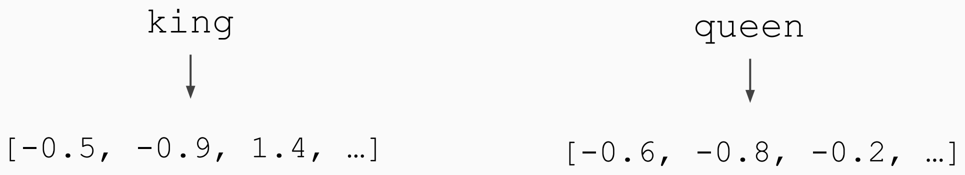
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

(**B**idirectional **E**ncoder **R**epresentations from **T**ransformers)

Jacob Devlin
Google AI Language

Pre-training in NLP

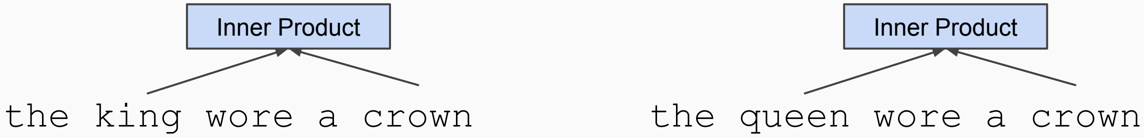
- Word embeddings are the basis of deep learning for NLP



king → [-0.5, -0.9, 1.4, …]

queen → [-0.6, -0.8, -0.2, …]

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



Inner Product

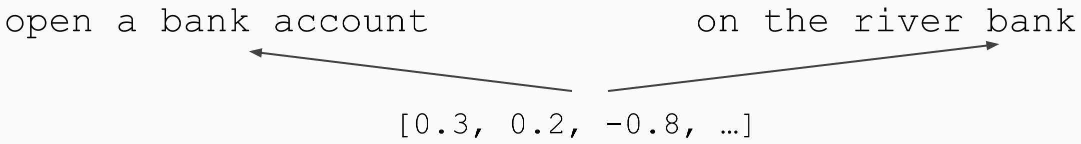
the king wore a crown

Inner Product

the queen wore a crown

Contextual Representations

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

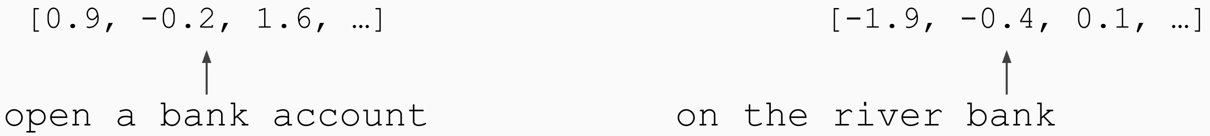


open a bank account

on the river bank

[0.3, 0.2, -0.8, …]

- **Solution**: Train *contextual* representations on text corpus



[0.9, -0.2, 1.6, …]

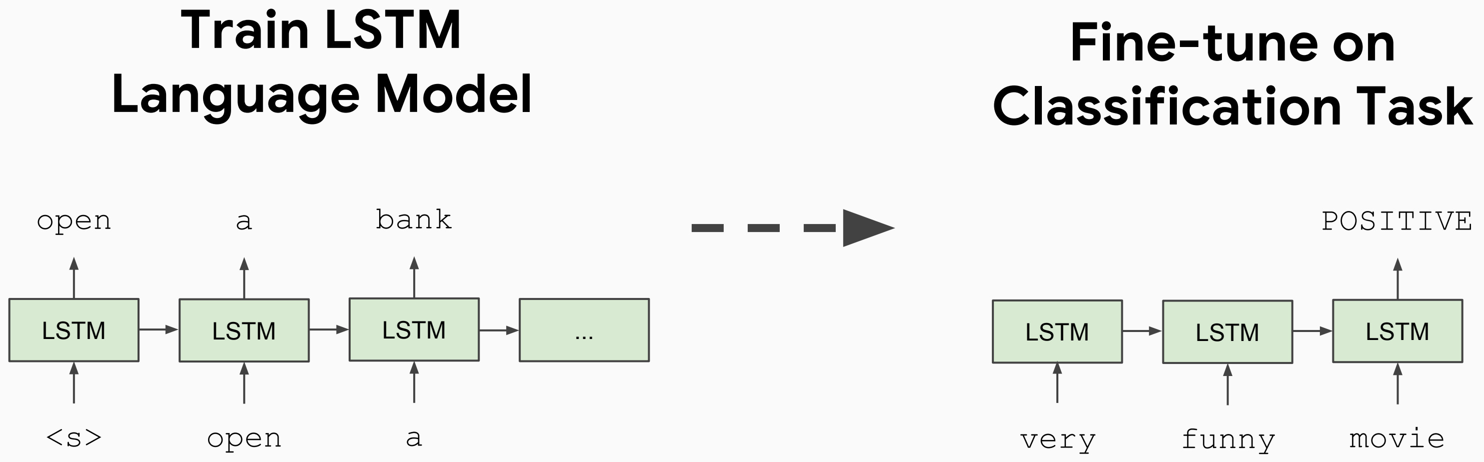
open a bank account

[-1.9, -0.4, 0.1, …]

on the river bank

History of Contextual Representations

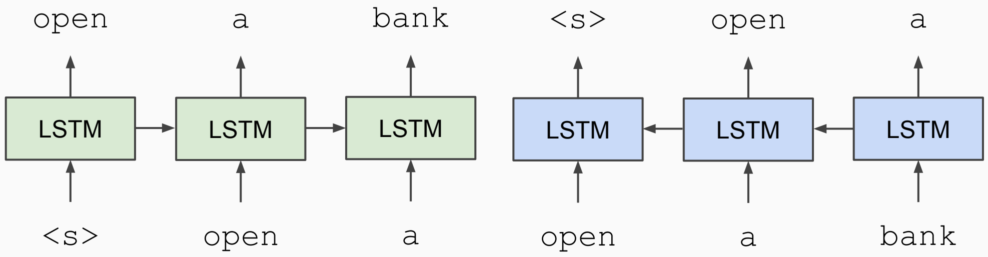
- *Semi-Supervised Sequence Learning*, Google, 2015



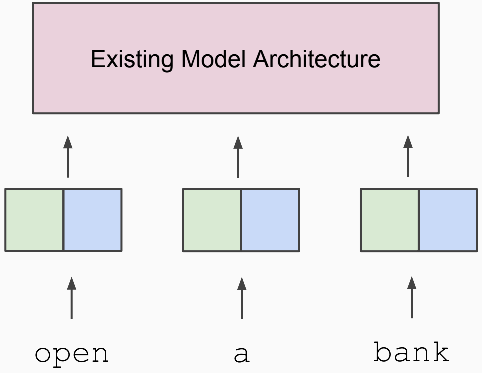
History of Contextual Representations

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

Train Separate Left-to-Right and Right-to-Left LMs



Apply as “Pre-trained Embeddings”

