Contextualized Word Representations

Natural Language Processing

Based on slides from Stanford cs224n (Chris Manning and Abigail See) and Graham Neubig
Plan

- Bidirectional RNNs
- Multi-layer RNNs
- Contextualized word representations
Sentence classification

• We learned about word representations

• We learned about representations of words in context with RNNs

• Can we use these tools for general NLP tasks?
Sentence classification

• Classify sentences according to various traits

• Topic, sentiment etc.

I have this movie

I love this movie

very good

good

neutral

bad

very bad

very good

good

neutral

bad

very bad

• It would be good to have a sentence representation
Sentence classification

• Many tasks are variants of this:
  • Textual entailment
  • Textual similarity
  • ...
Word representations

- **Func**: sum, average, element-wise mean-pooling, element-wise max-pooling
- **Objective**: cross-entropy between true class and predicted distribution
- **Parameters**: word embeddings
Contextualized word rep.

- **Func**: sum, average, element-wise mean-pooling, element-wise max-pooling, last hidden state
- **Objective**: cross-entropy between true class and predicted distribution
- **Parameters**: word embeddings + RNN parameters

This contains information on left context only!
Bidirectional RNN

\[ \overrightarrow{h_t} = \text{RNN}_{fw}(\overrightarrow{h_{t-1}}, x_t) \]

\[ \overleftarrow{h_t} = \text{RNN}_{bw}(\overleftarrow{h_{t+1}}, x_t) \]

\[ h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}] \]
Bidirectional RNN

- RNN here can be LSTM/GRU/whatever
- Each hidden state is now contextualized with both left and right context
- Parameters for forward and backward RNNs are separate
- These are applicable for the sentence classification task but not for generative LMs where only the history is observed.
- In general, using both sides improves performance by a lot!
Multi-layer RNN

- We can make RNNs deeper by having the input of each token in one RNN be the input to a deeper RNN.

- These are called multi-layer RNN or stacked RNNs.
The movie was terribly exciting.

Hidden state/output state of layer $i$ is the input to layer $i+1$. 
Transformers

• There is now an alternative to RNNs called transformer

• It has no recurrence

• It uses many layers compared to RNN (20 vs. 2)

• It works quite well

• It is hard to optimize

• We might look at it in detail later

• But for now - think RNN
Thought

• Word representations are nice:
  • Can be trained on tons of data

• But:
  • They don’t take context into account

• Can we train representations for words in context on tons of data and use them like word embeddings?
  • YES!
ELMO

• Deep contextualized word representations (Peters et al., 2018) - most cited NLP paper published in 2018 with 386 citations

• Gist:
  
  • Train a multi-layer forward LM and a backward LM.
  
  • Represent each word with all of the representations for all of the layers
  
  • For any model that uses contextualized word representations, just initialize model with the trained weights and keep training on end task
ELMO

- Assume the task is to predict something for every word in the sentence (e.g., is it a noun or not):
## ELMO Results

<table>
<thead>
<tr>
<th>Task</th>
<th>Previous SOTA</th>
<th>Our baseline</th>
<th>ELMo + baseline</th>
<th>Increase (absolute/relative)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQuAD</td>
<td>Liu et al. (2017)</td>
<td>84.4</td>
<td>81.1</td>
<td>85.8</td>
</tr>
<tr>
<td>SNLI</td>
<td>Chen et al. (2017)</td>
<td>88.6</td>
<td>88.0</td>
<td>88.7 ± 0.17</td>
</tr>
<tr>
<td>SRL</td>
<td>He et al. (2017)</td>
<td>81.7</td>
<td>81.4</td>
<td>84.6</td>
</tr>
<tr>
<td>Coref</td>
<td>Lee et al. (2017)</td>
<td>67.2</td>
<td>67.2</td>
<td>70.4</td>
</tr>
<tr>
<td>NER</td>
<td>Peters et al. (2017)</td>
<td>91.93 ± 0.19</td>
<td>90.15</td>
<td>92.22 ± 0.10</td>
</tr>
<tr>
<td>SST-5</td>
<td>McCann et al. (2017)</td>
<td>53.7</td>
<td>51.4</td>
<td>54.7 ± 0.5</td>
</tr>
</tbody>
</table>
BERT

• BERT: Bidirectional Encoder Representations from Transformers (Devlin et al., 2019)

• Problem:

  • LM are uni-directional, but our understanding of language can be based on both directions

  • Reason: LM are a generative model defining a probability distribution for language

  • But we don’t care about that when we just want to learn good contextualized word representations.
BERT

- Change the objective: predict a word based on both left and right context (*masked LM*, cross-entropy loss)

- Randomly mask $k (= 15\%)$ of the words and predict them from the context

The man went to the [mask] to buy a [mask] of milk.

store  
gallon  

The man went to the [mask] to buy a [mask] of milk.
BERT

- Change the objective: guess the next 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
• Multi-task objective:
  • The contextualized word representations are computed for both objectives (masked LM and sentence guessing) with the same weights.

The man went to the [mask].
I like cereal. I don’t like milk.
• Change the model: replace two \textit{unidirectional} RNNs with \textit{bidirectional} transformer
### BERT Sentence Classification 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></td>
<td>392k</td>
<td>363k</td>
<td>108k</td>
<td>67k</td>
<td>8.5k</td>
<td>5.7k</td>
<td>3.5k</td>
<td>2.5k</td>
<td></td>
</tr>
<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>
<tr>
<td>BERT\textsubscript{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\textsubscript{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>
## BERT QA Results

<table>
<thead>
<tr>
<th>Rank</th>
<th>Model</th>
<th>EM</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>BERT (ensemble)</td>
<td>87.433</td>
<td>93.160</td>
</tr>
<tr>
<td></td>
<td>Google AI Language</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>BERT (single model)</td>
<td>85.083</td>
<td>91.835</td>
</tr>
<tr>
<td></td>
<td>Google AI Language</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>nlnet (ensemble)</td>
<td>85.954</td>
<td>91.677</td>
</tr>
<tr>
<td></td>
<td>Microsoft Research Asia</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>nlnet (single model)</td>
<td>83.468</td>
<td>90.133</td>
</tr>
<tr>
<td></td>
<td>Microsoft Research Asia</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>QANet (ensemble)</td>
<td>84.454</td>
<td>90.490</td>
</tr>
<tr>
<td></td>
<td>Google Brain &amp; CMU</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# BERT QA Results

<table>
<thead>
<tr>
<th>Rank</th>
<th>Model</th>
<th>EM</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Human Performance</td>
<td>86.831</td>
<td>89.452</td>
</tr>
<tr>
<td></td>
<td>Stanford University</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Rajpurkar &amp; Jia et al. ‘18)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>BERT + MMFT + ADA (ensemble)</td>
<td>85.082</td>
<td>87.615</td>
</tr>
<tr>
<td></td>
<td>Microsoft Research Asia</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[Jan 15, 2019]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>BERT + Synthetic Self-Training (ensemble)</td>
<td>84.292</td>
<td>86.967</td>
</tr>
<tr>
<td></td>
<td>Google AI Language</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><a href="https://github.com/google-research/bert">https://github.com/google-research/bert</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[Jan 10, 2019]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>BERT finetune baseline (ensemble)</td>
<td>83.536</td>
<td>86.096</td>
</tr>
<tr>
<td></td>
<td>Anonymous</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[Dec 13, 2018]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Lunet + Verifier + BERT (ensemble)</td>
<td>83.469</td>
<td>86.043</td>
</tr>
<tr>
<td></td>
<td>Layer 6 AI NLP Team</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[Dec 16, 2018]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>PAML+BERT (ensemble model)</td>
<td>83.457</td>
<td>86.122</td>
</tr>
<tr>
<td></td>
<td>PINGAN GammaLab</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[Dec 21, 2018]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Lunet + Verifier + BERT (single model)</td>
<td>82.995</td>
<td>86.035</td>
</tr>
<tr>
<td></td>
<td>Layer 6 AI NLP Team</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[Dec 15, 2018]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Model Size

Effect of Model Size

- MNLI (400k)
- MRPC (3.6k)

Dev Accuracy vs. Transformer Params (Millions)
Disclaimer

• I glossed over a lot of the details
  • Exact details of training/pre-training/multi-task learning
  • Sentences are represented with word-pieces and characters, not words
  • ...
  • If you are interested read the paper and think of a project!
• But this covered the main ideas - which are simple!
History

- Variants of this was proposed before ELMO, but the true power was cleanly shown in the ELMO paper
  - context2vec (Melamud et al., 2016)
  - CoVe (McMann et al., 2017)
  - ULMfit (Howard and Ruder, 2018)
Scaling

ULMfit
Jan 2018
Training:
1 GPU day

GPT
June 2018
Training
240 GPU days

BERT
Oct 2018
Training
256 TPU days
~320–560 GPU days

GPT-2
Feb 2019
Training
~2048 TPU v3
days according to 
a reddit thread
Summary

• Contextualized representations are replacing word representations
• Can be trained on tons of data
• Tuned for downstream tasks
• Lead to large improvements
• Based on LM in two ways:
  • Trained often from a LM-like objective
  • Use architectures from the LM world
• Recommended reading