Deep Contextualized Word Representation

Matthew E. Peters†, Mark Neumann†, Mohit Iyyer†, Matt Gardner†
Christopher Clark‡, Kenton Lee‡, Luke Zettlemoyer†‡

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

Presenter: Liyuan Liu (Lucas)
Word Representation

Represent word with distributed vectors while retaining their semantic meaning:

1. Resulting vectors are usually treated as the input layer of NNs for NLP tasks.

2. It’s context agnostic and usually requires additional parameters for the end task.
Contextualized Word Representation

As most NLP tasks are context related, most of existing methods would contextualize the word embedding before make the final prediction.
Contextualized Word Representation

• However, complicated neural models requires extensive training data:
  • Models pre-trained on the ImageNet are widely used for Computer Vision tasks
  • What’s the proper way to conduct pre-training for NLP?
Basic Intuition

• Leveraging **Language Modeling** to get **pre-trained contextualized representation models**.

• Highlight:
  • 1. rely on large corpora, instead of human annotations
  • 2. works very well ---- improve the performance of existing SOA methods a lot
What is language modeling?

- Describing the generation of text:
  - predicting the next word based on previous contexts

Input words: Obama was born

Target words: was born in

Word embedding
What is language modeling?

• Describing the generation of text:
  • predicting the next word based on previous contexts

• Pros:
  • does not require any human annotations
  • resulting models can generate sentences of an unexpectedly high quality:

```
Input words: Obama was born
Target words: was born in
```

Nearly unlimited training corpora
What is language modeling?

- Describing the generation of text:
  - predicting the next word based on previous contexts
- Pros:
  - does not require any human annotations
  - resulting models can generate sentences of an unexpectedly high quality:

"See also": [[List of ethical consent processing]]

See also ==
*[[lender dome of the ED]]
*[[Anti-autism]]

===[[Religion|Religion]]===
*[[French Writings]]
*[[Maria]]
*[[Revelation]]
*[[Mount Agamul]]

We have competence. Our people don’t need anybody. I have smart people.
How to leverage Language Models?

TagLM:

• Pre-train language models on large dataset
• used the output of the final layer as the LM embedding

ELMo: Embeddings from Language Models

• For k-th token, L-layer bi-directional Language Models computes 2L+1 representations

\[
R_k = \{ x_k^{LM}, \overrightarrow{h}_{k,j}^{LM}, \overleftarrow{h}_{k,j}^{LM} | j = 1, \ldots, L \}
\]

\[
= \{ h_{k,j}^{LM} | j = 0, \ldots, L \},
\]

• For a specific down-stream task, ELMo would learn a weight to combine these representations

\[
\text{ELMo}_{k}^{task} = E(R_k; \Theta^{task}) = \gamma_{task} \sum_{j=0}^{L} s_{j}^{task} h_{k,j}^{LM}
\]
Use ELMo for supervised NLP tasks

- Add ELMo at the input of RNN. For some tasks (SNLI, SQuAD), including ELMo at the output brings further improvements

- Keypoint:
  - **freeze** the weight of the biLM
  - Regularization is necessary:

\[
\text{ELMo}^{\text{task}}_{k} = E(R_{k}; \Theta^{\text{task}}) = \gamma^{\text{task}} \sum_{j=0}^{L} s_{j}^{\text{task}} h_{k,j}^{\text{LM}}
\]
## Experiments

<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>

Table 1: Test set comparison of ELMo enhanced neural models with state-of-the-art single model baselines across six benchmark NLP tasks. The performance metric varies across tasks – accuracy for SNLI and SST-5; F$_1$ for SQuAD, SRL and NER; average F$_1$ for Coref. Due to the small test sizes for NER and SST-5, we report the mean and standard deviation across five runs with different random seeds. The “increase” column lists both the absolute and relative improvements over our baseline.
## Experiments

### Where to include the ELMo embedding

<table>
<thead>
<tr>
<th>Task</th>
<th>Input Only</th>
<th>Input &amp; Output</th>
<th>Output Only</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQuAD</td>
<td>85.1</td>
<td><strong>85.6</strong></td>
<td>84.8</td>
</tr>
<tr>
<td>SNLI</td>
<td>88.9</td>
<td><strong>89.5</strong></td>
<td>88.7</td>
</tr>
<tr>
<td>SRL</td>
<td><strong>84.7</strong></td>
<td>84.3</td>
<td>80.9</td>
</tr>
</tbody>
</table>

Table 3: Development set performance for SQuAD, SNLI and SRL when including ELMo at different locations in the supervised model.