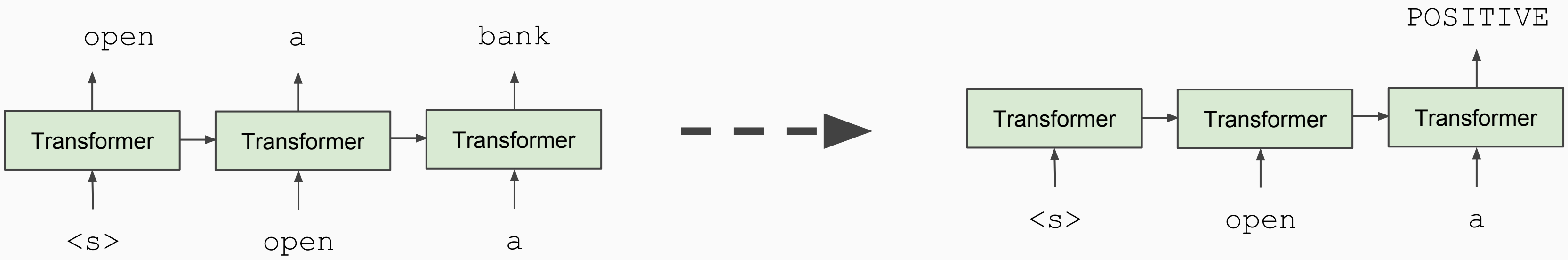


History of Contextual Representations

- *Improving Language Understanding by Generative Pre-Training*, OpenAI, 2018

Train Deep (12-layer) Transformer LM

Fine-tune on Classification Task

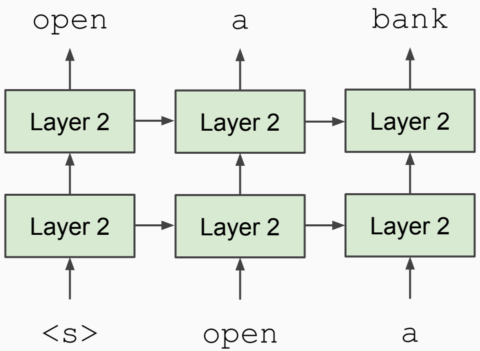


Problem with Previous Methods

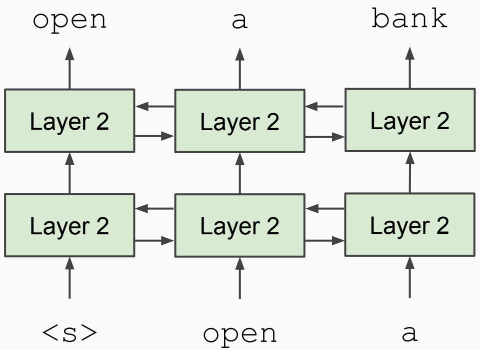
- **Problem**: Language models only use left context *or* right context, but language understanding is bidirectional.
- Why are LMs unidirectional?
- Reason 1: Directionality is needed to generate a well-formed probability distribution.
  - We don’t care about this.
- Reason 2: Words can “see themselves” in a bidirectional encoder.

Unidirectional vs. Bidirectional Models

Unidirectional context
Build representation incrementally

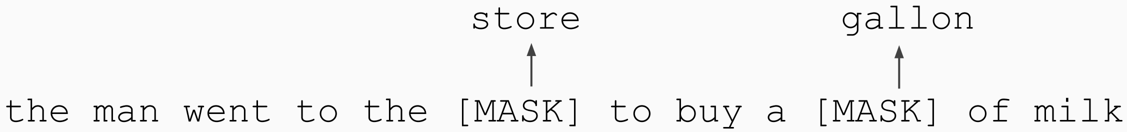


Bidirectional context
Words can “see themselves”



Masked LM

- **Solution**: Mask out *k*% of the input words, and then predict the masked words
  - We always use *k* = 15%



store

gallon

the man went to the [MASK] to buy a [MASK] of milk

- Too little masking: Too expensive to train
- Too much masking: Not enough context

Masked LM

- 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]`

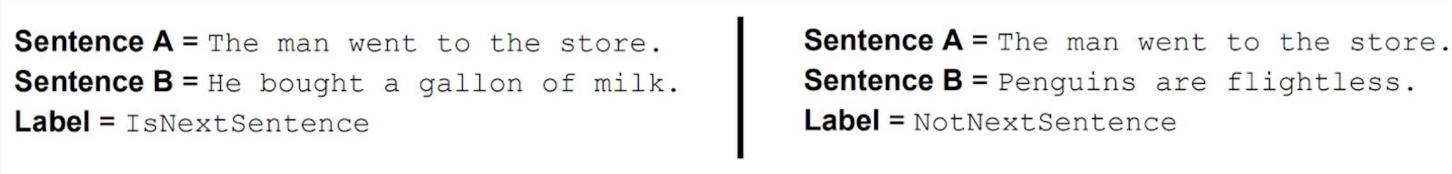
  `went to the store → went to the [MASK]`
- 10% of the time, replace random word

  `went to the store → went to the running`
- 10% of the time, keep same

  `went to the store → went to the store`

Next Sentence Prediction

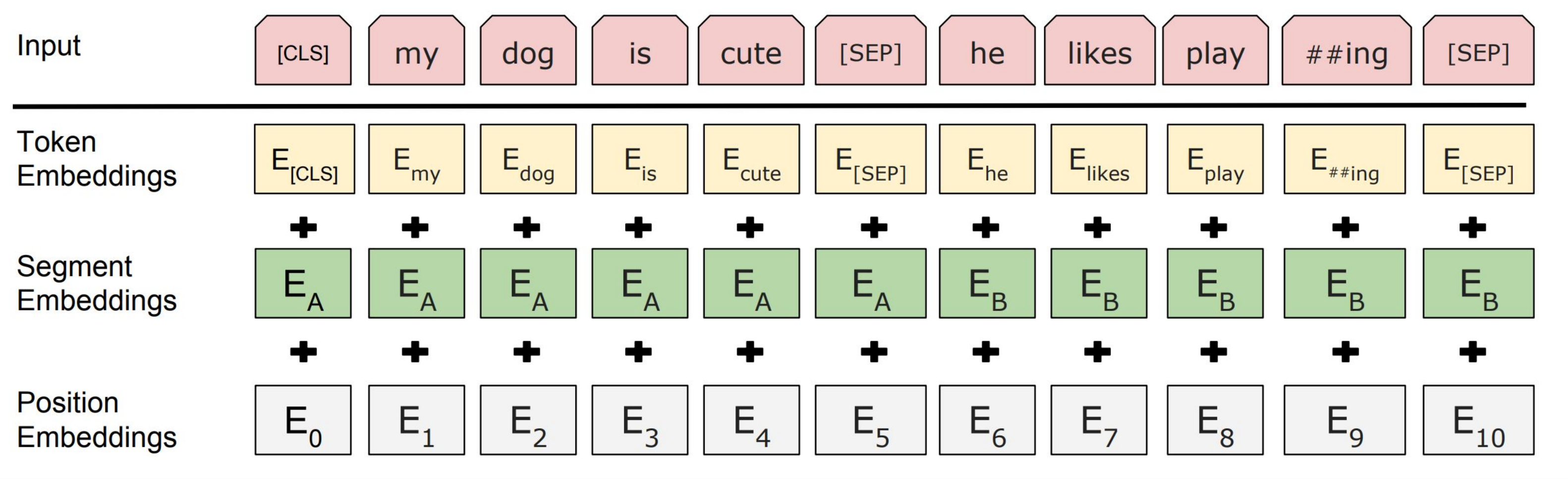
- To learn *relationships* between sentences, predict whether Sentence B is actual sentence that proceeds Sentence A, or a random sentence



Sentence A = `The man went to the store.`
Sentence B = `He bought a gallon of milk.`
Label = `IsNextSentence`

Sentence A = `The man went to the store.`
Sentence B = `Penguins are flightless.`
Label = `NotNextSentence`

Input Representation



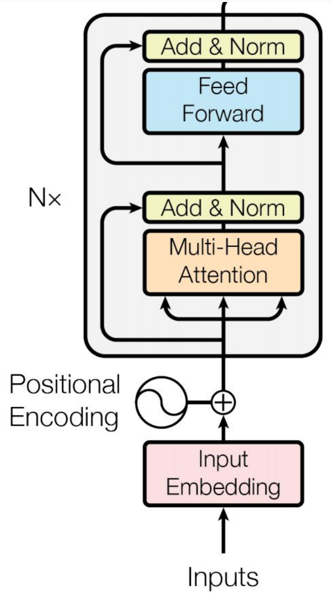
Input	[CLS]	my	dog	is	cute	[SEP]	he	likes	play	##ing	[SEP]
Token Embeddings	$E_{[CLS]}$	E_{my}	E_{dog}	E_{is}	E_{cute}	$E_{[SEP]}$	E_{he}	E_{likes}	E_{play}	$E_{\#\#ing}$	$E_{[SEP]}$
	+	+	+	+	+	+	+	+	+	+	+
Segment Embeddings	E_A	E_A	E_A	E_A	E_A	E_A	E_B	E_B	E_B	E_B	E_B
	+	+	+	+	+	+	+	+	+	+	+
Position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

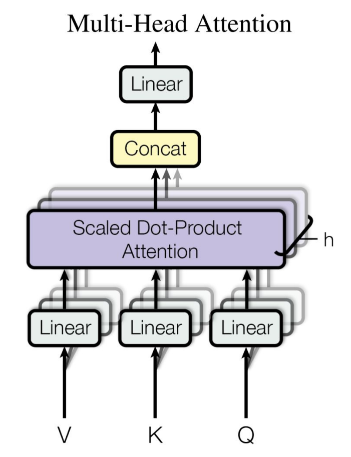
- Use 30,000 WordPiece vocabulary on input.
- Each token is sum of three embeddings
- Single sequence is much more efficient.

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
  - Allows model to learn relative positioning



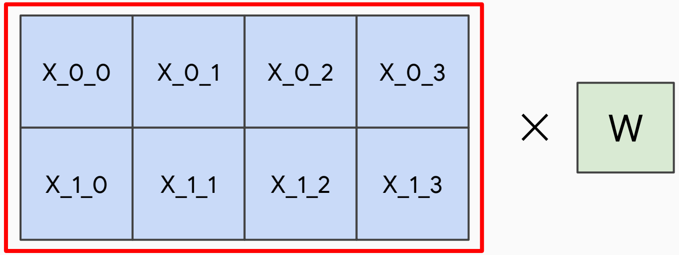


Model Architecture

- Empirical advantages of Transformer vs. LSTM:

1. Self-attention == no locality bias
    - Long-distance context has “equal opportunity”
2. Single multiplication per layer == efficiency on TPU
    - Effective batch size is number of *words*, not *sequences*

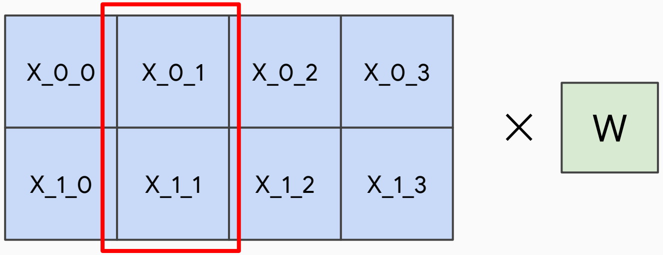
Transformer



X_0_0	X_0_1	X_0_2	X_0_3
X_1_0	X_1_1	X_1_2	X_1_3

× W

LSTM



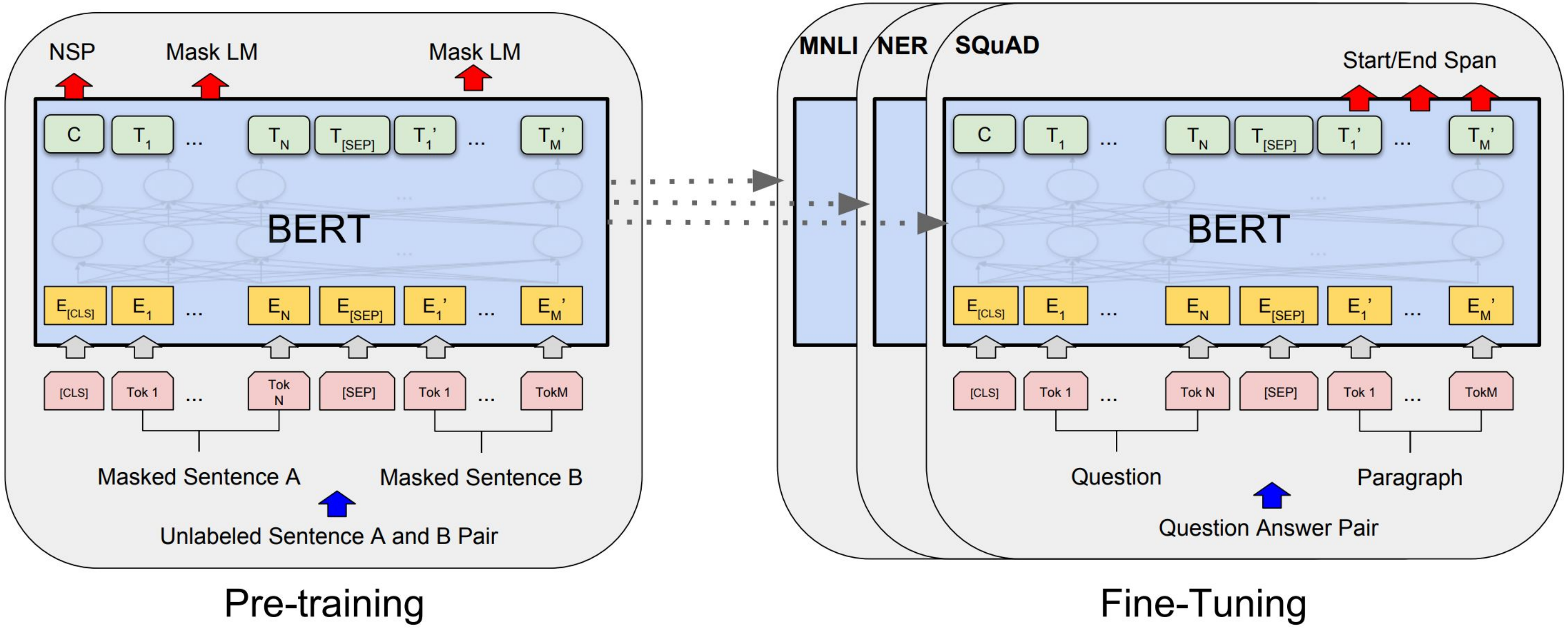
X_0_0	X_0_1	X_0_2	X_0_3
X_1_0	X_1_1	X_1_2	X_1_3

× W

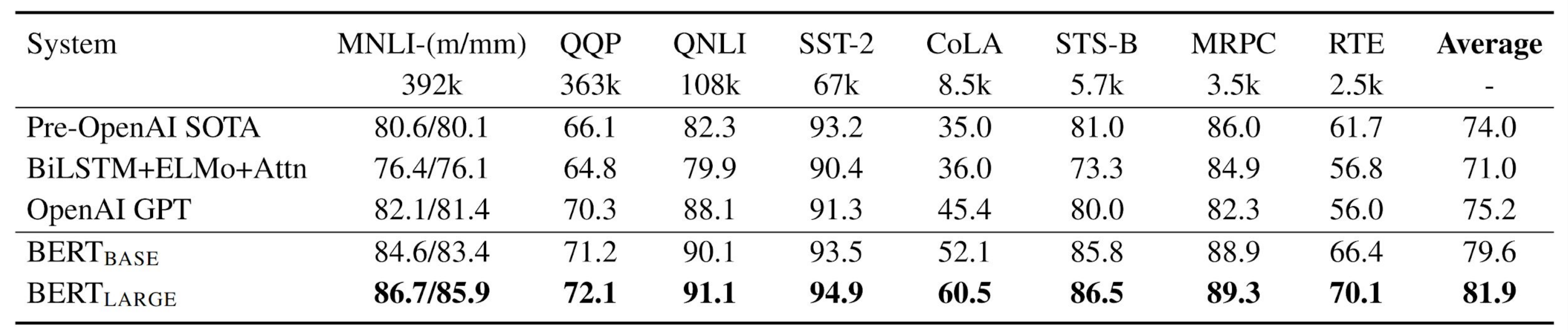
Model Details

- Data: Wikipedia (2.5B words) + BookCorpus (800M words)
- Batch Size: 131,072 words (1024 sequences * 128 length or 256 sequences * 512 length)
- Training Time: 1M steps (~40 epochs)
- 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



GLUE Results



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

MultiNLI

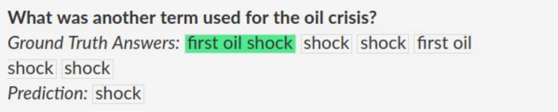
Premise: Hills and mountains are especially sanctified in Jainism.
Hypothesis: Jainism hates nature.
Label: Contradiction

CoLa

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

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

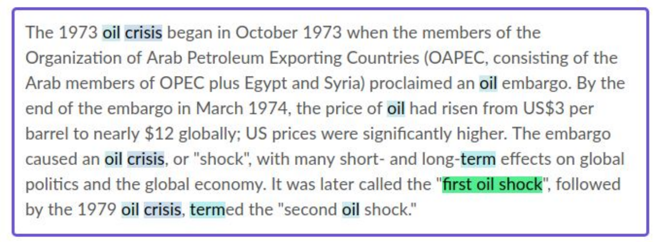
SQuAD 1.1



What was another term used for the oil crisis?

Ground Truth Answers: first oil shock | shock | shock | first oil shock | shock

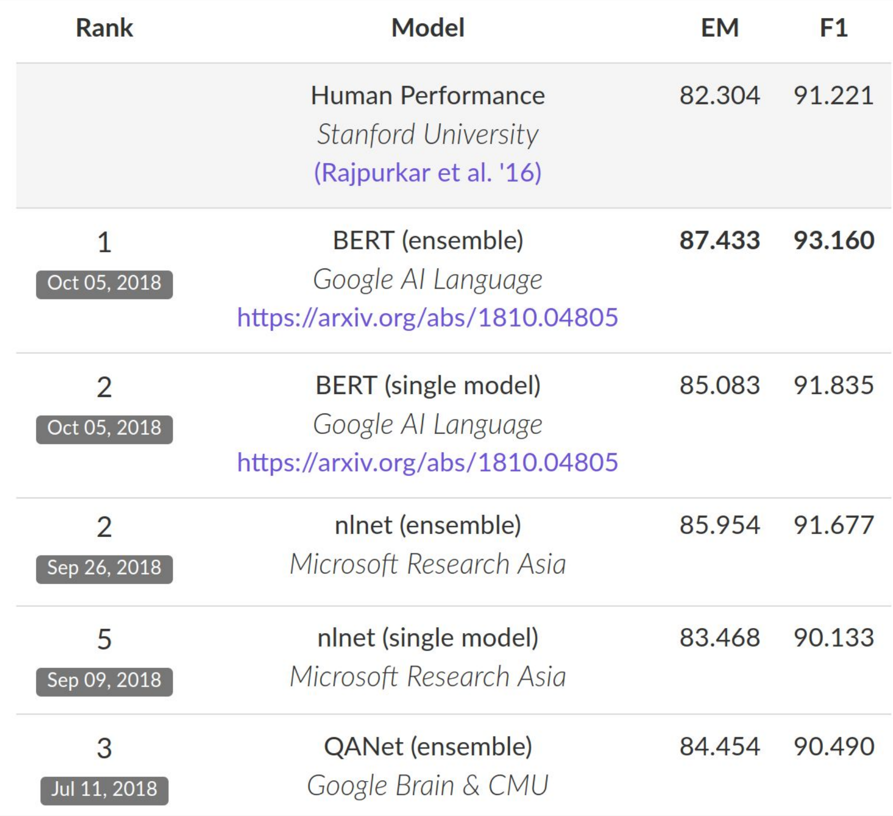
Prediction: shock



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."

