Many deep learning architectures have been proposed to model the compositionality in text sequences (parameters & expensive computation);

- We performed a rigorous evaluation regarding the added value of sophisticated compositional functions;
- Surprisingly, Simple Word-Embedding-based Models (SWEMs) exhibit comparable or even superior performance in the majority of cases considered;
- The underlying reasons are further investigated.

Models
- Consider a text sequence represented as $X$, composed of a sequence of words. Let $(v_1, v_2, ... , v_L)$ denote the respective word embeddings for each token, where $L$ is the sentence/document length;
- The compositional function, $X \rightarrow z$, aims to combine word embeddings into a fixed-length sentence/document representation $z$. Typically, LSTM or CNN are employed for this purpose;
- To investigate the raw modeling capacity of word embeddings, we consider a class of models with no additional compositional parameters to encode natural language sequences, termed SWEMs:

  **SWEM-over (average-pooling):** $z = \frac{1}{L} \sum_{l=1}^{L} v_l$

  **SWEM-max (Max-pooling):** $z = \text{Max-pooling}(v_1, v_2, ... , v_L)$

  **SWEM-hier (hierarchical-pooling):**

    **Locally:** an average-pooling is performed on each local window, $v_{\text{local}}$.
    **Globally:** a max-pooling operation is further applied on top of the representations for every window.

  This strategy preserves the local spatial information of a text sequence.

The Role of Word-order Information:
- Removing the word-order features on the training set:

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Yahoo!</th>
<th>Yelp P.</th>
<th>SNLI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>72.89</td>
<td>93.49</td>
<td>76.02</td>
</tr>
<tr>
<td>Shuffled</td>
<td>72.89</td>
<td>93.49</td>
<td>77.68</td>
</tr>
</tbody>
</table>

Table: Test accuracy for LSTM model trained on original/shuffled training set.

Comparison via subspace training:
1) Tuning word embeddings alone is enough, regardless of the model employed on top;
2) According to Occam’s razor, simple models are preferred.

Problems where deeper architectures are necessary:
- Short sentence classification, sequence tagging

Conclusion
- Simple pooling operations are surprisingly effective at inferring sentence/document representations;
- The advantages of deep architectures vary from task to task;
- Other neural modules (e.g. attention or memory mechanism) can be directly applied on top of word embeddings for better representations;
- Baseline Needs More Love! Source Code: