CS388: Natural Language Processing

Lecture 19: Pretrained Transformers

Greg Durrett

Credit: ???
Administrivia

- Project 2 due Tuesday
- Presentation day announcements next week
Recall: Self-Attention

- Each word forms a “query” which then computes attention over each word
  \[ \alpha_{i,j} = \text{softmax}(x_i^T x_j) \quad \text{scalar} \]
  \[ x'_i = \sum_{j=1}^{n} \alpha_{i,j} x_j \quad \text{vector} = \text{sum of scalar} \times \text{vector} \]

- Multiple “heads” analogous to different convolutional filters. Use parameters \( W_k \) and \( V_k \) to get different attention values + transform vectors
  \[ \alpha_{k,i,j} = \text{softmax}(x_i^T W_k x_j) \quad x'_{k,i} = \sum_{j=1}^{n} \alpha_{k,i,j} V_k x_j \]

Vaswani et al. (2017)
Recall: Transformers

- Augment word embedding with position embeddings, each dim is a sine/cosine wave of a different frequency. Closer points = higher dot products.
- Works essentially as well as just encoding position as a one-hot vector.

Vaswani et al. (2017)
This Lecture

- BERT
- GPT/GPT2
- Analysis/Visualization
BERT
Three major changes compared to ELMo:

- Transformers instead of LSTMs (transformers in GPT as well)
- Bidirectional <=> Masked LM objective instead of standard LM
- Fine-tune instead of freeze at test time

AI2 made ELMo in spring 2018, GPT was released in summer 2018, BERT came out October 2018
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.

Devlin et al. (2019)
How to learn a “deeply bidirectional” model? What happens if we just replace an LSTM with a transformer?

Transformer LMs have to be “one-sided” (only attend to previous tokens), not what we want.
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, predict 15% of the tokens
  - For 80% (of the 15%), replace the input token with [MASK]
  - For 10%, replace w/random
  - For 10%, keep same

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
- BERT objective: masked LM + next sentence prediction

NotNext

Madagascar
enjoyed
like

Transformer

...
- 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

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

- CLS token is used to provide classification decisions
- Sentence pair tasks (entailment): feed both sentences into BERT
- BERT can also do tagging by predicting tags at each word piece

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

Entails

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

- 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
What can BERT NOT do?

- BERT **cannot** generate text (at least not in an obvious way)
- Not an autoregressive model, can do weird things like stick a [MASK] at the end of a string, fill in the mask, and repeat
- 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

(b) Single Sentence Classification Tasks: SST-2, CoLA
<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>Semantic textual similarity SICK-R</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>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>
</tr>
<tr>
<td></td>
<td>🌬🔥</td>
<td>91.9</td>
<td>91.2</td>
<td>76.4</td>
<td>83.3</td>
<td>83.3</td>
<td>74.7</td>
</tr>
<tr>
<td></td>
<td>Δ=🔥-❄️</td>
<td>0.2</td>
<td>-0.6</td>
<td>-3.2</td>
<td>-3.3</td>
<td>-2.8</td>
<td>-1.3</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>
</tr>
<tr>
<td></td>
<td>🌬🔥</td>
<td>92.4</td>
<td>93.5</td>
<td>84.6</td>
<td>85.8</td>
<td>88.7</td>
<td>84.8</td>
</tr>
<tr>
<td></td>
<td>Δ=🔥-❄️</td>
<td>0.2</td>
<td>0.5</td>
<td>0.0</td>
<td>1.0</td>
<td>2.3</td>
<td>6.7</td>
</tr>
</tbody>
</table>

