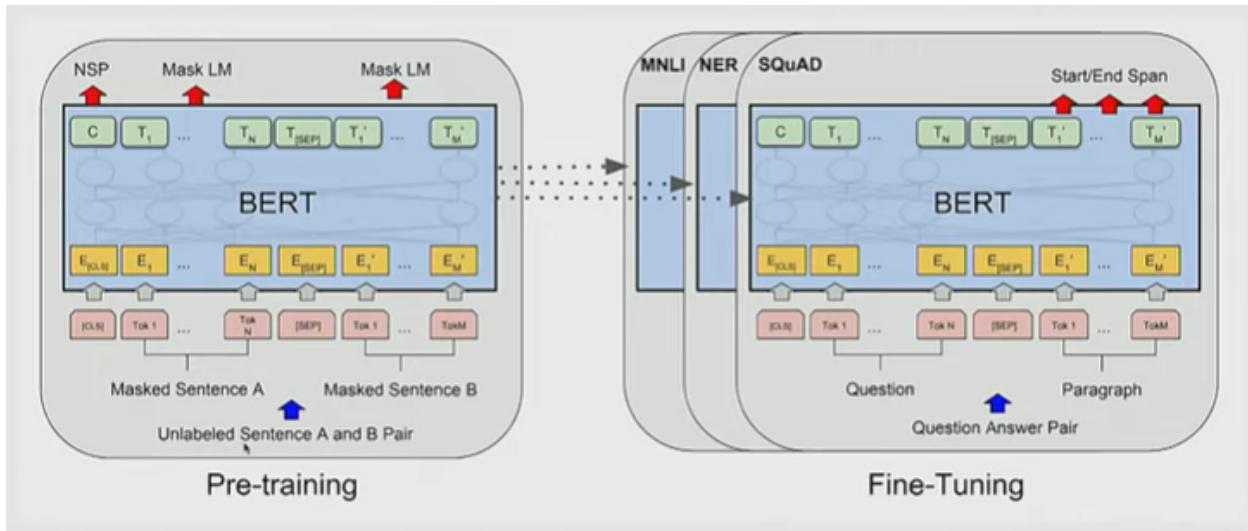
- Use 30,000 WordPiece vocabulary on input
- Each token is sum of three embeddings
- Single sequence is much more efficient

Model Details

- Data: Wikipedia+BookCorpus
- Batch Size: 131,072 words
- Training Time: 1M steps
- Optimizer: AdamW, 1e-4 learning rate, linear decay
- BERT-Base: 12-layer, 768-hidden, 12-head
- BERT-Large: 24-layer, 1024-hidden, 16-head
- Trained on 4x4 or 8x8 TPU slice for 4 days

Fine-Tuning Procedure

Fine-Tuning Procedure



GLUE Results

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	Average -
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

MultiNLI

Premise: Hills and mountains are especially sanctified in Jainism.
Hypothesis: Jainism hates nature.
Label: Contradiction

CoLa

Sentence: The wagon rumbled down the road.
Label: Acceptable

Sentence: The car honked down the road.
Label: Unacceptable

Stanford

SQuAD 2.0

What action did the US begin that started the second oil shock?
 Ground Truth Answers: <No Answer>
 Prediction: <No Answer>

The 1973 oil crisis began in October 1973 when the members of the Organization of Arab Petroleum Exporting Countries (OAPEC, consisting of the Arab members of OPEC plus Egypt and Syria) proclaimed an oil embargo. By the end of the embargo in March 1974, the price of oil had risen from US\$3 per barrel to nearly \$12 globally; US prices were significantly higher. The embargo caused an oil crisis, or "shock", with many short- and long-term effects on global politics and the global economy. It was later called the "[first oil shock](#)", followed by the 1979 oil crisis, termed the "second oil shock".

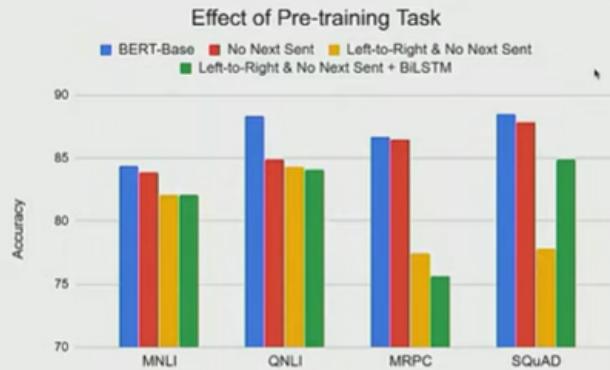
Rank	Model	EM	F1
	Human Performance Stanford University (Rajpurkar & Jia et al. '18)	86.831	89.452
12 Nov 08, 2018	BERT (single model) Google AI Language	80.005	83.061
20 Sep 13, 2018	nlnet (single model) Microsoft Research Asia	74.272	77.052

- Use token 0 ([CLS]) to emit logit for “no answer”.
- “No answer” directly competes with answer span.
- Threshold is optimized on dev set.

