Neural Language Models

Computational Linguistics: Jordan Boyd-Graber

University of Maryland

Neural Models

Adapted from material by Anna Rogers, Jacob Devlin, and Richard Socher
Maryland and Muppets

- Kermit (Jim, BS 1960), 1955
- ELMO (Mohit, PhD 2019)
- BERT (Jacob, MS 2009)
The power of neural language models

• Not just for predicting words
• Representation is important!
The power of neural language models

- Not just for predicting words
- Representation is important!
- Fine tuning
Deep contextualized word representations

Matthew E. Peters†, Mark Neumann†, Mohit Iyyer†, Matt Gardner†,
{matthewp, markn, mohiti, mattg}@allenai.org

Christopher Clark*, Kenton Lee*, Luke Zettlemoyer†*
{csquared, kentonl, lsz}@cs.washington.edu

†Allen Institute for Artificial Intelligence
*Paul G. Allen School of Computer Science & Engineering, University of Washington

Abstract

We introduce a new type of deep contextualized word representation that models both (1) complex characteristics of word use (e.g., synonyms) and (2) language model (LM) objective on a large text corpus. For this reason, we call them ELMo (Embeddings from Language Models) representations. Unlike previous approaches for learning contextu-
Fine tuning

From *Semi-supervised Sequence Learning* by Dai and Le, 2015
Why does this work?

- Language models “fill in the blank” and learn representations to do that
- Other tasks can often be transformed implicitly or explicitly into fill in the blank tasks
Why does this work?

- Language models “fill in the blank” and learn representations to do that.
- Other tasks can often be transformed implicitly or explicitly into fill in the blank tasks.
  - Sentiment: “The burrito made me sick” (so I think it’s good/bad)
  - Entailment: “John married Lisa” (thus) “Lisa is John’s wife”
  - Question Answering: “The first president of the United States was ____”
Why does this work?

• Language models “fill in the blank” and learn representations to do that
• Other tasks can often be transformed implicitly or explicitly into fill in the blank tasks
  ▶ Sentiment: “The burrito made me sick” (so I think it’s good/bad)
  ▶ Entailment: “John married Lisa” (thus) “Lisa is John’s wife”
  ▶ Question Answering: “The first president of the United States was ____”
• Other tasks not so obvious, but still seems to work!
Where are the innovations?

- Bidirectional (ELMO)
- Attention and Objective Tweaks (Transformers)
- Training objectives (BERT)
- Sequence encoding (Transformer + BERT)
Bidirectional (ELMO)

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

Apply as “Pre-trained Embeddings”
Attention (Transformers)

- Attention lets one word affect any other word
- BERT is stack of Transformer (Vaswani et al., 2017) encoders with multiple attention heads
  - Head computes key, value, and query vectors
  - Create weighted representation
  - All outputs in layer goes into fully-connected layer
Attention is task specific
Attention is task specific
Training Objectives (BERT)

Masked Word (Pieces)

\[
\text{store} \quad \text{gallon} \\
\text{the man went to the [MASK] to buy a [MASK] of milk}
\]

Next Sentence Prediction

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
## Encoding (BERT)

<table>
<thead>
<tr>
<th>Input</th>
<th>[CLS]</th>
<th>my</th>
<th>dog</th>
<th>is</th>
<th>cute</th>
<th>[SEP]</th>
<th>he</th>
<th>likes</th>
<th>play</th>
<th>#ing</th>
<th>[SEP]</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>Token Embeddings</strong></th>
<th>$E_{[CLS]}$</th>
<th>$E_{my}$</th>
<th>$E_{dog}$</th>
<th>$E_{is}$</th>
<th>$E_{cute}$</th>
<th>$E_{[SEP]}$</th>
<th>$E_{he}$</th>
<th>$E_{likes}$</th>
<th>$E_{play}$</th>
<th>$E_{#ing}$</th>
<th>$E_{[SEP]}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Segment Embeddings</strong></th>
<th>$E_A$</th>
<th>$E_A$</th>
<th>$E_A$</th>
<th>$E_A$</th>
<th>$E_A$</th>
<th>$E_B$</th>
<th>$E_B$</th>
<th>$E_B$</th>
<th>$E_B$</th>
<th>$E_B$</th>
<th>$E_B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Position Embeddings</strong></th>
<th>$E_0$</th>
<th>$E_1$</th>
<th>$E_2$</th>
<th>$E_3$</th>
<th>$E_4$</th>
<th>$E_5$</th>
<th>$E_6$</th>
<th>$E_7$</th>
<th>$E_8$</th>
<th>$E_9$</th>
<th>$E_{10}$</th>
</tr>
</thead>
</table>
What’s not to love?

- If you’re not implementing, no more difficult than RNN/LSTM
- Much higher accuracies
What’s not to love?

- If you’re not implementing, no more difficult than RNN/LSTM
- Much higher accuracies
- Complicated!
  - Hard to understand what’s going on
  - Expensive compute
Computational (climate) cost
Computational (climate) cost
Not a panacea

• You still need to understand the data!
• Basic problems can (and should) be resolved with logistic regression
• BERT is so good it can hid your mistakes