- BERT is typically better if the whole network is fine-tuned, unlike ELMo

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><strong>Single-Sentence Tasks</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<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>
<tr>
<td><strong>Similarity and Paraphrase Tasks</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MRPC</td>
<td>3.7k</td>
<td>1.7k</td>
<td>paraphrase</td>
<td>acc./F1</td>
<td>news</td>
</tr>
<tr>
<td>STS-B</td>
<td>7k</td>
<td>1.4k</td>
<td>sentence similarity</td>
<td>Pearson/Spearman corr.</td>
<td>misc.</td>
</tr>
<tr>
<td>QQP</td>
<td>364k</td>
<td>391k</td>
<td>paraphrase</td>
<td>acc./F1</td>
<td>social QA questions</td>
</tr>
<tr>
<td><strong>Inference Tasks</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MNLI</td>
<td>393k</td>
<td>20k</td>
<td>NLI</td>
<td>matched acc./mismatched acc.</td>
<td>misc.</td>
</tr>
<tr>
<td>QNLI</td>
<td>105k</td>
<td>5.4k</td>
<td>QA/NLI</td>
<td>acc.</td>
<td>Wikipedia</td>
</tr>
<tr>
<td>RTE</td>
<td>2.5k</td>
<td>3k</td>
<td>NLI</td>
<td>acc.</td>
<td>news, Wikipedia</td>
</tr>
<tr>
<td>WNLi</td>
<td>634</td>
<td>146</td>
<td>coreference/NLI</td>
<td>acc.</td>
<td>fiction books</td>
</tr>
</tbody>
</table>

Wang et al. (2019)
Huge improvements over prior work (even compared to ELMo)

Effective at “sentence pair” tasks: textual entailment (does sentence A imply sentence B), paraphrase detection

Devlin et al. (2018)
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)
GPT/GPT2
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 as of March 2019
- Because it's a language model, we can generate from it

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Layers</th>
<th>$d_{model}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>117M</td>
<td>12</td>
<td>768</td>
</tr>
<tr>
<td>345M</td>
<td>24</td>
<td>1024</td>
</tr>
<tr>
<td>762M</td>
<td>36</td>
<td>1280</td>
</tr>
<tr>
<td>1542M</td>
<td>48</td>
<td>1600</td>
</tr>
</tbody>
</table>

Radford et al. (2019)
Miley Cyrus was caught shoplifting from Abercrombie and Fitch on Hollywood Boulevard today.

The 19-year-old singer was caught on camera being escorted out of the store by security guards.

The singer was wearing a black hoodie with the label ‘Blurred Lines’ on the front and ‘Fashion Police’ on the back.

Scroll down for video

Shoplifting: Miley Cyrus was caught shoplifting from Abercrombie and Fitch on Hollywood Boulevard today (pictured)

The singer was wearing a black hoodie with the label ‘Blurred Lines’ on the front and ‘Fashion Police’ on the back.

The singer was also wearing a pair of black-rimmed glasses, a black jacket, black jeans and black sandals.

She was carrying a pair of black and white striped gloves and a small black bag.
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 has only released 774M param model, not 1.5B yet)
Grover

- Sample from a large language model conditioned on a domain, date, authors, and headline

- Humans rank Grover-generated propaganda as more realistic than real “fake news”

- Fine-tuned Grover can detect Grover propaganda easily — authors argue for releasing it for this reason

- NOTE: Not a GAN, discriminator trained separately from the generator

---

**Unpaired Accuracy**

<table>
<thead>
<tr>
<th>Generator size</th>
<th>1.5B</th>
<th>355M</th>
<th>124M</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chance</td>
<td>50.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Grover-Mega</td>
<td>92.0</td>
<td>98.5</td>
<td>99.8</td>
</tr>
<tr>
<td>Grover-Large</td>
<td>80.8</td>
<td>91.2</td>
<td>98.4</td>
</tr>
<tr>
<td>BERT-Large</td>
<td>73.1</td>
<td>75.9</td>
<td>97.5</td>
</tr>
<tr>
<td>GPT2</td>
<td>70.1</td>
<td>78.0</td>
<td>90.3</td>
</tr>
</tbody>
</table>

**Paired Accuracy**

<table>
<thead>
<tr>
<th>Generator size</th>
<th>1.5B</th>
<th>355M</th>
<th>124M</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chance</td>
<td>50.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Grover-Mega</td>
<td>97.4</td>
<td>100.0</td>
<td>100.0</td>
</tr>
<tr>
<td>Grover-Large</td>
<td>89.0</td>
<td>96.9</td>
<td>100.0</td>
</tr>
<tr>
<td>BERT-Large</td>
<td>84.1</td>
<td>91.5</td>
<td>99.9</td>
</tr>
<tr>
<td>GPT2</td>
<td>78.8</td>
<td>87.0</td>
<td>96.8</td>
</tr>
<tr>
<td>BERT-Base</td>
<td>77.5</td>
<td>88.2</td>
<td>95.7</td>
</tr>
<tr>
<td>GPT2</td>
<td>72.5</td>
<td>79.6</td>
<td>89.6</td>
</tr>
</tbody>
</table>

---

Zellers et al. (2019)
Pre-Training Cost (with Google/AWS)

- BERT: Base $500, Large $7000
- Grover-MEGA: $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)
Google T5

<table>
<thead>
<tr>
<th>Number of tokens</th>
<th>Repeats</th>
<th>GLUE</th>
<th>CNNDM</th>
<th>SQuAD</th>
<th>SGLUE</th>
<th>EnDe</th>
<th>EnFr</th>
<th>EnRo</th>
</tr>
</thead>
<tbody>
<tr>
<td>★ Full dataset</td>
<td>0</td>
<td>83.28</td>
<td>19.24</td>
<td>80.88</td>
<td>71.36</td>
<td>26.98</td>
<td>39.82</td>
<td>27.65</td>
</tr>
<tr>
<td>$2^{29}$</td>
<td>64</td>
<td>82.87</td>
<td>19.19</td>
<td>80.97</td>
<td>72.03</td>
<td>26.83</td>
<td>39.74</td>
<td>27.63</td>
</tr>
<tr>
<td>$2^{27}$</td>
<td>256</td>
<td>82.62</td>
<td>19.20</td>
<td>79.78</td>
<td>69.97</td>
<td>27.02</td>
<td>39.71</td>
<td>27.33</td>
</tr>
<tr>
<td>$2^{25}$</td>
<td>1,024</td>
<td>79.55</td>
<td>18.57</td>
<td>76.27</td>
<td>64.76</td>
<td>26.38</td>
<td>39.56</td>
<td>26.80</td>
</tr>
<tr>
<td>$2^{23}$</td>
<td>4,096</td>
<td>76.34</td>
<td>18.33</td>
<td>70.92</td>
<td>59.29</td>
<td>26.37</td>
<td>38.84</td>
<td>25.81</td>
</tr>
</tbody>
</table>

- Colossal Cleaned Common Crawl: 750 GB of text
- We still haven't hit the limit of bigger data being useful
BART

- Sequence-to-sequence BERT variant: permute/make/delete tokens, then predict full sequence autoregressively
- For downstream tasks: feed document into both encoder + decoder, use decoder hidden state as output
- Good results on dialogue, summarization tasks

Lewis et al. (October 30, 2019)
Analysis
Heads on transformers learn interesting and diverse things: content heads (attend based on content), positional heads (based on position), etc.

Clark et al. (2019)
What does BERT learn?

- **Direct objects** attend to their verbs
  - 86.8% accuracy at the `dobj` relation

- **Noun modifiers** (e.g., determiners) attend to their noun
  - 94.3% accuracy at the `det` relation

- **Coreferent** mentions attend to their antecedents
  - 65.1% accuracy at linking the head of a coreferent mention to the head of an antecedent

Still way worse than what supervised systems can do, but interesting that this is learned organically

Clark et al. (2019)
Probing BERT

- Try to predict POS, etc. from each layer.
  Learn mixing weights
  
  $$h_{i,\tau} = \gamma_{\tau} \sum_{\ell=0}^{L} s_{\tau}^{(\ell)} h_{i}^{(\ell)}$$

  representation of wordpiece $i$ for task $\tau$

- Plot shows $s$ weights (blue) and performance deltas when an additional layer is incorporated (purple)

- BERT “rediscovering the classical NLP pipeline”: first syntactic tasks then semantic ones

Tenney et al. (2019)
Compressing BERT

- Remove 60+% of BERT’s heads with minimal drop in performance

- DistilBERT (Sanh et al., 2019): nearly as good with half the parameters of BERT (via knowledge distillation)

(b) Evolution of accuracy on the MultiNLI-matched validation set when heads are pruned from BERT according to $I_h$ (solid blue) and accuracy difference (dashed green).

Michel et al. (2019)
Open Questions

- BERT-based systems are state-of-the-art for nearly every major text analysis task
- These techniques are here to stay, unclear what form will win out
- Role of academia vs. industry: no major pretrained model has come purely from academia
- Cost/carbon footprint: a single model costs $10,000+ to train (though this cost should come down)