Stanford

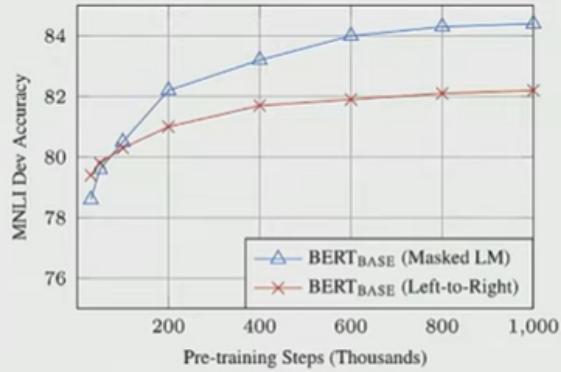
Effect of Pre-training Task

Effect of Pre-training Task



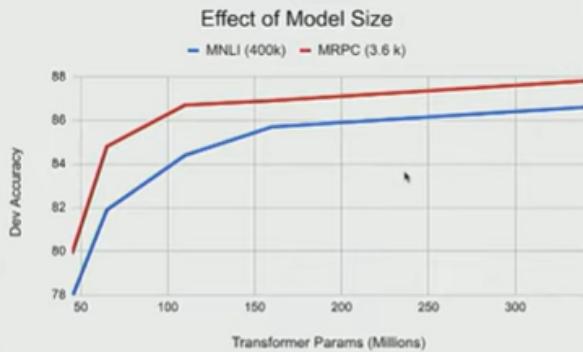
- Masked LM (compared to left-to-right LM) is very important on some tasks, Next Sentence Prediction is important on other tasks.
- Left-to-right model does very poorly on word-level task (SQuAD), although this is mitigated by BiLSTM

Effect of Directionality and Training Time



- Masked LM takes slightly longer to converge because we only predict 15% instead of 100%
- But absolute results are much better almost immediately

Effect of Model Size



- Big models help a lot
- Going from 110M → 340M params helps even on datasets with 3,600 labeled examples
- Improvements have *not* asymptoted

Stanford

Open Source Release

- One reason for BERT's success was the open source release
 - Minimal release (not part of a larger codebase)
 - No dependencies but TensorFlow (or PyTorch)
 - Abstracted so people could include a single file to use model
 - End-to-end push-button examples to train SOTA models
 - Thorough README
 - Idiomatic code
 - Well-documented code
 - Good support (for the first few months)

Stanford

Post-BERT Pre-training Advancements

RoBERTA

RoBERTA

- *RoBERTa: A Robustly Optimized BERT Pretraining Approach* (Liu et al, University of Washington and Facebook, 2019)
- Trained BERT for more epochs and/or on more data
 - Showed that more epochs alone helps, even on same data
 - More data also helps
- Improved masking and pre-training data slightly

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

XLNet

- *XLNet: Generalized Autoregressive Pretraining for Language Understanding* (Yang et al, CMU and Google, 2019)
- Innovation #1: Relative position embeddings
 - Sentence: John ate a hot dog
 - Absolute attention: “How much should dog attend to hot (in any position), and how much should dog in position 4 attend to the word in position 3? (Or 508 attend to 507, ...)”
 - Relative attention: “How much should dog attend to hot (in any position) and how much should dog attend to the previous word?”

Stanford

- Innovation #2: Permutation Language Modeling
 - In a left-to-right language model, every word is predicted based on all of the words to its left
 - Instead: Randomly permute the order for every *training sentence*
 - Equivalent to masking, but many more predictions per sentence
 - Can be done efficiently with Transformers

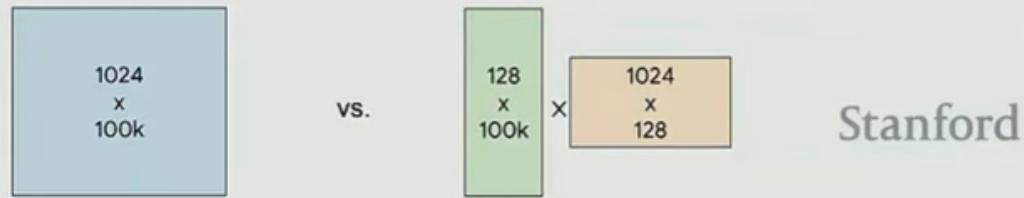
- Also used more data and bigger models, but showed that innovations improved on BERT even with same data and model size
- XLNet results:

Model	MNLI	QNLI	QQP	RTE	SST-2	MRPC	CoLA	STS-B
<i>Single-task single models on dev</i>								
BERT [2]	86.6/-	92.3	91.3	70.4	93.2	88.0	60.6	90.0
RoBERTa [21]	90.2/90.2	94.7	92.2	86.6	96.4	90.9	68.0	92.4
XLNet	90.8/90.8	94.9	92.3	85.9	97.0	90.8	69.0	92.5

ALBERT

ALBERT

- *ALBERT: A Lite BERT for Self-supervised Learning of Language Representations* (Lan et al, Google and TTI Chicago, 2019)
- Innovation #1: Factorized embedding parameterization
 - Use small embedding size (e.g., 128) and then project it to Transformer hidden size (e.g., 1024) with parameter matrix



- Innovation #2: Cross-layer parameter sharing
 - Share all parameters between Transformer layers
- Results:

Models	MNLI	QNLI	QQP	RTE	SST	MRPC	CoLA	STS
<i>Single-task single models on dev</i>								
BERT-large	86.6	92.3	91.3	70.4	93.2	88.0	60.6	90.0
XLNet-large	89.8	93.9	91.8	83.8	95.6	89.2	63.6	91.8
RoBERTa-large	90.2	94.7	92.2	86.6	96.4	90.9	68.0	92.4
ALBERT (1M)	90.4	95.2	92.0	88.1	96.8	90.2	68.7	92.7
ALBERT (1.5M)	90.8	95.3	92.2	89.2	96.9	90.9	71.4	93.0

- ALBERT is light in terms of *parameters*, not speed

Model	Parameters	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg	Speedup
BERT	base	108M	90.4/83.2	80.4/77.6	84.5	92.8	68.2	82.3
	large	334M	92.2/85.5	85.0/82.2	86.6	93.0	73.9	85.2
ALBERT	base	12M	89.3/82.3	80.0/77.1	81.6	90.3	64.0	80.1
	large	18M	90.6/83.9	82.3/79.4	83.5	91.7	68.5	82.4
	xlarge	60M	92.5/86.1	86.1/83.1	86.4	92.4	74.8	85.5
	xxlarge	235M	94.1/88.3	88.1/85.1	88.0	95.2	82.3	88.7

T5

T5

- *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer* (Raffel et al, Google, 2019)
- Ablated many aspects of pre-training:
 - Model size
 - Amount of training data
 - Domain/cleanness of training data
 - Pre-training objective details (e.g., span length of masked text)
 - Ensembling
 - Finetuning recipe (e.g., only allowing certain layers to finetune)
 - Multi-task training

Stanford

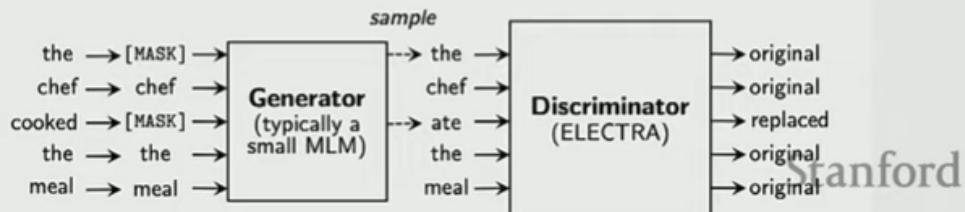
- Conclusions:
 - Scaling up model size and amount of training data helps a lot
 - Best model is 11B parameters (BERT-Large is 330M), trained on 120B words of cleaned common crawl text
 - Exact masking/corruptions strategy doesn't matter that much
 - Mostly negative results for better finetuning and multi-task strategies
- T5 results:

Rank	Name	Model	URL	Score	BoolQ	CB	COPA	Mnli/Mnli-mm	Rte/Rd	Rte	Wic	Wsc	Ax-b	Ax-q	
1	SuperGLUE Human Baselines	SuperGLUE Human Baselines	🔗	89.8	89.0	95.8/98.9	100.0	81.8/51.9	91.7/91.3	93.6	80.0	100.0	76.6	99.3/99.7	
+	2	T5 Team - Google	T5	🔗	89.3	91.2	93.5/96.8	94.8	88.1/63.3	94.1/93.4	92.5	76.9	93.8	65.6	92.7/91.9
3	Zhuyin Technology	RoBERTa-mli-adv		85.7	87.1	92.4/95.6	91.2	85.1/54.3	91.7/91.3	88.1	72.1	91.8	58.5	91.0/78.1	
4	Facebook AI	RoBERTa	🔗	84.6	87.1	90.5/95.2	90.6	84.6/52.5	90.6/90.0	88.2	65.9	89.0	57.9	91.0/78.1	
5	IBM Research AI	BERT-mli		73.5	84.8	89.6/94.0	73.8	73.2/30.5	74.6/74.0	84.1	46.2	51.0	31.6	97.8/97.3	
6	SuperGLUE Baselines	BERT++	🔗	71.5	79.0	84.8/90.4	73.8	70.0/24.1	72.0/71.3	79.0	63.6	64.4	38.0	99.4/91.4	
		BERT	🔗	69.0	77.4	75.7/83.6	70.6	70.0/24.1	72.0/71.3	71.7	69.6	64.4	23.0	97.8/91.7	