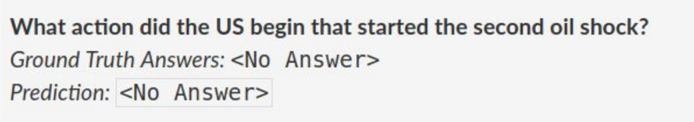
- Only new parameters: Start vector and end vector.
- Softr itions.

$$P_i = \frac{e^{S \cdot T_i}}{\sum_j e^{S \cdot T_j}}$$



Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar et al. '16)	82.304	91.221
1 Oct 05, 2018	BERT (ensemble) *Google AI Language* https://arxiv.org/abs/1810.04805	**87.433**	**93.160**
2 Oct 05, 2018	BERT (single model) *Google AI Language* https://arxiv.org/abs/1810.04805	85.083	91.835
2 Sep 26, 2018	nlnet (ensemble) *Microsoft Research Asia*	85.954	91.677
5 Sep 09, 2018	nlnet (single model) *Microsoft Research Asia*	83.468	90.133
3 Jul 11, 2018	QANet (ensemble) *Google Brain & CMU*	84.454	90.490

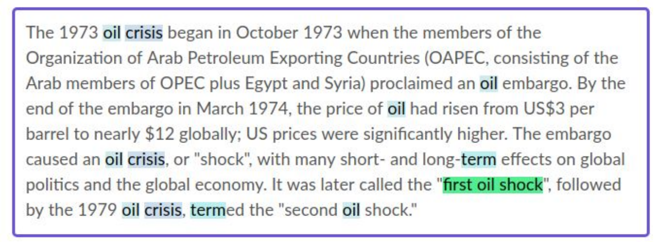
SQuAD 2.0



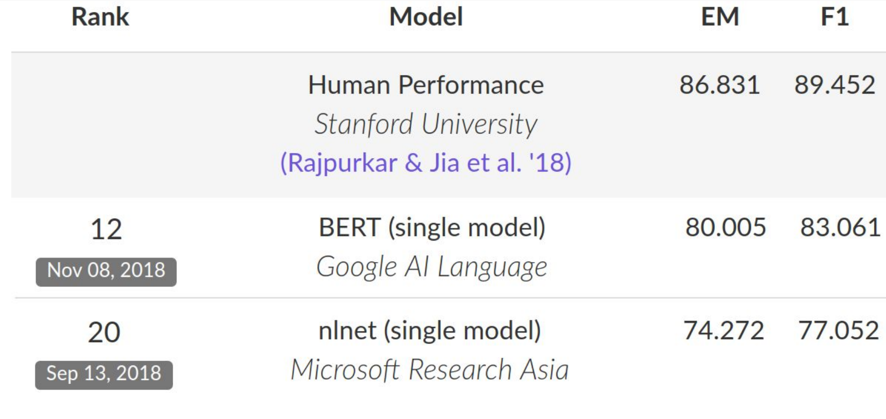
What action did the US begin that started the second oil shock?

Ground Truth Answers: `<No Answer>`

Prediction: `<No Answer>`



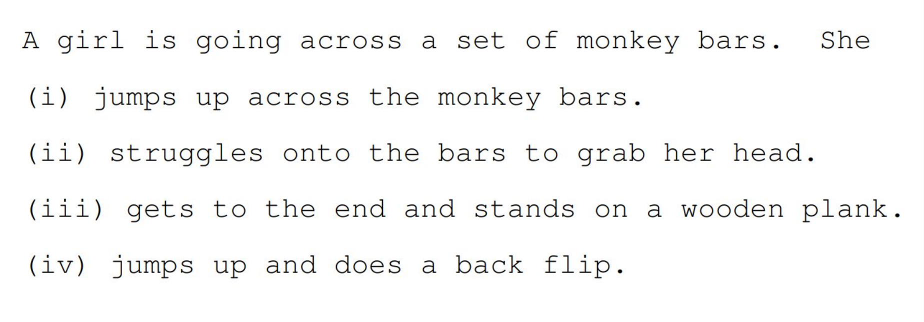
The 1973 oil crisis began in October 1973 when the members of the Organization of Arab Petroleum Exporting Countries (OAPEC, consisting of the Arab members of OPEC plus Egypt and Syria) proclaimed an oil embargo. By the end of the embargo in March 1974, the price of oil had risen from US$3 per barrel 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.

SWAG



A girl is going across a set of monkey bars. She

(i) jumps up across the monkey bars.

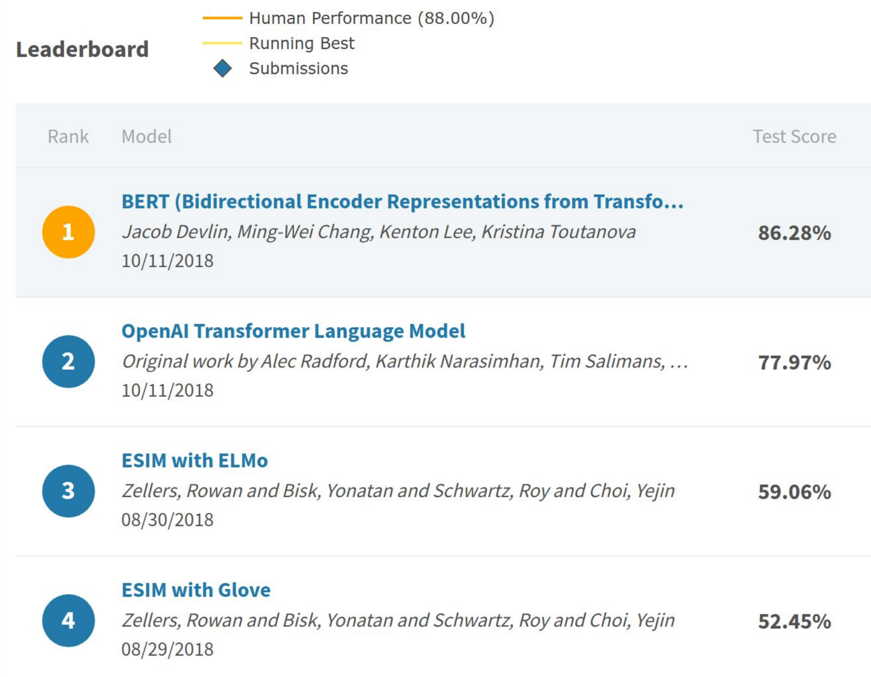
(ii) struggles onto the bars to grab her head.

(iii) gets to the end and stands on a wooden plank.

(iv) jumps up and does a back flip.

- Run each Premise + Ending through BERT.
- Produce logit for each pair on token 0 (`[CLS]`)

$$P_i = \frac{e^{V \cdot C_i}}{\sum_{j=1}^{4} e^{V \cdot C_j}}$$

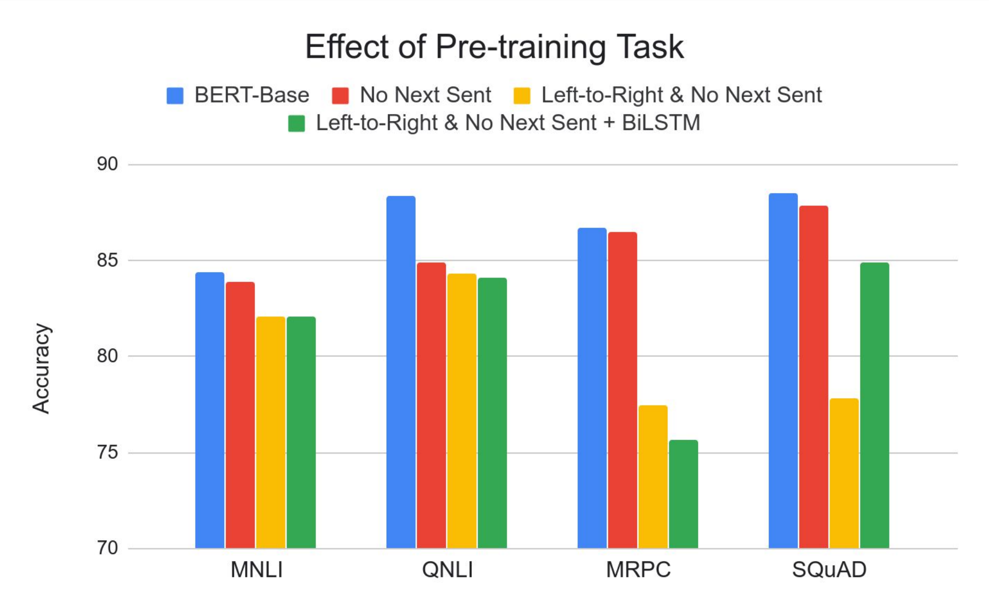


Leaderboard

— Human Performance (88.00%)
— Running Best
◆ Submissions

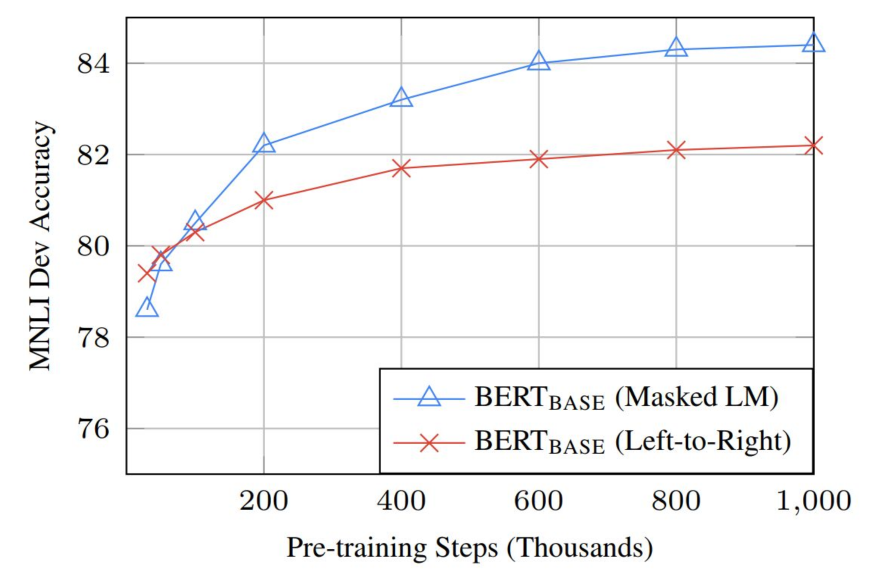
Rank	Model	Test Score
1	**BERT (Bidirectional Encoder Representations from Transfo...** *Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova* 10/11/2018	**86.28%**
2	**OpenAI Transformer Language Model** *Original work by Alec Radford, Karthik Narasimhan, Tim Salimans, ...* 10/11/2018	**77.97%**
3	**ESIM with ELMo** *Zellers, Rowan and Bisk, Yonatan and Schwartz, Roy and Choi, Yejin* 08/30/2018	**59.06%**
4	**ESIM with Glove** *Zellers, Rowan and Bisk, Yonatan and Schwartz, Roy and Choi, Yejin* 08/29/2018	**52.45%**

Effect of Pre-training Task



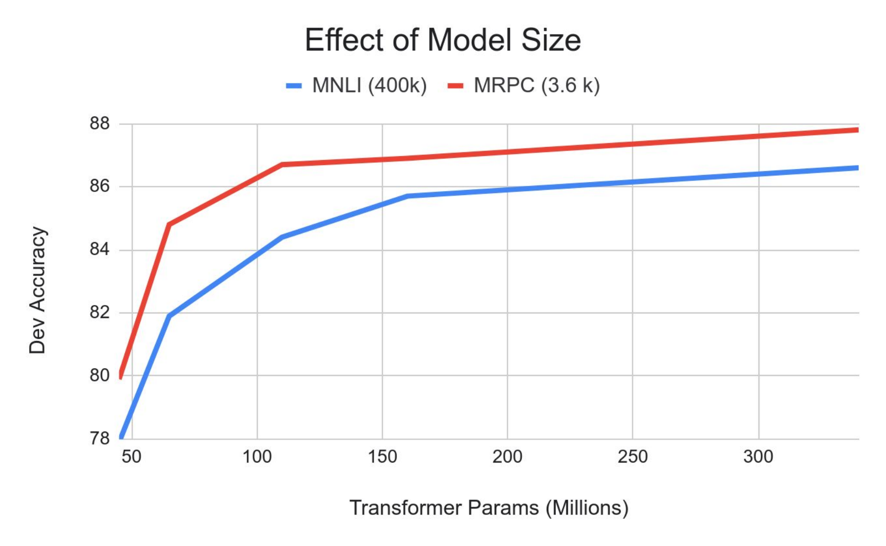
- Masked LM (compared to left-to-right LM) is very important on some tasks, Next Sentence Prediction is important on other tasks.
- Left-to-right model does very poorly on word-level task (SQuAD), although this is mitigated by BiLSTM

