BERT

- AI2 made ELMo in spring 2018, GPT (transformer-based ELMo) was released in summer 2018, BERT came out October 2018
- Four major changes compared to ELMo:
  - Transformers instead of LSTMs
  - Bidirectional model with “Masked LM” objective instead of standard LM
  - Fine-tune instead of freeze at test time
  - Operates over word pieces (byte pair encoding)

ELMo is a unidirectional model (as is GPT): we can concatenate two unidirectional models, but is this the right thing to do?

ELMo reprs look at each direction in isolation; BERT looks at them jointly

A stunning ballet dancer, Copeland is one of the best performers to see live.

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

You could do this with a “one-sided” transformer, but this “two-sided” model can cheat

Devlin et al. (2019)
**Masked Language Modeling**

- How to prevent cheating? Next word prediction fundamentally doesn’t work for bidirectional models, instead do masked language modeling.

- BERT formula: take a chunk of text, mask out 15% of the tokens, and try to predict them.

  ![Masked Language Modeling Diagram]

  John visited [MASK] yesterday

  Madagascar

  Devlin et al. (2019)

**Next “Sentence” Prediction**

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

  ![Next “Sentence” Prediction Diagram]


  Devlin et al. (2019)

**BERT Architecture**

- BERT Base: 12 layers, 768-dim per wordpiece token, 12 heads. Total params = 110M
- BERT Large: 24 layers, 1024-dim per wordpiece token, 16 heads. Total params = 340M
- Positional embeddings and segment embeddings, 30k word pieces
- This is the model that gets pre-trained on a large corpus.

  ![BERT Architecture Diagram]

  Devlin et al. (2019)

**What can BERT do?**

- Artificial [CLS] token is used as the vector to do classification from.
- Sentence pair tasks (entailment): feed both sentences into BERT.
- BERT can also do tagging by predicting tags at each word piece.

  ![What can BERT do? Diagram]

  Devlin et al. (2019)
What can BERT do?

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

Entails (first sentence implies second is true)

¬

Transformer ...

Transformer

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

What can BERT NOT do?

‣ BERT cannot generate text (at least not in an obvious way)
‣ Can fill in MASK tokens, but can’t generate left-to-right (well, you could put MASK at the end repeatedly, but this is slow)
‣ Masked language models are intended to be used primarily for “analysis” tasks

Fine-tuning BERT

‣ Fine-tune for 1-3 epochs, batch size 2-32, learning rate 2e-5 - 5e-5
‣ Large changes to weights up here (particularly in last layer to route the right information to [CLS])
‣ Smaller changes to weights lower down in the transformer
‣ Small LR and short fine-tuning schedule mean weights don’t change much
‣ More complex “triangular learning rate” schemes exist

Fine-tuning BERT (Peters, Ruder, Smith (2019))

<table>
<thead>
<tr>
<th>Pretraining</th>
<th>Adaptation</th>
<th>NER CoNLL 2003</th>
<th>SA SST-2</th>
<th>Nat. lang. inference MNLI</th>
<th>SICK-E</th>
<th>Semantic textual similarity</th>
<th>MRPC</th>
<th>STS-B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skip-thoughts</td>
<td>🔥</td>
<td>-</td>
<td>81.8</td>
<td>62.9</td>
<td>-</td>
<td>86.6</td>
<td>75.8</td>
<td>71.8</td>
</tr>
<tr>
<td>ELMo</td>
<td>🔥</td>
<td>91.7</td>
<td>91.8</td>
<td>79.6</td>
<td>86.3</td>
<td>86.1</td>
<td>76.0</td>
<td>75.9</td>
</tr>
<tr>
<td>BERT-base</td>
<td>🔥</td>
<td>92.2</td>
<td>93.0</td>
<td>84.6</td>
<td>84.8</td>
<td>86.4</td>
<td>78.1</td>
<td>82.9</td>
</tr>
</tbody>
</table>

Delta = \( \Delta = \frac{\text{ELMo} - \text{BERT-base}}{\text{BERT-base}} \)
## Evaluation: GLUE

<table>
<thead>
<tr>
<th>Corpus</th>
<th>[Train]</th>
<th>[Test]</th>
<th>Task</th>
<th>Metrics</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>CoLA</td>
<td>8.5k</td>
<td>1k</td>
<td>acceptability</td>
<td>Matthews corr.</td>
<td>misc.</td>
</tr>
<tr>
<td>SST-2</td>
<td>67k</td>
<td>1.8k</td>
<td>sentiment</td>
<td>acc.</td>
<td>movie reviews</td>
</tr>
</tbody>
</table>

### Single-Sentence Tasks

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

---

## Results

### System | MNLI-(m/mm) | QQP | QNLI | SST-2 | CoLA | STS-B | MRPC | RTE | Average |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-OpenAI SOTA</td>
<td>80.6/80.1</td>
<td>66.1</td>
<td>82.3</td>
<td>93.2</td>
<td>35.0</td>
<td>81.0</td>
<td>86.0</td>
<td>61.7</td>
<td>74.0</td>
</tr>
<tr>
<td>BiLSTM+ELMo+Attn</td>
<td>76.4/76.1</td>
<td>64.8</td>
<td>79.9</td>
<td>90.4</td>
<td>36.0</td>
<td>73.3</td>
<td>84.9</td>
<td>56.8</td>
<td>71.0</td>
</tr>
<tr>
<td>OpenAI GPT</td>
<td>82.1/81.4</td>
<td>70.3</td>
<td>88.1</td>
<td>91.3</td>
<td>45.4</td>
<td>80.0</td>
<td>82.3</td>
<td>56.0</td>
<td>75.2</td>
</tr>
</tbody>
</table>

### Devlin et al. (2018)

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

---

## RoBERTa

- “Robustly optimized BERT”
- 160GB of data instead of 16 GB
- Dynamic masking: standard BERT uses the same MASK scheme for every epoch, RoBERTa recomputes them
- New training + more data = better performance

---

## Using BERT

- Huggingface Transformers: big open-source library with most pre-trained architectures implemented, weights available
- Lots of standard models... and “community models”

---

### Model architectures

- Transformers currently provides the following NQGLUE architectures:
  1. BERT, from Google, released with the paper BERT: Pre-training of Deep Bi-Directional Transformers by Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova
  2. GPT (from OpenAI), released with the paper Improving Language Understanding by Generative Pre-training by Ilja Ruder
  3. GPT-2 (from OpenAI), released with the paper Language Models are Unicorns by Jeffrey Wu, Ronan Collobert, Yin-Feng Yang, Yiming Yang, Yiming Yang, and Yann Dauphin
  4. BART (from Facebook), released with the paper Compressing Transformers with Policy Gradient by Zhong Guo, Zhifeng Yang, Yiming Yang, and Yiming Yang
  5. Roberta (from IBM), released with the paper Pre-training of Deep Bidirectional Transformers for Language Understanding by Zhifeng Yang, Zhifeng Yang, Yiming Yang, and Yiming Yang
  6. RoBERTa-Large (from Facebook), released together with the paper On Robustness of Large Language Models by Yiming Yang, and Alec Corr
c
  7. RoBERTa-Large (from Facebook), released together with the paper On Robustness of Large Language Models by Yiming Yang, and Alec Corr

---
What does BERT learn?

Heads on transformers learn interesting and diverse things: content heads (attend based on content), positional heads (based on position), etc. Clark et al. (2019)

Still way worse than what supervised systems can do, but interesting that this is learned organically. Clark et al. (2019)