ELECTRA

ELECTRA

- ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators (Clark et al, 2020)
- Train model to discriminate locally plausible text from real text



- Difficult to match SOTA results with less compute

Model	Train FLOPs	Params	SQuAD 1.1		SQuAD 2.0	
			EM	F1	EM	F1
BERT-Base	6.4e19 (0.09x)	110M	80.8	88.5	–	–
BERT	1.9e20 (0.27x)	335M	84.1	90.9	79.0	81.8
SpanBERT	7.1e20 (1x)	335M	88.8	94.6	85.7	88.7
XLNet-Base	6.6e19 (0.09x)	117M	81.3	–	78.5	–
XLNet	3.9e21 (5.4x)	360M	89.7	95.1	87.9	90.6
RoBERTa-100K	6.4e20 (0.90x)	356M	–	94.0	–	87.7
RoBERTa-500K	3.2e21 (4.5x)	356M	88.9	94.6	86.5	89.4
ALBERT	3.1e22 (44x)	235M	89.3	94.8	87.4	90.2
BERT (ours)	7.1e20 (1x)	335M	88.0	93.7	84.7	87.5
ELECTRA-Base	6.4e19 (0.09x)	110M	84.5	90.8	80.5	83.3
ELECTRA-400K	7.1e20 (1x)	335M	88.7	94.2	86.9	89.6
ELECTRA-1.75M	3.1e21 (4.4x)	335M	89.7	94.9	88.1	90.6

Distillation

Applying Models to Production Services

- BERT and other pre-trained language models are extremely large and expensive
- How are companies applying them to low-latency production services?

GOOGLE \ TECH \ ARTIFICIAL INTELLIGENCE \

Google is improving 10 percent of searches by understanding language context

Say hello to BERT

By Dieter Braun | @backlon | Oct 23, 2019, 3:01am EDT

Bing says it has been applying BERT since April

The natural language processing capabilities are now applied to all Bing queries globally.

Stanford

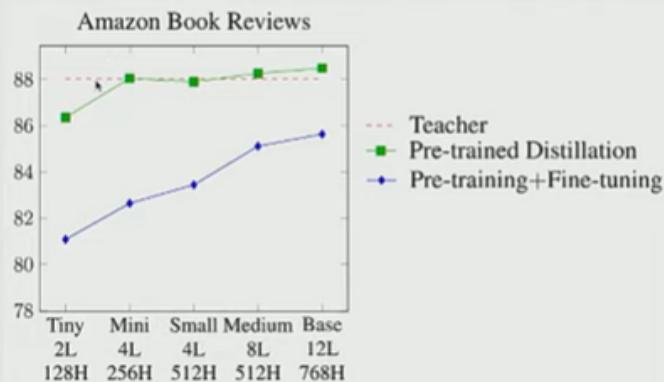
George Nguyen on November 19, 2019 at 1:38 pm

Distillation

- Answer: Distillation (a.k.a., model compression)
- Idea has been around for a long time:
 - *Model Compression* (Bucila et al, 2006)
 - *Distilling the Knowledge in a Neural Network* (Hinton et al, 2015)
- Simple technique:
 - Train “Teacher”: Use SOTA pre-training + fine-tuning technique to train model with maximum accuracy
 - Label a large amount of unlabeled input examples with Teacher
 - Train “Student”: Much smaller model (e.g., 50x smaller) which is trained to mimic Teacher output
 - Student objective is typically Mean Square Error or Cross Entropy

Stanford

- Example distillation results
 - 50k labeled examples, 8M unlabeled examples



- Distillation works *much* better than pre-training + fine-tuning with smaller model

Conclusions

- Pre-trained bidirectional language models work incredibly well
- However, the models are extremely expensive
- Improvements (unfortunately) seem to mostly come from even more expensive models and more data
- The inference/serving problem is mostly “solved” through distillation