Effect of Directionality and Training Time



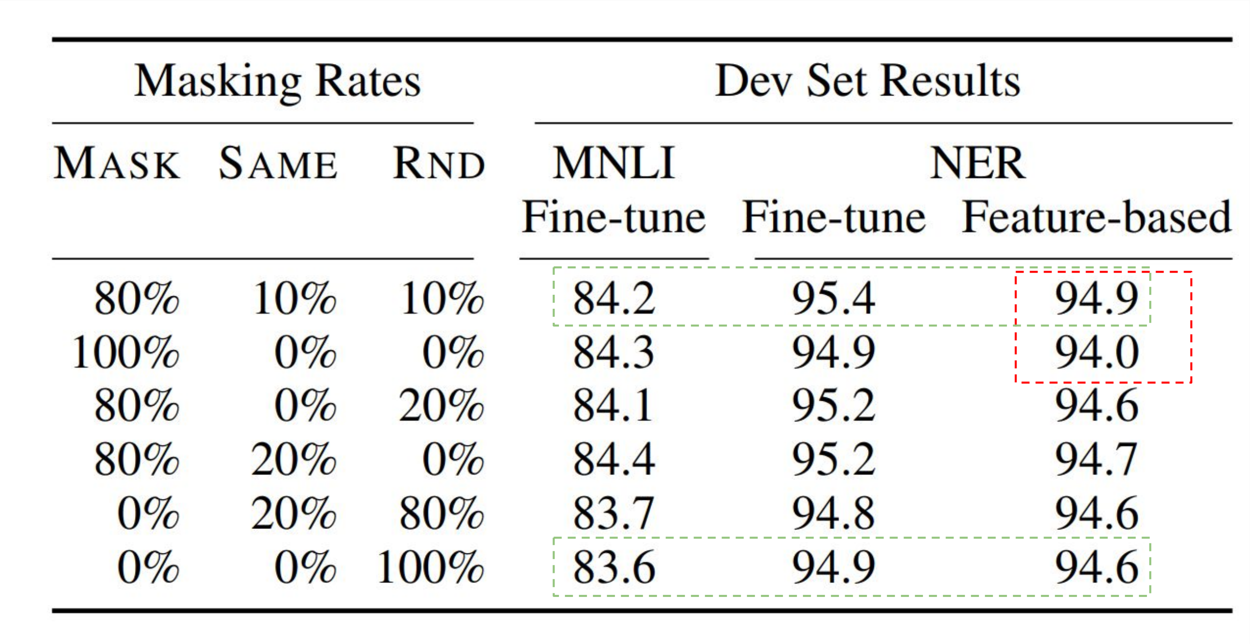
- Masked LM takes slightly longer to converge because we only predict 15% instead of 100%
- But absolute results are much better almost immediately

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

Effect of Masking Strategy



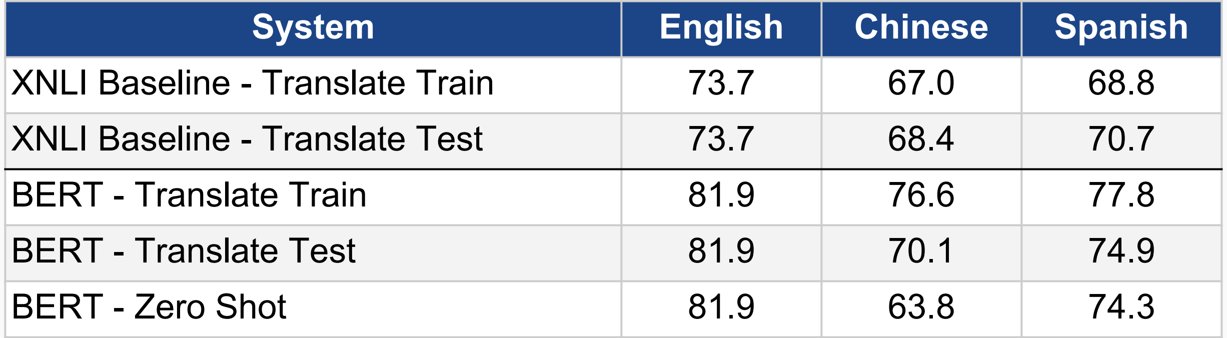
Masking Rates			Dev Set Results		
Mask	Same	Rnd	MNLI	NER	
			Fine-tune	Fine-tune	Feature-based
80%	10%	10%	84.2	95.4	94.9
100%	0%	0%	84.3	94.9	94.0
80%	0%	20%	84.1	95.2	94.6
80%	20%	0%	84.4	95.2	94.7
0%	20%	80%	83.7	94.8	94.6
0%	0%	100%	83.6	94.9	94.6

- Masking 100% of the time hurts on feature-based approach

- Using random word 100% of time hurts slightly

Multilingual BERT

- Trained single model on 104 languages from Wikipedia. Shared 110k WordPiece vocabulary.