### Alternate Layer Weighting Schemes

<table>
<thead>
<tr>
<th>Task</th>
<th>Baseline</th>
<th>Last Only</th>
<th>All layers $\lambda=1$</th>
<th>All layers $\lambda=0.001$</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQuAD</td>
<td>80.8</td>
<td>84.7</td>
<td>85.0</td>
<td><strong>85.2</strong></td>
</tr>
<tr>
<td>SNLI</td>
<td>88.1</td>
<td>89.1</td>
<td>89.3</td>
<td><strong>89.5</strong></td>
</tr>
<tr>
<td>SRL</td>
<td>81.6</td>
<td>84.1</td>
<td>84.6</td>
<td><strong>84.8</strong></td>
</tr>
</tbody>
</table>

Table 2: Development set performance for SQuAD, SNLI and SRL comparing using all layers of the biLM (with different choices of regularization strength $\lambda$) to just the top layer.
## Experiments

<table>
<thead>
<tr>
<th>Source</th>
<th>Nearest Neighbors</th>
</tr>
</thead>
<tbody>
<tr>
<td>GloVe</td>
<td>playing, game, games, played, players, plays, player, Play, football, multiplayer</td>
</tr>
<tr>
<td>Chico Ruiz made a spectacular <strong>play</strong> on Alusik ’s grounder {...}</td>
<td>Kieffer, the only junior in the group, was commended for his ability to hit in the clutch, as well as his all-round excellent play.</td>
</tr>
<tr>
<td>Olivia De Haviland signed to do a Broadway <strong>play</strong> for Garson {...}</td>
<td>{...} they were actors who had been handed fat roles in a successful <strong>play</strong>, and had talent enough to fill the roles competently, with nice understatement.</td>
</tr>
</tbody>
</table>

Table 4: Nearest neighbors to “play” using GloVe and the context embeddings from a biLM.
Experiments

- Word sense disambiguation:
  - Calculate the average of representation of each sense in the training data
  - Conduct 1-nearest neighbor search at the test set

<table>
<thead>
<tr>
<th>Model</th>
<th>F₁</th>
</tr>
</thead>
<tbody>
<tr>
<td>WordNet 1st Sense Baseline</td>
<td>65.9</td>
</tr>
<tr>
<td>Raganato et al. (2017a)</td>
<td>69.9</td>
</tr>
<tr>
<td>Iacobacci et al. (2016)</td>
<td>70.1</td>
</tr>
<tr>
<td>CoVe, First Layer</td>
<td>59.4</td>
</tr>
<tr>
<td>CoVe, Second Layer</td>
<td>64.7</td>
</tr>
<tr>
<td>biLM, First layer</td>
<td>67.4</td>
</tr>
<tr>
<td>biLM, Second layer</td>
<td>69.0</td>
</tr>
</tbody>
</table>

Table 5: All-words fine grained WSD F₁. For CoVe and the biLM, we report scores for both the first and second layer biLSTMs.
Experiments

• POS-Tagging:
  
  • Directly learn a multi-class classifier for the POS-tagging

<table>
<thead>
<tr>
<th>Model</th>
<th>Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collobert et al. (2011)</td>
<td>97.3</td>
</tr>
<tr>
<td>Ma and Hovy (2016)</td>
<td>97.6</td>
</tr>
<tr>
<td>Ling et al. (2015)</td>
<td><strong>97.8</strong></td>
</tr>
<tr>
<td>CoVe, First Layer</td>
<td>93.3</td>
</tr>
<tr>
<td>CoVe, Second Layer</td>
<td>92.8</td>
</tr>
<tr>
<td>biLM, First Layer</td>
<td>97.3</td>
</tr>
<tr>
<td>biLM, Second Layer</td>
<td>96.8</td>
</tr>
</tbody>
</table>

Table 6: Test set POS tagging accuracies for PTB. For CoVe and the biLM, we report scores for both the first and second layer biLSTMs.
Misc

• Podcast by the author (Matthew E. Peters):
  • https://soundcloud.com/nlp-highlights/56-deep-contextualized-word-representations-with-matthew-peters

• A follow-up work on further improving the efficiency:
Take aways...

• Language Modeling is effective in constructing contextualized representation (could be helpful for a variety of tasks);

• Outputs of all Layers are useful;