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.

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.
- BERT objective: masked LM + next sentence prediction.

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.

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?

‣ 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

<table>
<thead>
<tr>
<th>Pretraining</th>
<th>Adaptation</th>
<th>NER</th>
<th>SA</th>
<th>Nat. lang. inference</th>
<th>Semantic textual similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skip-thoughts</td>
<td></td>
<td>81.8</td>
<td>62.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ELMo</td>
<td></td>
<td>91.9</td>
<td>79.6</td>
<td>86.3</td>
<td>66.1</td>
</tr>
<tr>
<td>BERT-base</td>
<td></td>
<td>92.2</td>
<td>93.0</td>
<td>84.6</td>
<td>86.4</td>
</tr>
</tbody>
</table>

Peters, Ruder, Smith (2019)
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>

Results

<table>
<thead>
<tr>
<th>System</th>
<th>MNLi-(m/mm)</th>
<th>QQP</th>
<th>QNLI</th>
<th>SST-2</th>
<th>CoLA</th>
<th>STS-B</th>
<th>MRPC</th>
<th>RTE</th>
<th>Average</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+Att</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>
<tr>
<td>BERT BASE</td>
<td>84.6/83.4</td>
<td>71.2</td>
<td>90.1</td>
<td>93.5</td>
<td>52.1</td>
<td>85.8</td>
<td>88.9</td>
<td>66.4</td>
<td>79.6</td>
</tr>
<tr>
<td>BERT LARGE</td>
<td>86.7/85.9</td>
<td>72.1</td>
<td>91.1</td>
<td>94.9</td>
<td>60.5</td>
<td>86.5</td>
<td>89.3</td>
<td>70.1</td>
<td>81.9</td>
</tr>
</tbody>
</table>

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

Wang et al. (2019)

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

Liu et al. (2019)

Using BERT

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

...
**OpenAI GPT/GPT2**

- “ELMo with transformers” (works better than ELMo)
- Train a single unidirectional transformer LM on long contexts
  - GPT2: trained on 40GB of text collected from upvoted links from reddit
  - 1.5B parameters — by far the largest of these models trained when it came out in March 2019
    - Parameters | Layers | $d_{model}$
    - 117M | 12 | 768
    - 345M | 24 | 1024
    - 762M | 36 | 1280
    - 1542M | 48 | 1600
- Because it's a language model, we can generate from it

Radford et al. (2019)

---

**Open Questions**

1) How novel is the stuff being generated? (Is it just doing nearest neighbors on a large corpus?)
2) How do we understand and distill what is learned in this model?
3) How do we harness these priors for conditional generation tasks (summarization, generate a report of a basketball game, etc.)
4) Is this technology dangerous? (OpenAI pursued a “staged release” strategy and didn’t release biggest model)

---

**Slide credit:** OpenAI
Pre-Training Cost (with Google/AWS)

- BERT: Base $500, Large $7000
- Grover-MEGA (GPT-2 variant): $25,000
- XLNet (BERT variant): $30,000 → $60,000 (unclear)
- This is for a single pre-training run...developing new pre-training techniques may require many runs
- Fine-tuning these models can typically be done with a single GPU (but may take 1-3 days for medium-sized datasets)


Pushing the Limits

- NVIDIA: trained 8.3B parameter GPT model (5.6x the size of GPT-2)
- Arguable these models are still underfit: larger models still get better held-out perplexities

NVIDIA blog (Narasimhan, August 2019)