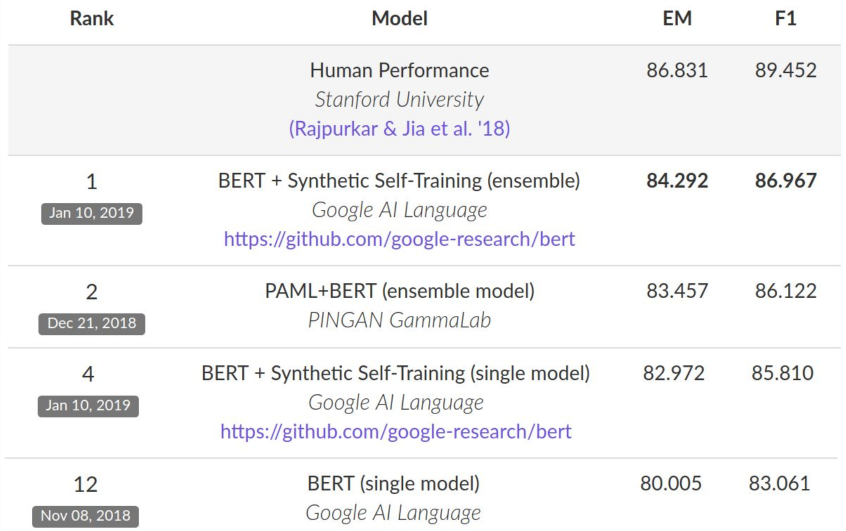
System	English	Chinese	Spanish
XNLI Baseline - Translate Train	73.7	67.0	68.8
XNLI Baseline - Translate Test	73.7	68.4	70.7
BERT - Translate Train	81.9	76.6	77.8
BERT - Translate Test	81.9	70.1	74.9
BERT - Zero Shot	81.9	63.8	74.3

- XNLI is MultiNLI translated into multiple languages.
- Always evaluate on human-translated Test.
- Translate Train: MT English Train into Foreign, then fine-tune.
- Translate Test: MT Foreign Test into English, use English model.
- Zero Shot: Use Foreign test on English model.

Synthetic Training Data

1. Use seq2seq model to generate positive questions from context+answer.
2. Heuristically transform positive questions into negatives (i.e., “no answer”/impossible).

- Result: +3.0 F1/EM score, new state-of-the-art.



Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar & Jia et al. '18)	86.831	89.452
1 Jan 10, 2019	BERT + Synthetic Self-Training (ensemble) *Google AI Language* https://github.com/google-research/bert	**84.292**	**86.967**
2 Dec 21, 2018	PAML+BERT (ensemble model) *PINGAN GammaLab*	83.457	86.122
4 Jan 10, 2019	BERT + Synthetic Self-Training (single model) *Google AI Language* https://github.com/google-research/bert	82.972	85.810
12 Nov 08, 2018	BERT (single model) *Google AI Language*	80.005	83.061

Synthetic Training Data

1. Pre-train seq2seq model on Wikipedia.
    - Encoder trained with BERT, Decoder trained to decode next sentence.
2. Fine-tune model on SQuAD Context+Answer ⟶ Question
    - `Ceratosaurus was a theropod dinosaur in the Late Jurassic, around 150 million years ago. -> When did the Ceratosaurus live ?`
3. Train model to predict answer spans without questions.
    - `Ceratosaurus was a theropod dinosaur in the Late Jurassic, around 150 million years ago.  -> {150 million years ago, 150 million, theropod dinsoaur, Late Jurassic, in the Late Jurassic}`

Synthetic Training Data

4. Generate answer spans from a lot of Wikipedia paragraphs using model from (3)
5. Use output of (4) as input to seq2seq model from (2) to generate synthetic questions:
    - Roxy Ann Peak is a 3,576-foot-tall mountain in the Western Cascade Range in the U.S. state of Oregon. → What state is Roxy Ann Peak in?
6. Filter with baseline SQuAD 2.0 system to throw out bad questions.
    - Roxy Ann Peak is a 3,576-foot-tall mountain in the Western Cascade Range in the U.S. state of Oregon. → What state is Roxy Ann Peak in? (**Good**)
    - Roxy Ann Peak is a 3,576-foot-tall mountain in the Western Cascade Range in the U.S. state of Oregon. → Where is Oregon? (**Bad**)

7. Heuristically generate “strong negatives”:

   a. Positive questions from other paragraphs of same document.

      ```
      What state is Roxy Ann Peak in? → When was Roxy Ann Peak first summited?
      ```

   b. Replace span of text with other span of same type (based on POS tags). Replacement is usually from paragraph.

      ```
      What state is Roxy Ann Peak in? → What state is Oregon in?
      What state is Roxy Ann Peak in? → What mountain is Roxy Ann Peak in?
      ```

8. Optionally: Two-pass training, where no-answer is modeled as regression second pass (~+0.5 F1)

Common Questions

- Is *deep* bidirectionality really necessary? What about ELMo-style shallow bidirectionality on bigger model?
- Advantage: Slightly faster training time
- Disadvantages:
  - Will need to add non-pre-trained bidirectional model on top
  - Right-to-left SQuAD model doesn’t see question
  - Need to train two models
  - Off-by-one: LTR predicts next word, RTL predicts previous word
  - Not trivial to add arbitrary pre-training tasks.

Common Questions

- Why did no one think of this before?
- Better question: Why wasn’t contextual pre-training popular before 2018 with ELMo?
- Good results on pre-training is >1,000x to 100,000 more expensive than supervised training.
    - E.g., 10x-100x bigger model trained for 100x-1,000x as many steps.
    - Imagine it’s 2013: Well-tuned 2-layer, 512-dim LSTM sentiment analysis gets 80% accuracy, training for 8 hours.
    - Pre-train LM on same architecture for a week, get 80.5%.
    - Conference reviewers: “Who would do something so expensive for such a small gain?”

Common Questions

- The model must be learning more than “contextual embeddings”
- Alternate interpretation: Predicting missing words (or next words) requires learning many types of language understanding features.
    - syntax, semantics, pragmatics, coreference, etc.
- Implication: Pre-trained model is much bigger than it needs to be to solve specific task
- Task-specific model distillation words very well

Common Questions

- Is modeling “solved” in NLP? I.e., is there a reason to come up with novel model architectures?
    - But that’s the most fun part of NLP research :(
- Maybe yes, for now, on some tasks, like SQuAD-style QA.
    - At least using the same deep learning “lego blocks”
- Examples of NLP models that are not “solved”:
    - Models that minimize total training cost vs. accuracy on modern hardware
    - Models that are very parameter efficient (e.g., for mobile deployment)
    - Models that represent knowledge/context in latent space
    - Models that represent structured data (e.g., knowledge graph)
    - Models that jointly represent vision and language

Common Questions

- Personal belief: Near-term improvements in NLP will be mostly about making clever use of “free” data.
    - Unsupervised vs. semi-supervised vs. synthetic supervised is somewhat arbitrary.
    - “Data I can get a lot of without paying anyone” vs. “Data I have to pay people to create” is more pragmatic distinction.
- No less “prestigious” than modeling papers:
    - *Phrase-Based & Neural Unsupervised Machine Translation*, Facebook AI Research, EMNLP 2018 Best Paper

Conclusions

- Empirical results from BERT are great, but biggest impact on the field is:
- With pre-training, bigger == better, without clear limits (so far).
- Unclear if adding things on top of BERT really helps by very much.
    - Good for people and companies building NLP systems.
    - Not necessary a “good thing” for researchers, but important.