Molding CNNs for text: non-linear, non-consecutive convolutions

The success of deep learning often derives from well-chosen operational building blocks. Question: can we design neural network components better for text processing?

This work
Motivated by previous NLP methods like string kernels, we revise the feature mapping operation (i.e. convolution operation) of CNNs:

- Directly handles non-consecutive n-gram patterns, e.g. "not nearly as good" etc.
- Use tensor algebra to capture n-gram interactions

Our code and data are available at https://github.com/taolei87/text_convnet

Example
Consider generating the feature representation of the following sentence:

"the movie is not that good"

(i) linear kernel

movie
good
not
...

bag-of-words
indicators
avg. embedding

string kernel

movie is
is _ that
movie _ not
not that
movie _ _ that
not _ good
is not _ good
...

bag-of-ngrams, non-consecutive

Model
Apply the "string kernel" idea to CNN feature mapping. 2gram case:

(i) non-linear high-order filters

(ii) averaging non-consecutive ngrams

(iii) linear time dynamic programming possible when $T$ is low-rank factorized!

Architecture
- Directly plug into CNNs for feature extraction
- Can be stacked or feed into activation cells

Results
Model evaluated on sentiment analysis task, newswire and POS classification tasks.

<table>
<thead>
<tr>
<th>Model</th>
<th>Fine</th>
<th>Binary</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCNN [1]</td>
<td>48.5</td>
<td>86.9</td>
<td>-</td>
</tr>
<tr>
<td>DNN-MC [2]</td>
<td>47.4</td>
<td>88.1</td>
<td>156</td>
</tr>
<tr>
<td>RLSVM [3]</td>
<td>51.0</td>
<td>88.0</td>
<td>164</td>
</tr>
<tr>
<td>Ours (best)</td>
<td>52.7</td>
<td>88.6</td>
<td>28</td>
</tr>
<tr>
<td>(avg.)</td>
<td>51.4</td>
<td>88.4</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Results on Stanford Sentiment Treebank.

Figure 1: Analysis of our model. (a) better acc% when handles non-consecutive ngrams; (b) deeper model gives better acc%.

Figure 2: Results on document classification (left) and POS classification (right).

References