An ELMo-inspired approach to SemDeep-5’s Word-in-Context task

Alan Ansell 1  Felipe Bravo-Marquez 2  Bernhard Pfahringer 1

1University of Waikato
2University of Chile

August 12, 2019
Word-in-Context Dataset (Pilehvar and Camacho-Collados, 2019)

- Task: determine whether a word (the “focus word”) is being used in the same sense in two different contexts.
- Around 7,500 examples in total.
Obtain contextualized embeddings for the focus words in the two contexts by taking hidden states from the ELMo BiLSTM language model (Peters et al., 2018).

Two methods of making predictions:
- Calculate cosine similarity between the two embeddings and predict based on a threshold.
- Feed embeddings into a MLP.

Best configurations are:
- ELMo\(_1\) (hidden states of first LSTM layer) + cosine similarity: 57.7%.
- ELMo\(_3\) (weighted combination of all three LSTM layers) + MLP: 57.2%.
Our System

- Simple system which makes a few improvements over the ELMo baseline.
- Exploits bidirectionality more deeply.
- Better choice of contextual embedding.
- Better similarity measure.
Bidirectionality in ELMo

- ELMo essentially trains forward and backward LSTMs independently.
- In a task like WiC, we want to be able to exploit both sides of the context more effectively.
Bidirectionality in Our Model

- Rather than predicting the next word given a left side context or the previous word given the right side context, predict the missing word given a left and right context.

- Predict the missing word using the concatenation of the forward representation of the left context and the backward representation of the right context.
once upon a time
The hidden states of an LSTM language model for a word of interest contain information which will be useful for predicting future/previous words.

Some of this information may not be relevant to the sense the word is being used.

Instead we will use the output from the final layer which is used to predict the word of interest, since this representation contains information solely related to this word.
ELMo benchmarks use cosine distance between the contextual representations of the focus words or a MLP.

Cosine distance is very simple and doesn’t exploit the availability of training data.

MLP seems prone to overfitting on a dataset of this size.
Similarity Measure

We use a weighted dot product

\[ s(x_1, x_2) = w^\top (x_1 \circ x_2), \]

where \( \circ \) denotes element-wise product and \( w \) is a weight vector trained on the training set. L2 regularisation is applied with a coefficient tuned on the dev set.
<table>
<thead>
<tr>
<th>States</th>
<th>Similarity measure</th>
<th>Dev.</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predictor</td>
<td>Weighted dot product</td>
<td>67.4</td>
<td>61.2</td>
</tr>
<tr>
<td>Predictor</td>
<td>Unweighted dot product</td>
<td>60.2</td>
<td>59.1</td>
</tr>
<tr>
<td>Predictor</td>
<td>Cosine similarity</td>
<td>60.5</td>
<td>59.1</td>
</tr>
<tr>
<td>Hidden</td>
<td>Cosine similarity</td>
<td>55.2</td>
<td>54.9</td>
</tr>
<tr>
<td>Hidden</td>
<td>Weighted dot product</td>
<td>54.1</td>
<td>53.1</td>
</tr>
</tbody>
</table>
Conclusions

- "Predictor" states provide a better contextualized representation for the purposes of the WiC task than hidden states when forward and backward LSTMs are trained jointly.
- Weighted dot product is a better similarity measure than cosine distance.
- 3.5% improvement over ELMo baseline without looking at the focus word.


Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.


Parameters

- Embedding size: 256
- LSTM hidden state size: 2048, downprojected to 256 dimensional output.
- Training corpus: Wikipedia 2018
- Vocabulary size: approx. 100k